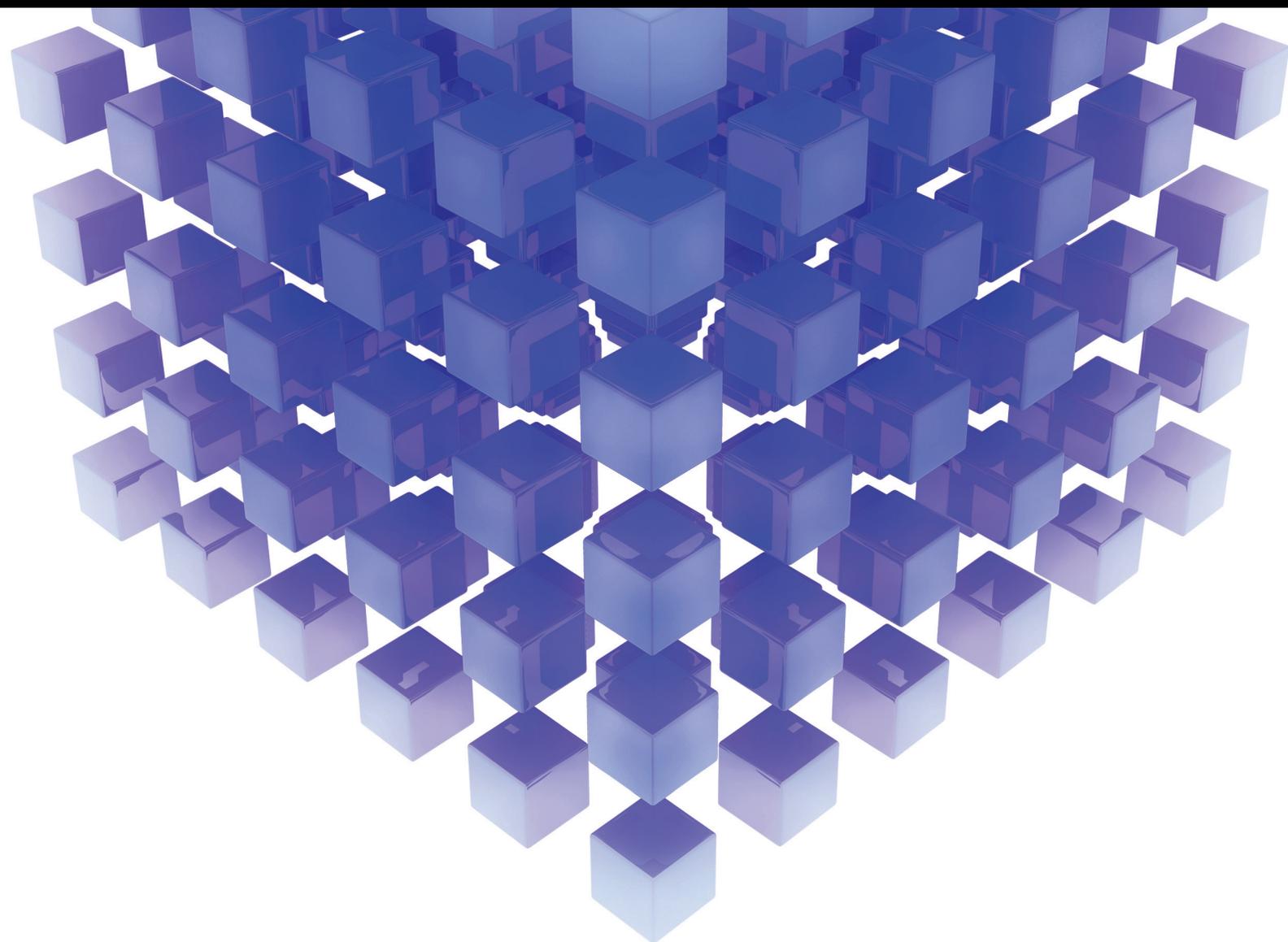


Mathematical Problems in Engineering

# Optimization Algorithms Combining (Meta)heuristics and Mathematical Programming and Its Application in Engineering

Lead Guest Editor: Nivaldo Rodríguez

Guest Editors: Abhishek K. Gupta, Guillermo Cabrera G., and Paula L. Zabala





---

**Optimization Algorithms Combining  
(Meta)heuristics and Mathematical  
Programming and Its Application  
in Engineering**

Mathematical Problems in Engineering

---

**Optimization Algorithms Combining  
(Meta)heuristics and Mathematical  
Programming and Its Application  
in Engineering**

Lead Guest Editor: Nivaldo Rodríguez

Guest Editors: Abhishek K. Gupta, Guillermo Cabrera G.,  
and Paula L. Zabala



---

Copyright © 2018 Hindawi. All rights reserved.

This is a special issue published in “Mathematical Problems in Engineering.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## Editorial Board

- Mohamed Abd El Aziz, Egypt  
AITOUCHE Abdelouhab, France  
Leonardo Acho, Spain  
José A. Acosta, Spain  
Daniela Addressi, Italy  
Paolo Adesso, Italy  
Claudia Adduce, Italy  
Ramesh Agarwal, USA  
Francesco Aggogeri, Italy  
Juan C. Agüero, Australia  
R Aguilar-López, Mexico  
Tarek Ahmed-Ali, France  
Elias Aifantis, USA  
Muhammad N. Akram, Norway  
Guido Ala, Italy  
Andrea Alaimo, Italy  
Reza Alam, USA  
Nicholas Alexander, UK  
Salvatore Alfonzetti, Italy  
Mohammad D. Aliyu, Canada  
Juan A. Almendral, Spain  
José Domingo Álvarez, Spain  
Cláudio Alves, Portugal  
J. P. Amezcua-Sanchez, Mexico  
Lionel Amodeo, France  
Sebastian Anita, Romania  
Renata Archetti, Italy  
Felice Arena, Italy  
Sabri Arik, Turkey  
Francesco Aristodemo, Italy  
Fausto Arpino, Italy  
Alessandro Arsie, USA  
Edoardo Artioli, Italy  
Fumihiko Ashida, Japan  
Farhad Aslani, Australia  
Mohsen Asle Zaeem, USA  
Romain Aubry, USA  
Matteo Aureli, USA  
Richard I. Avery, USA  
Viktor Avrutin, Germany  
Francesco Aymerich, Italy  
Sajad Azizi, Belgium  
Michele Baccocchi, Italy  
Seungik Baek, USA  
Adil Bagirov, Australia  
Khaled Bahlali, France  
Laurent Bako, France  
Pedro Balaguer, Spain  
Stefan Balint, Romania  
Ines Tejado Balsera, Spain  
Alfonso Banos, Spain  
Jerzy Baranowski, Poland  
Roberto Baratti, Italy  
Andrzej Bartoszewicz, Poland  
David Bassir, France  
Chiara Bedon, Italy  
Azeddine Beghdadi, France  
Denis Benasciutti, Italy  
Ivano Benedetti, Italy  
Rosa M. Benito, Spain  
Elena Benvenuti, Italy  
Giovanni Berselli, Italy  
Giorgio Besagni, Italy  
Michele Betti, Italy  
Jean-Charles Beugnot, France  
Pietro Bia, Italy  
Carlo Bianca, France  
Simone Bianco, Italy  
Vincenzo Bianco, Italy  
Vittorio Bianco, Italy  
Gennaro N. Bifulco, Italy  
David Bigaud, France  
Antonio Bilotta, Italy  
Paul Bogdan, USA  
Guido Bolognesi, UK  
Rodolfo Bontempo, Italy  
Alberto Borboni, Italy  
Paolo Boscariol, Italy  
Daniela Boso, Italy  
Guillermo Botella-Juan, Spain  
Boulaïd Boulkroune, Belgium  
Fabio Bovenga, Italy  
Francesco Braghin, Italy  
Ricardo Branco, Portugal  
Maurizio Brocchini, Italy  
Julien Bruchon, France  
Matteo Bruggi, Italy  
Michele Brun, Italy  
Vasilis Burganos, Greece  
Tito Busani, USA  
Raquel Caballero-Águila, Spain  
Filippo Cacace, Italy  
Pierfrancesco Cacciola, UK  
Salvatore Caddemi, Italy  
Roberto Caldelli, Italy  
Alberto Campagnolo, Italy  
Eric Campos-Canton, Mexico  
Marko Canadija, Croatia  
Salvatore Cannella, Italy  
Francesco Cannizzaro, Italy  
Javier Cara, Spain  
Ana Carpio, Spain  
Caterina Casavola, Italy  
Sara Casciati, Italy  
Federica Caselli, Italy  
Carmen Castillo, Spain  
Inmaculada T. Castro, Spain  
Miguel Castro, Portugal  
Giuseppe Catalanotti, UK  
Nicola Caterino, Italy  
Alberto Cavallo, Italy  
Gabriele Cazzulani, Italy  
Luis Cea, Spain  
Miguel Cerrolaza, Venezuela  
M. Chadli, France  
Gregory Chagnon, France  
Ludovic Chamoin, France  
Ching-Ter Chang, Taiwan  
Qing Chang, USA  
Michael J. Chappell, UK  
Kacem Chehdi, France  
Peter N. Cheimets, USA  
Xinkai Chen, Japan  
Luca Chiapponi, Italy  
Francisco Chicano, Spain  
Nicholas Chileshe, Australia  
Adrian Chmielewski, Poland  
Ioannis T. Christou, Greece  
Hung-Yuan Chung, Taiwan  
Simone Cinquemani, Italy  
Roberto G. Citarella, Italy  
Joaquim Ciurana, Spain

John D. Clayton, USA  
Francesco Clementi, Italy  
Piero Colajanni, Italy  
Giuseppina Colicchio, Italy  
Vassilios Constantoudis, Greece  
Enrico Conte, Italy  
Francesco Conte, Italy  
Alessandro Contento, USA  
Mario Cools, Belgium  
José A. Correia, Portugal  
Jean-Pierre Corriou, France  
J.-C. Cortés, Spain  
Carlo Cosentino, Italy  
Paolo Crippa, Italy  
Andrea Crivellini, Italy  
Frederico R. B. Cruz, Brazil  
Erik Cuevas, Mexico  
Maria C. Cunha, Portugal  
Peter Dabnichki, Australia  
Luca D'Acierno, Italy  
Weizhong Dai, USA  
Andrea Dall'Asta, Italy  
P. Damodaran, USA  
Farhang Daneshmand, Canada  
Giuseppe D'Aniello, Italy  
Sergey N. Dashkovskiy, Germany  
Fabio De Angelis, Italy  
Samuele De Bartolo, Italy  
Abílio De Jesus, Portugal  
Pietro De Lellis, Italy  
Alessandro De Luca, Italy  
Stefano de Miranda, Italy  
Filippo de Monte, Italy  
M. do Rosário de Pinho, Portugal  
Michael Defoort, France  
Alessandro Della Corte, Italy  
Xavier Delorme, France  
Laurent Dewasme, Belgium  
Angelo Di Egidio, Italy  
Roberta Di Pace, Italy  
Ramón I. Diego, Spain  
Yannis Dimakopoulos, Greece  
Zhengtao Ding, UK  
M. Djemai, France  
Alexandre B. Dolgui, France  
Georgios Dounias, Greece  
Florent Duchaine, France  
George S. Dulikravich, USA  
Bogdan Dumitrescu, Romania  
Horst Ecker, Austria  
Saeed Eftekhar Azam, USA  
Ahmed El Hajjaji, France  
Antonio Elipe, Spain  
Fouad Erchiqui, Canada  
Anders Eriksson, Sweden  
R. Emre Erkmen, Australia  
Gilberto Espinosa-Paredes, Italy  
Leandro F. F. Miguel, Brazil  
Andrea L. Facci, Italy  
Giacomo Falcucci, Italy  
Giovanni Falsone, Italy  
Hua Fan, China  
Nicholas Fantuzzi, Italy  
Yann Favennec, France  
Fiorenzo A. Fazzolari, UK  
Giuseppe Fedele, Italy  
Roberto Fedele, Italy  
Arturo J. Fernández, Spain  
Jesus M. Fernandez Oro, Spain  
Massimiliano Ferraioli, Italy  
Massimiliano Ferrara, Italy  
Francesco Ferrise, Italy  
Eric Feulvarch, France  
Barak Fishbain, Israel  
S. Douwe Flapper, Netherlands  
Thierry Floquet, France  
Eric Florentin, France  
Alessandro Formisano, Italy  
Francesco Franco, Italy  
Elisa Francomano, Italy  
Tomonari Furukawa, USA  
Juan C. G. Prada, Spain  
Mohamed Gadala, Canada  
Matteo Gaeta, Italy  
Mauro Gaggero, Italy  
Zoran Gajic, USA  
Erez Gal, Israel  
Jaime Gallardo-Alvarado, Mexico  
Ugo Galvanetto, Italy  
Akemi Gálvez, Spain  
Rita Gamberini, Italy  
Maria L. Gandarias, Spain  
Arman Ganji, Canada  
Zhiwei Gao, UK  
Zhong-Ke Gao, China  
Giovanni Garcea, Italy  
Luis Rodolfo Garcia Carrillo, USA  
Jose M. Garcia-Aznar, Spain  
Akhil Garg, China  
Alessandro Gasparetto, Italy  
Oleg V. Gendelman, Israel  
Stylios Georgantzinos, Greece  
Fotios Georgiades, UK  
Mergen H. Ghayesh, Australia  
Georgios I. Giannopoulos, Greece  
Agathoklis Giaralis, UK  
Pablo Gil, Spain  
Anna M. Gil-Lafuente, Spain  
Ivan Giorgio, Italy  
Gaetano Giunta, Luxembourg  
Alessio Gizzi, Italy  
Jefferson L.M.A. Gomes, UK  
Emilio Gómez-Déniz, Spain  
Antonio M. Gonçalves de Lima, Brazil  
David González, Spain  
Chris Goodrich, USA  
Rama S. R. Gorla, USA  
Kannan Govindan, Denmark  
Antoine Grall, France  
George A. Gravvanis, Greece  
Fabrizio Greco, Italy  
David Greiner, Spain  
Simonetta Grilli, Italy  
Jason Gu, Canada  
Federico Guarracino, Italy  
Michele Guida, Italy  
José L. Guzmán, Spain  
Quang Phuc Ha, Australia  
Petr Hájek, Czech Republic  
Zhen-Lai Han, China  
Thomas Hanne, Switzerland  
Mohammad A. Hariri-Ardebili, USA  
Xiao-Qiao He, China  
Luca Heltai, Italy  
Nicolae Herisanu, Romania  
Alfredo G. Hernández-Díaz, Spain  
M.I. Herreros, Spain  
Eckhard Hitzer, Japan  
Paul Honeine, France  
Jaromir Horacek, Czech Republic  
Muneo Hori, Japan

András Horváth, Italy  
S. H. Hosseinnia, Netherlands  
Mengqi Hu, USA  
Gordon Huang, Canada  
Sajid Hussain, Canada  
Asier Ibeas, Spain  
Orest V. Iftime, Netherlands  
Przemyslaw Ignaciuk, Poland  
Giacomo Innocenti, Italy  
Emilio Insfran Pelozo, Spain  
Alessio Ishizaka, UK  
Nazrul Islam, USA  
Benoit Jung, France  
Benjamin Ivorra, Spain  
Payman Jalali, Finland  
Mahdi Jalili, Australia  
Łukasz Jankowski, Poland  
Samuel N. Jator, USA  
Juan C. Jauregui-Correa, Mexico  
Reza Jazar, Australia  
Khalide Jbilou, France  
Piotr Jędrzejowicz, Poland  
Isabel S. Jesus, Portugal  
Linni Jian, China  
Bin Jiang, China  
Zhongping Jiang, USA  
Emilio Jiménez Macías, Spain  
Ningde Jin, China  
Xiaoliang Jin, USA  
Liang Jing, Canada  
Dylan F. Jones, UK  
Palle E. Jorgensen, USA  
Vyacheslav Kalashnikov, Mexico  
Tamas Kalmar-Nagy, Hungary  
Tomasz Kapitaniak, Poland  
Julius Kaplunov, UK  
Haranath Kar, India  
K. Karamanos, Belgium  
Krzysztof Kecik, Poland  
Jean-Pierre Kenne, Canada  
Ch. M. Khaliq, South Africa  
Do Wan Kim, Republic of Korea  
Nam-Il Kim, Republic of Korea  
Jan Koci, Czech Republic  
Ioannis Kostavelis, Greece  
Sotiris B. Kotsiantis, Greece  
Manfred Krafczyk, Germany

Frederic Kratz, France  
Petr Krysl, USA  
Krzysztof S. Kulpa, Poland  
Shailesh I. Kundalwal, India  
Jurgen Kurths, Germany  
Cedrick A. K. Kwuimy, USA  
Kyandoghere Kyamakya, Austria  
Davide La Torre, Italy  
Risto Lahdelma, Finland  
Hak-Keung Lam, UK  
Giovanni Lancioni, Italy  
Jimmy Lauber, France  
Antonino Laudani, Italy  
Hervé Laurent, France  
Aimé Lay-Ekuakille, Italy  
Nicolas J. Leconte, France  
Dimitri Lefebvre, France  
Eric Lefevre, France  
Marek Lefik, Poland  
Yaguo Lei, China  
Kauko Leiviskä, Finland  
Thibault Lemaire, France  
Roman Lewandowski, Poland  
Chen-Feng Li, China  
Jian Li, USA  
En-Qiang Lin, USA  
Zhiyun Lin, China  
Peide Liu, China  
Peter Liu, Taiwan  
Wanquan Liu, Australia  
Bonifacio Llamazares, Spain  
Alessandro Lo Schiavo, Italy  
Jean Jacques Loiseau, France  
Francesco Lolli, Italy  
Paolo Lonetti, Italy  
Sandro Longo, Italy  
António M. Lopes, Portugal  
Sebastian López, Spain  
Pablo Lopez-Crespo, Spain  
Luis M. López-Ochoa, Spain  
Ezequiel López-Rubio, Spain  
Vassilios C. Loukopoulos, Greece  
Jose A. Lozano-Galant, Spain  
Helen Lu, Australia  
Gabriel Luque, Spain  
Valentin Lychagin, Norway  
Antonio Madeo, Italy

José María Maestre, Spain  
Alessandro Magnani, Italy  
Fazal M. Mahomed, South Africa  
Noureddine Manamanni, France  
Paolo Manfredi, Italy  
Didier Maquin, France  
Giuseppe Carlo Marano, Italy  
Damijan Markovic, France  
Francesco Marotti de Sciarra, Italy  
Rui Cunha Marques, Portugal  
Luis Martínez, Spain  
Rodrigo Martinez-Bejar, Spain  
Guiomar Martín-Herrán, Spain  
Denizar Cruz Martins, Brazil  
Benoit Marx, France  
Elio Masciari, Italy  
Franck Massa, France  
Paolo Massioni, France  
Alessandro Mauro, Italy  
Fabio Mazza, Italy  
Laura Mazzola, Italy  
Driss Mehdi, France  
Roderick Melnik, Canada  
Pasquale Memmolo, Italy  
Xiangyu Meng, USA  
Jose Merodio, Spain  
Alessio Merola, Italy  
Mahmoud Mesbah, Iran  
Luciano Mescia, Italy  
Laurent Mevel, France  
Mariusz Michta, Poland  
Aki Mikkola, Finland  
Giovanni Minafò, Italy  
Hiroyuki Mino, Japan  
Pablo Mira, Spain  
Dimitrios Mitsotakis, New Zealand  
Vito Mocella, Italy  
Sara Montagna, Italy  
Roberto Montanini, Italy  
Francisco J. Montáns, Spain  
Luiz H. A. Monteiro, Brazil  
Gisele Mophou, France  
Rafael Morales, Spain  
Marco Morandini, Italy  
Javier Moreno-Valenzuela, Mexico  
Simone Morganti, Italy  
Caroline Mota, Brazil

Aziz Moukrim, France  
Dimitris Mourtzis, Greece  
Emiliano Mucchi, Italy  
Josefa Mula, Spain  
Jose J. Muñoz, Spain  
Giuseppe Muscolino, Italy  
Marco Mussetta, Italy  
Hakim Naceur, France  
Alessandro Naddeo, Italy  
Hassane Naji, France  
M. Nakano-Miyatake, Mexico  
Keivan Navaie, UK  
AMA Neves, Portugal  
Luís C. Neves, UK  
Dong Ngoduy, New Zealand  
Nhon Nguyen-Thanh, Singapore  
Tatsushi Nishi, Japan  
Xesús Nogueira, Spain  
Ben T. Nohara, Japan  
Mohammed Nouari, France  
Mustapha Nourelfath, Canada  
Włodzimierz Ogryczak, Poland  
Roger Ohayon, France  
Krzysztof Okarma, Poland  
Mitsuhiro Okayasu, Japan  
Alberto Olivares, Spain  
Enrique Onieva, Spain  
Calogero Orlando, Italy  
A. Ortega-Moñux, Spain  
Sergio Ortobelli, Italy  
Naohisa Otsuka, Japan  
Erika Ottaviano, Italy  
Pawel Packo, Poland  
Arturo Pagano, Italy  
Alkis S. Paipetis, Greece  
Roberto Palma, Spain  
Alessandro Palmeri, UK  
Pasquale Palumbo, Italy  
Jürgen Pannek, Germany  
Elena Panteley, France  
Achille Paolone, Italy  
George A. Papakostas, Greece  
Xosé M. Pardo, Spain  
Vicente Parra-Vega, Mexico  
Manuel Pastor, Spain  
Petr Páta, Czech Republic  
Pubudu N. Pathirana, Australia

Surajit Kumar Paul, India  
Sitek Paweł, Poland  
Luis Payá, Spain  
Alexander Paz, USA  
Igor Pažanin, Croatia  
Libor Pekař, Czech Republic  
Francesco Pellicano, Italy  
Marcello Pellicciari, Italy  
Haipeng Peng, China  
Mingshu Peng, China  
Zhengbiao Peng, Australia  
Zhi-ke Peng, China  
Marzio Pennisi, Italy  
Maria Patrizia Pera, Italy  
Matjaz Perc, Slovenia  
A. M. Bastos Pereira, Portugal  
Ricardo Perera, Spain  
Francesco Pesavento, Italy  
Ivo Petras, Slovakia  
Francesco Petrini, Italy  
Lukasz Pieczonka, Poland  
Dario Piga, Switzerland  
Paulo M. Pimenta, Brazil  
Antonina Pirrotta, Italy  
Marco Pizzarelli, Italy  
Vicent Pla, Spain  
Javier Plaza, Spain  
Kemal Polat, Turkey  
Dragan Poljak, Croatia  
Jorge Pomares, Spain  
Sébastien Poncet, Canada  
Volodymyr Ponomaryov, Mexico  
Jean-Christophe Ponsart, France  
Mauro Pontani, Italy  
Cornelio Posadas-Castillo, Mexico  
Francesc Pozo, Spain  
Christopher Pretty, New Zealand  
Luca Pugi, Italy  
Krzysztof Puszynski, Poland  
Giuseppe Quaranta, Italy  
Vitomir Racic, Italy  
Jose Ragot, France  
Carlo Rainieri, Italy  
K. Ramamani Rajagopal, USA  
Ali Ramazani, USA  
Higinio Ramos, Spain  
Alain Rassineux, France

S.S. Ravindran, USA  
Alessandro Reali, Italy  
Jose A. Reinoso, Spain  
Oscar Reinoso, Spain  
Fabrizio Renno, Italy  
Nidhal Rezg, France  
Ricardo Riaza, Spain  
Francesco Riganti-Fulginei, Italy  
Gerasimos Rigatos, Greece  
Francesco Ripamonti, Italy  
Jorge Rivera, Mexico  
Eugenio Roanes-Lozano, Spain  
Bruno G. M. Robert, France  
Ana Maria A. C. Rocha, Portugal  
José Rodellar, Spain  
Luigi Rodino, Italy  
Rosana Rodríguez López, Spain  
Ignacio Rojas, Spain  
Alessandra Romolo, Italy  
Debasish Roy, India  
Gianluigi Rozza, Italy  
Rubén Ruiz, Spain  
Antonio Ruiz-Cortes, Spain  
Ivan D. Rukhlenko, Australia  
Mazen Saad, France  
Kishin Sadarangani, Spain  
Andrés Sáez, Spain  
Mehrddad Saif, Canada  
John S. Sakellariou, Greece  
Salvatore Salamone, USA  
Vicente Salas, Spain  
Jose Vicente Salcedo, Spain  
Nunzio Salerno, Italy  
Miguel A. Salido, Spain  
Roque J. Saltarén, Spain  
Alessandro Salvini, Italy  
Sylwester Samborski, Poland  
Ramon Sancibrian, Spain  
Giuseppe Sanfilippo, Italy  
Miguel A. F. Sanjuan, Spain  
Vittorio Sansalone, France  
José A. Sanz-Herrera, Spain  
Nickolas S. Sapidis, Greece  
Evangelos J. Sapountzakis, Greece  
Luis Saucedo-Mora, Spain  
Marcelo A. Savi, Brazil  
Andrey V. Savkin, Australia

Roberta Sburlati, Italy  
Gustavo Scaglia, Argentina  
Thomas Schuster, Germany  
Oliver Schütze, Mexico  
Lotfi Senhadji, France  
Junwon Seo, USA  
Joan Serra-Sagrsta, Spain  
Gerardo Severino, Italy  
Ruben Sevilla, UK  
Stefano Sfarra, Italy  
Mohamed Shaat, Egypt  
Mostafa S. Shadloo, France  
Leonid Shaikhet, Israel  
Hassan M. Shanechi, USA  
Bo Shen, Germany  
Suzanne M. Shontz, USA  
Babak Shotorban, USA  
Zhan Shu, UK  
Nuno Simões, Portugal  
Christos H. Skiadas, Greece  
Konstantina Skouri, Greece  
Neale R. Smith, Mexico  
Bogdan Smolka, Poland  
Delfim Soares Jr., Brazil  
Alba Sofi, Italy  
Francesco Soldovieri, Italy  
Raffaele Solimene, Italy  
Jussi Sopanen, Finland  
Marco Spadini, Italy  
Bernardo Spagnolo, Italy  
Paolo Spagnolo, Italy  
Ruben Specogna, Italy  
Vasilios Spitas, Greece  
Sri Sridharan, USA  
Ivanka Stamova, USA  
Rafał Stanisławski, Poland  
Florin Stoican, Romania  
Salvatore Strano, Italy  
Yakov Strelniker, Israel  
Guang-Yong Sun, China  
Ning Sun, China  
Sergey A. Suslov, Australia  
Thomas Svensson, Sweden  
Andrzej Swierniak, Poland  
Andras Szekrenyes, Hungary  
Kumar K. Tamma, USA  
Yang Tang, Germany

Hafez Tari, USA  
Alessandro Tasora, Italy  
Sergio Teggi, Italy  
Ana C. Teodoro, Portugal  
Tai Thai, Australia  
Alexander Timokha, Norway  
Gisella Tomasini, Italy  
Francesco Tornabene, Italy  
Antonio Tornambe, Italy  
Javier Martinez Torres, Spain  
Mariano Torrisi, Italy  
George Tsiatas, Greece  
Antonios Tsourdos, UK  
Federica Tubino, Italy  
Nerio Tullini, Italy  
Andrea Tundis, Italy  
Emilio Turco, Italy  
Vladimir Turetsky, Israel  
Mustafa Tutar, Spain  
Ilhan Tuzcu, USA  
Efstratios Tzirtzilakis, Greece  
Filippo Ubertini, Italy  
Francesco Ubertini, Italy  
Mohammad Uddin, Australia  
Hassan Ugail, UK  
Giuseppe Vairo, Italy  
Eusebio Valero, Spain  
Pandian Vasant, Malaysia  
Marcello Vasta, Italy  
Carlos-Renato Vázquez, Mexico  
Miguel E. Vázquez-Méndez, Spain  
Josep Vehi, Spain  
Martin Velasco Villa, Mexico  
K. C. Veluvolu, Republic of Korea  
Fons J. Verbeek, Netherlands  
Franck J. Vernerey, USA  
Georgios Veronis, USA  
Vincenzo Vespri, Italy  
Renato Vidoni, Italy  
V. Vijayaraghavan, Australia  
Anna Vila, Spain  
Rafael J. Villanueva, Spain  
Francisco R. Villatoro, Spain  
Uchechukwu E. Vincent, UK  
Gareth A. Vio, Australia  
Francesca Vipiana, Italy  
Stanislav Vítek, Czech Republic

Thuc P. Vo, UK  
Jan Vorel, Czech Republic  
Michael Vynnycky, Sweden  
Hao Wang, USA  
Liliang Wang, UK  
Shuming Wang, China  
Yongqi Wang, Germany  
Roman Wan-Wendner, Austria  
Jarosław Wąs, Poland  
P.H. Wen, UK  
Waldemar T. Wójcik, Poland  
Changzhi Wu, Australia  
Desheng D. Wu, Sweden  
Hong-Yu Wu, USA  
Yuqiang Wu, China  
Michalis Xenos, Greece  
Guangming Xie, China  
Xue-Jun Xie, China  
Gen Q. Xu, China  
Hang Xu, China  
Joseph J. Yame, France  
Xinggang Yan, UK  
Mijia Yang, USA  
Yongheng Yang, Denmark  
Luis J. Yebra, Spain  
Peng-Yeng Yin, Taiwan  
Qin Yuming, China  
Elena Zaitseva, Slovakia  
Arkadiusz Zak, Poland  
Daniel Zaldivar, Mexico  
Francesco Zammori, Italy  
Vittorio Zampoli, Italy  
Rafal Zdunek, Poland  
Ibrahim Zeid, USA  
Huaguang Zhang, China  
Kai Zhang, China  
Qingling Zhang, China  
Xian-Ming Zhang, Australia  
Xuping Zhang, Denmark  
Zhao Zhang, China  
Yifan Zhao, UK  
Jian G. Zhou, UK  
Quanxin Zhu, China  
Mustapha Zidi, France  
Gaetano Zizzo, Italy  
Zhixiang Zou, Germany

# Contents

## **Optimization Algorithms Combining (Meta)heuristics and Mathematical Programming and Its Application in Engineering**

Nibaldo Rodríguez , Abhishek Gupta, Paula L. Zabala, and Guillermo Cabrera-Guerrero   
Editorial (3 pages), Article ID 3967457, Volume 2018 (2018)

## **A Hybrid Simulated Annealing/Linear Programming Approach for the Cover Printing Problem**

Federico Alonso-Pecina  and David Romero   
Research Article (11 pages), Article ID 6193649, Volume 2018 (2018)

## **An Improved Lagrangian Relaxation Algorithm for the Robust Generation Self-Scheduling Problem**

Ping Che , Zhenhao Tang , Hua Gong , and Xiaoli Zhao  
Research Article (12 pages), Article ID 6303596, Volume 2018 (2018)

## **Multiple-Try Simulated Annealing Algorithm for Global Optimization**

Wei Shao  and Guangbao Guo   
Research Article (11 pages), Article ID 9248318, Volume 2018 (2018)

## **Reconstruction of Medical Images Using Artificial Bee Colony Algorithm**

Nur 'Afifah Rusdi , Zainor Ridzuan Yahya, Nurshazneem Roslan ,  
and Wan Zuki Azman Wan Muhamad   
Research Article (7 pages), Article ID 8024762, Volume 2018 (2018)

## **A Three-Term Conjugate Gradient Algorithm with Quadratic Convergence for Unconstrained Optimization Problems**

Gaoyi Wu, Yong Li, and Gonglin Yuan   
Research Article (15 pages), Article ID 4813030, Volume 2018 (2018)

## **Local Negative Base Transform and Image Scrambling**

Gangqiang Xiong , Shengqian Zheng, Jiang Wang, Zhanchuan Cai, and Dongxu Qi  
Research Article (18 pages), Article ID 8087958, Volume 2018 (2018)

## **A Projection Neural Network for Circular Cone Programming**

Yaling Zhang  and Hongwei Liu  
Research Article (12 pages), Article ID 4607853, Volume 2018 (2018)

## **A New Generalized Deep Learning Framework Combining Sparse Autoencoder and Taguchi Method for Novel Data Classification and Processing**

Ahmad M. Karim, Mehmet S. Güzel , Mehmet R. Tolun , Hilal Kaya, and Fatih V. Çelebi  
Research Article (13 pages), Article ID 3145947, Volume 2018 (2018)

## **An Objective Penalty Function-Based Method for Inequality Constrained Minimization Problem**

Shujun Lian , Sitong Meng, and Yiju Wang   
Research Article (7 pages), Article ID 7484256, Volume 2018 (2018)

## **Filled Function Method for Nonlinear Model Predictive Control**

Hajer Degachi , Bechir Naffeti, Wassila Chagra, and Moufida Ksouri  
Research Article (8 pages), Article ID 9497618, Volume 2018 (2018)

**Multiobjective Optimization for a Wireless Ad Hoc Sensor Distribution on Shaped-Bounded Areas**

Armando Céspedes-Mota , Gerardo Castañón , Alberto F. Martínez-Herrera ,  
and Leopoldo Eduardo Cárdenas-Barrón 

Research Article (22 pages), Article ID 7873984, Volume 2018 (2018)

**A Hybrid Epigraph Directions Method for Nonsmooth and Nonconvex Constrained Optimization via Generalized Augmented Lagrangian Duality and a Genetic Algorithm**

Wilhelm P. Freire, Afonso C. C. Lemonge , Tales L. Fonseca , and Hernando J. R. Franco 

Research Article (21 pages), Article ID 8750740, Volume 2018 (2018)

**A Modified Priority-Based Encoding for Design of a Closed-Loop Supply Chain Network Using a Discrete League Championship Algorithm**

Javid Ghahremani Nahr, Ramez Kian , and Hassan Rezazadeh

Research Article (16 pages), Article ID 8163927, Volume 2018 (2018)

**Heuristic Determination of Resolving Controls for Exact and Approximate Controllability of Nonlinear Dynamic Systems**

Asatur Zh. Khurshudyan 

Research Article (16 pages), Article ID 9496371, Volume 2018 (2018)

**An Adaptive Gradient Projection Algorithm for Piecewise Convex Optimization and Its Application in Compressed Spectrum Sensing**

Tianjing Wang , Hang Shen, Xiaomei Zhu, Guoqing Liu, and Hua Jiang

Research Article (9 pages), Article ID 9547934, Volume 2018 (2018)

**Allocation of Distributed Energy Systems at District-Scale over Wide Areas for Sustainable Urban Planning with a MILP Model**

Yeşim Ok  and Mehmet Atak

Research Article (14 pages), Article ID 4208415, Volume 2018 (2018)

**Patch Based Collaborative Representation with Gabor Feature and Measurement Matrix for Face Recognition**

Zhengyuan Xu, Yu Liu, Mingquan Ye, Lei Huang, Hao Yu, and Xun Chen 

Research Article (13 pages), Article ID 3025264, Volume 2018 (2018)

**Global Optimization for Generalized Linear Multiplicative Programming Using Convex Relaxation**

Yingfeng Zhao  and Ting Zhao

Research Article (8 pages), Article ID 9146309, Volume 2018 (2018)

**Local Search Algorithms for the Beam Angles' Selection Problem in Radiotherapy**

Guillermo Cabrera-Guerrero , Nivaldo Rodriguez, Carolina Lagos , Enrique Cabrera,  
and Franklin Johnson

Research Article (9 pages), Article ID 4978703, Volume 2018 (2018)

**Optimum Assembly Sequence Planning System Using Discrete Artificial Bee Colony Algorithm**

Özkan Özmen , Turgay Batbat , Tolgan Özen , Cem Sinanoğlu , and Ayşegül Güven 

Research Article (14 pages), Article ID 3407646, Volume 2018 (2018)

**Operation Optimization of Natural Gas Transmission Pipelines Based on Stochastic Optimization Algorithms: A Review**

Xia Wu , Changjun Li , Yufa He, and Wenlong Jia 

Review Article (18 pages), Article ID 1267045, Volume 2018 (2018)

**Grey Relational Bidirectional Projection Method for Multicriteria Decision Making with Hesitant Intuitionistic Fuzzy Linguistic Information**

Yuqi Zang , Xiaodong Zhao, Shiyong Li , and Adnan Nazir

Research Article (12 pages), Article ID 5691905, Volume 2018 (2018)

**An Improved Shuffled Frog Leaping Algorithm and Its Application in Dynamic Emergency Vehicle Dispatching**

Xiaohong Duan , Tianyong Niu, and Qi Huang

Research Article (34 pages), Article ID 7896926, Volume 2018 (2018)

**A Position-Level Global Optimization Inverse Kinematic Solution Algorithm for Dual Redundant Robots Based on Motion Characteristics**

Jingjie He 

Research Article (10 pages), Article ID 9167837, Volume 2018 (2018)

**Generalized Beamforming Design for Cooperative MIMO Multirelay Networks with Infinite Constraints and Imperfect CSI**

Haiyang Yu, Wei Duan , Qiang Sun, Xia Wang, Jue Wang, Jun Li, and Jaeho Choi 

Research Article (8 pages), Article ID 9797549, Volume 2018 (2018)

**A Conjugate Gradient Algorithm under Yuan-Wei-Lu Line Search Technique for Large-Scale Minimization Optimization Models**

Xiangrong Li , Songhua Wang, Zhongzhou Jin, and Hongtruong Pham

Research Article (11 pages), Article ID 4729318, Volume 2018 (2018)

## Editorial

# Optimization Algorithms Combining (Meta)heuristics and Mathematical Programming and Its Application in Engineering

**Nibaldo Rodríguez** <sup>1</sup>, **Abhishek Gupta**,<sup>2</sup>  
**Paula L. Zabala**,<sup>3</sup> and **Guillermo Cabrera-Guerrero** <sup>1</sup>

<sup>1</sup>Escuela de Ingeniería Informática, Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile

<sup>2</sup>Data Science and Artificial Intelligence Research Lab, School of Computer Science and Engineering,  
Nanyang Technological University, Singapore

<sup>3</sup>Universidad de Buenos Aires-CONICET, Argentina

Correspondence should be addressed to Nibaldo Rodríguez; [nibaldo.rodriguez@pucv.cl](mailto:nibaldo.rodriguez@pucv.cl)

Received 23 August 2018; Accepted 23 August 2018; Published 18 September 2018

Copyright © 2018 Nibaldo Rodríguez et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Complex optimization problems can be tackled by means of mathematical programming methods as well as by means of (meta)heuristic methods. On the one hand, mathematical programming methods give us a guarantee of optimality while (meta)heuristic methods do not. On the other hand, heuristic methods can handle large and complex optimization problems while mathematical programming methods tend to fail as the size of the optimization problem increases. Thus, it makes sense to combine these two strategies to obtain better solutions to the problem that is being addressed. During the last two decades or so, algorithms that either include mathematical programming solvers into (meta)heuristic frameworks or include (meta)heuristic concepts within mathematical programming methods have demonstrated to be very effective in solving large complex optimization problems. These hybrid algorithms are also called *matheuristics*. These kinds of algorithms have been successfully applied to a wide range of optimization problems arising in engineering.

In this special issue, we aimed to highlight those new approaches that take advantage of the main features of both mathematical programming and heuristic algorithms to solve challenging optimization problems. We received 129 submissions from all around the world. From these, only 25 articles were accepted after a rigorous peer-reviewed process, that is, a 19% acceptance rate. In the following, we briefly introduce each paper and try to organise them based on their main focus.

Lagrangian relaxation (LR) based algorithms were one of the topics we include in this special issue. In the paper “An Improved Lagrangian Relaxation Algorithm for the Robust Generation Self-Scheduling Problem” P. Che et al. addressed the robust generation self-scheduling problem under electricity price uncertainty which is reformulated as a MINLP problem. Authors combine an LR approach and linear programming algorithms to approximately solve this problem. LR is also considered in the paper “A Hybrid Epigraph Directions Method for Nonsmooth and Nonconvex Constrained Optimization via Generalized Augmented Lagrangian Duality and a Genetic Algorithm” by W. P. Freire et al. In this case, authors combine the generalized augmented Lagrangian duality approach and genetic algorithms. The proposed approach is applied to a set of optimization problems from mathematics and mechanical engineering.

Global optimization algorithms were also considered within this special issue. In the paper “Multiple-Try Simulated Annealing Algorithm for Global Optimization” W. Shao and G. Guo propose an algorithm that combines simulated annealing and the multiple-try metropolis algorithm. The proposed algorithm has a rapid decreasing schedule while guaranteeing global optimum values. In the paper “A Modified Priority-Based Encoding for Design of a Closed-Loop Supply Chain Network Using a Discrete League Championship Algorithm” by J. G. Nahr et al., a novel league championship algorithm (LCA) with a modified priority-based encoding is applied to find a near-optimal solution.

Authors propose new operators for the LCA to search the discrete space. Their algorithm is applied to a very difficult problem in logistics. In the paper “Global Optimization for Generalized Linear Multiplicative Programming Using Convex Relaxation” Y. Zhao and T. Zhao present a simple yet efficient algorithm that combines a new convex relaxation method and the well-known branch and bound scheme with some accelerating techniques. They applied their algorithm to the generalized linear multiplicative programming problem. In “An Improved Shuffled Frog Leaping Algorithm and Its Application in Dynamic Emergency Vehicle Dispatching” X. Duan et al. present an improved shuffled frog leaping algorithm which uses the probability model of estimation of distribution algorithm to avoid locally optimal solutions. The proposed algorithm is applied to a routing problem arising in health-systems logistics. In the paper “Optimum Assembly Sequence Planning System Using Discrete Artificial Bee Colony Algorithm” Ö. Özmen et al. present a computer program developed based on a matrix-based approach and the discrete artificial bee colony algorithm, which determines the optimum assembly sequence among numerous feasible assembly sequences (FAS). The assembly sequences of three-dimensional parts are first coded using the matrix-based methodology and the resulting FAS are assessed and the optimum assembly sequence is selected according to the assembly time optimization criterion using the artificial bee algorithm. The paper “A Position-Level Global Optimization Inverse Kinematic Solution Algorithm for Dual Redundant Robots Based on Motion Characteristics” by J. He presents an inverse kinematics optimization algorithm for PRRPR-S redundant degrees of freedom camera robot. This paper analyses the motion characteristics, in Genetic Mix method, and then proposes a Simplify Mix (SM) method that, according to the authors, can stably converge to the global optimal solution in a short time. In “An Objective Penalty Function-Based Method for Inequality Constrained Minimization Problem” by S. Lian et al., a method to globally solve inequality constrained minimization problem based on penalty functions is presented. Finally, a review on global optimization methods to address problems in natural gas transmission pipelines is presented in “Operation Optimization of Natural Gas Transmission Pipelines Based on Stochastic Optimization Algorithms: A Review” by X. Wu et al.

Three articles included in this special issue present novel gradient-based strategies. The paper “A Three-Term Conjugate Gradient Algorithm with Quadratic Convergence for Unconstrained Optimization Problems” by G. Wu et al. presents a three-term WYL conjugate gradient algorithm for which both global and linear convergence is proved. In “An Adaptive Gradient Projection Algorithm for Piecewise Convex Optimization and Its Application in Compressed Spectrum Sensing” T. Wang et al. propose an Adaptive Gradient Projection (AGP) algorithm to solve the piecewise convex optimization in signal sparse representation. To find a sparser solution, AGP provides an adaptive step size to move the iteration solution out of the attraction basin of a suboptimal sparse solution and enter the attraction basin of a sparser solution. A modified Hestenes and Stiefel conjugate gradient algorithm is presented in “A Conjugate Gradient

Algorithm under Yuan-Wei-Lu Line Search Technique for Large-Scale Minimization Optimization Models” by X. Li et al. Authors claim that their algorithm has global convergence for nonconvex functions and that the new search direction possesses not only a sufficient descent property but also a trust region feature.

Papers on multiobjective optimization were also included in this special issue. In the paper “Multiobjective Optimization for a Wireless Ad Hoc Sensor Distribution on Shaped-Bounded Areas” A. Céspedes-Mota et al. present multiobjective differential evolution algorithm to solve a problem arising in networks optimization. The differential evolution algorithm is combined with the Prim-Dijkstra and the Hungarian algorithms to improve its efficiency. Authors define objectives as maximising coverage area of the network and minimizing energy consumption. In “Grey Relational Bidirectional Projection Method for Multicriteria Decision Making with Hesitant Intuitionistic Fuzzy Linguistic Information”, Y. Zang et al. propose a comparison method of hesitant intuitionistic fuzzy linguistic term sets. Based on this comparison method, the authors propose the grey relational bidirectional projection method for dealing with MCDM problems. The vector of weights is computed by solving a nonlinear optimization model.

Algorithms for nonlinear optimization problems have been also published in this special issue. In “Filled Function Method for Nonlinear Model Predictive Control”, H. Degachi et al. present a framework to solve a nonlinear model predictive control for the Hammerstein model. They use a filled function to approach the global optimum of the problem and then find it using local search strategies. In the paper “Heuristic Determination of Resolving Controls for Exact and Approximate Controllability of Nonlinear Dynamic Systems”, A. Zh. Khurshudyan aims to determine, heuristically, control functions providing exact and approximate controllability of dynamic systems with nonlinear state constraints. Using a recently developed approach based on Green’s function method, the controllability analysis of nonlinear dynamic systems, in general, is reduced to nonlinear integral constraints with respect to the control function. In the paper “Local Search Algorithms for the Beam Angles’ Selection Problem in Radiotherapy” G. Cabrera-Guerrero et al. deal with a nonlinear problem arising in radiotherapy for cancer treatment. In their framework, the authors combine local search strategies and an interior point method.

Image processing algorithms are also part of this special issue. In “Reconstruction of Medical Images Using Artificial Bee Colony Algorithm” N. A. Rusdi et al. combine an artificial bee colony algorithm and the Douglas Peucker algorithm to reconstruct medical images. In “Patch Based Collaborative Representation with Gabor Feature and Measurement Matrix for Face Recognition”, Z. Xu et al. propose a patch based collaborative representation method for face recognition via Gabor feature and measurement matrix. Authors claim that their method can solve the problem of the lack of accuracy for the linear representation of the small sample size in face recognition. In “Local Negative Base Transform and Image Scrambling” by G. Xiong et al. a new class of scrambling algorithms for image encryption and hiding is

obtained by exploiting negative integer as the base of number representation to express the natural numbers.

Finally, other more-specific topics were also included in this special issue. For instance, neural networks for circular cone programming are presented in “A Projection Neural Network for Circular Cone Programming” by Y. Zhang and H. Liu. A deep learning classification algorithm is introduced in “A New Generalized Deep Learning Framework Combining Sparse Autoencoder and Taguchi Method for Novel Data Classification and Processing” by A. M. Karim et al. A MILP problem arising in energy systems is modelled and solved in “Allocation of Distributed Energy Systems at District-Scale over Wide Areas for Sustainable Urban Planning with a MILP Model” by Y. Ok and M. Atak. Finally, a cooperative MIMO multirelay network is implemented in “Generalized Beamforming Design for Cooperative MIMO Multirelay Networks with Infinite Constraints and Imperfect CSI” by H. Yu et al.

As guest editors, we deeply hope that this special issue can be a step forward in the development of optimization algorithms that combine (meta)heuristics and mathematical programming.

### **Conflicts of Interest**

The authors declare that there are no conflicts of interest regarding the publication of this article.

*Nibaldo Rodríguez*  
*Abhishek Gupta*  
*Paula L. Zabala*  
*Guillermo Cabrera-Guerrero*

## Research Article

# A Hybrid Simulated Annealing/Linear Programming Approach for the Cover Printing Problem

Federico Alonso-Pecina <sup>1</sup> and David Romero <sup>2</sup>

<sup>1</sup>Universidad Autónoma del Estado de Morelos, 62209 Cuernavaca, Morelos, Mexico

<sup>2</sup>Instituto de Matemáticas, Universidad Nacional Autónoma de México, 62210 Cuernavaca, Morelos, Mexico

Correspondence should be addressed to Federico Alonso-Pecina; federicoalonsopecina@hotmail.com

Received 30 January 2018; Revised 25 June 2018; Accepted 16 July 2018; Published 27 August 2018

Academic Editor: Guillermo Cabrera-Guerrero

Copyright © 2018 Federico Alonso-Pecina and David Romero. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The NP-hard cover printing problem addressed here consists in determining the number and composition of equal size impression grids, as well as the number of times each grid is printed, in order to fulfill the demand of different book covers at minimum total printing cost. The considered costs come from printing sheets and for composing grids. Thus, to deal with this combinatorial optimization problem we investigated two heuristics: one combines simulated annealing and linear programming techniques and the other is a hybrid of Tabu Search and an ad hoc procedure. Through intensive testing on available instances, these algorithms proved to be superior to previous approaches.

## 1. Introduction

This paper deals with a scheduling problem motivated by the printing industry. We will refer to it as the cover printing problem, although including its variants is also known as the job splitting [1], the label printing [2, 3], and the advertisement printing [4] problem.

Central to the process of producing  $m$  types of equal size book covers is to print an impression grid (or press, template, master, etc.) composed of a set of  $t$  equal size plates, where some of them may be identical. To fulfill a given demand of book covers interrelated decisions must be made on the number and composition of impression grids (grids, for short) and on the amount of imprints made with each. One imprint of a grid produces one large printed sheet of paper which, once cut into  $t$  parts, yields  $t$  cover copies.

Two constants are relevant to the decision process: the cost for printing one sheet  $C_1$  and the cost for composing one grid  $C_2$ . Thus, the cover printing problem (CPP) consists

in determining —at minimum total cost—the number and composition of grids, and the number of imprints made with each, so as to fulfill the required demand for each type of cover.

*A Toy Example.* Let  $m = 5$  be the number of covers,  $t = 4$  the grid size, and  $(12, 18, 44, 47, 79)$  the required demand vector, for a total of 200 copies. A feasible solution of the CPP is found by composing  $k = 3$  grids as described by the  $(m \times k)$ -matrix  $A = (a_{ij})$  of (1) (matrices  $B$  to be discussed later), where  $a_{ij}$  is the number of plates of cover  $i$  in grid  $j$ , and  $\sum_{i=1}^m a_{ij} = t$ , for  $j = 1, 2, 3$ . Then, if grids 1, 2, 3, are printed 29, 16, and 22 times, respectively, the covers' demand is satisfied. Observe that  $29 + 16 + 22 = 67$  imprints are made, producing  $67t = 268$  copies, with  $268 - 200 = 68$  wasted, namely, 34% of wastage. Further,  $67C_1 + 3C_2$  is the total cost of this solution, whose optimality clearly depends on the relative values of  $C_1$  and  $C_2$ .

Matrices corresponding to selected solutions of the toy example.

$$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \\ 0 & 3 & 0 \\ 2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 & 0 \\ 0 & 2 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 2 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 2 \\ 1 & 0 \\ 1 & 1 \\ 2 & 0 \end{pmatrix} \quad (1)$$

$A \qquad B_0 \qquad \hat{B} \qquad B^*$

The computational complexity of the CPP is not known to date. However, some special cases have been polynomially solved by Ekici et al. [1] and Romero & Alonso-Pecina [5]. On the other hand, the CPP is strongly *NP*-hard when  $C_1 \gg C_2$  [1]. Besides a handful of graduate thesis (namely, four Belgian graduate thesis cited in [6–8], not available to the authors, and the Latin American graduate thesis by Calderón [9] and Carrera [10]) few internationally published papers have studied heuristic approaches for this highly combinatorial problem and its variants. These approaches are assorted, going from simulated annealing (Teghem et al. [7], Yiu et al. [3]) to GRASP (Tuytens & Vandaele [8, 11]), as well as from evolutionary algorithms (Elaoud et al. [6], Hsieh et al. [2]) to ad hoc heuristics (Ekici et al. [1], Mohan et al. [4], and Romero & Alonso-Pecina [5]).

Several CPP variants have been investigated. Romero & Alonso-Pecina [5] proposed to add a Hamiltonian path procedure to their ad hoc heuristic when there is also a cost for each produced plate. Other variants consist in the inclusion of additional constraints: while Mohan et al. [4] tackle the case of lower and upper bounds on the number of imprints made with each grid, Yiu et al. [3] and Mohan et al. [4] assume the number of grids as prescribed. Finally, Yiu et al. [3], Hsieh et al. [2], and Tuytens & Vandaele [11] consider the constraint that all copies of any cover should be produced by one single grid.

The aim of this paper is to propose two new heuristics to successfully approach the solution of the CPP. One, hybridized with linear programming, has the simulated annealing technique as motif; the other strategically combines Tabu Search and ad hoc procedures. We are witnessing a recent, interesting trend about the hybridization of metaheuristics and other techniques for optimization, like local search or exact algorithms (see the excellent survey by Blum et al. [12], as well as Gomes & Oliveira [13], Caserta & Voss [14], Popović et al. [15], López-Ibáñez et al. [16], and Mestria [17], among others). Such types of hybridization are intended to take advantage of strengths from each algorithm to improve the algorithms' performance when solving different combinatorial optimization problems.

This paper is organized as follows. Section 2 provides an integer programming formulation of the CPP. Then, our simulated annealing programming and Tabu Search/ad hoc heuristics are described in Sections 3 and 4, respectively. Computationally tested on the 88 instances available in the specialized literature, our heuristics, taken together, improved

the known best solution of more than two-thirds of them; this is discussed in Section 5. Finally, Section 6 contains our conclusions.

## 2. Integer Programming Formulation

The following mathematical formulation of the cover printing problem improves on the one proposed by us in [5]. Let  $M = \{1, \dots, m\}$ ,  $(d_1, \dots, d_m)$  and  $t, C_1$ , and  $C_2$  be the set of covers, the requirements vector, the grid size, the cost for printing one sheet, and the cost for composing one grid, respectively. Without loss of generality assume  $0 < d_1 \leq d_2 \leq \dots \leq d_m$ . Letting  $N = \{1, \dots, n\}$  be the set of possible impression grids, where  $n$  can be computed as  $n = \binom{m+t-1}{t}$ , consider the  $m$ -by- $n$  matrix  $A = (a_{ij})$  where  $a_{ij}$  is the number of plates of cover  $i$  in grid  $j$ . Obviously each  $A$  column sums up to  $t$ .

Thus, the CPP can be posed as one problem of integer programming:

$$\begin{aligned} (\min) \quad & C_1 \sum_{j \in N} x_j + C_2 \sum_{j \in N} w_j \\ \text{subject to} \quad & \sum_{j \in N} x_j a_{ij} \geq d_i \quad i \in M, \quad (1) \\ & x_j \leq w_j d_m \quad j \in N, \quad (2) \\ & x_j \geq 0 \text{ and integer} \quad j \in N, \quad (3) \\ & w_j \in \{0, 1\} \quad j \in N, \quad (4) \end{aligned} \quad (P)$$

where  $x_j$  and  $w_j$  are the decision variables, with  $w_j = 1$  if and only if grid  $j$  is selected and  $x_j$  being the number of its imprints. Problem (P) is intractable since the number  $n$  of possible grids is exponential. For the toy example of Section 1 we have  $n = 70$ , yielding 140 integer variables and  $3n + m = 215$  constraints.

Let  $B = (b_{ij})$  be an  $m$ -by- $k$  submatrix of  $A$ , whose columns have been reindexed,  $1, \dots, k$ , and consider the integer programming problem

$$\begin{aligned} (\min) \quad & \sum_{j=1}^k x_j \\ \text{subject to} \quad & \sum_{j=1}^k x_j b_{ij} \geq d_i \quad i \in M, \quad (5) \end{aligned}$$

$$x_j \geq 0 \quad j = 1, \dots, k, \quad (6)$$

$$x_j \text{ integer} \quad j = 1, \dots, k. \quad (7) \quad (Q)$$

Obviously, any feasible solution of (Q) can be extended to an also feasible solution of problem (P) by making  $x_j = 0$  for all  $A$  columns not belonging to  $B$ , and  $w_j = 1 \iff x_j > 0$  for  $j \in N$ . In the sequel we assume this extension is implicitly made.

It has been remarked [1, 5] that no feasible solution to the CPP exists with less than  $\lceil m/t \rceil$  grids, and there is no optimal solution to the CPP with more than  $m$  grids. Hence, define  $\Omega$  as the set of  $m$ -by- $k$  submatrices of  $A$  without zero rows, with  $k$  satisfying  $\lceil m/t \rceil \leq k \leq m$ .

Now, for  $B \in \Omega$ , denote by  $X'(B)$  the optimal solution of problem (Q), and denote by  $X(B)$  the rounded up, optimal solution of problem (Q) when solved without integrality constraints (7). From the previous paragraph not only the pair  $(B, X(B))$  is a feasible solution to the CPP, but there exists  $B^* \in \Omega$  such that  $(B^*, X'(B^*))$  is an optimal solution to the CPP.

In view of the high demand values normally found in the internationally published instances (most demand values are between  $10^4$  and  $10^5$  [6–8]), in our approach we opted to only deal with the CPP solutions of type  $(B, X(B))$ , considering its inherent error immaterial.

### 3. Simulated Annealing/Linear Programming Approach

In this section we present the first of the two heuristic approaches we propose for the solution of the cover printing problem. This heuristic—we call it SALP—is centered in the simulated annealing concept hybridized with linear programming techniques. A recent, good overview of simulated annealing can be found in [18], and for linear programming we refer the reader to [19]. SALP is similar in essence to the Teghem et al. proposal [7]; however, arising from the final remarks of Section 2, it differs in regard to the neighborhood definition. Namely, for a given matrix  $B = (b_{ij}) \in \Omega$ , the *neighborhood* of solution  $S = (B, X(B))$  is the set  $\mathcal{N}(S)$  containing every solution  $(\check{B}, X(\check{B}))$  such that  $\check{B} \in \Omega$  is obtainable from  $B$  by either

- (i) adding 1 to  $b_{u,w}$  and subtracting 1 from  $b_{v,w}$ , for some indexes  $u, v, w$ , or
- (ii) adding 1 to  $b_{u,w}$  and  $b_{v,x}$  and subtracting 1 from  $b_{u,x}$  and  $b_{v,w}$ , for some indexes  $u, v, w, x$ .

Thus, SALP consists in performing the procedure ANNEAL followed by the procedure REFINEMENT, the former drawing upon procedures LINPROG, RAND, INITIALSOL, RESIZE, and NEIGHBOR. Each is described below. We thoroughly assume all random numbers are generated with uniform probability distribution.

LINPROG( $B$ ). Given an  $m$ -by- $k$  matrix  $B = (b_{ij}) \in \Omega$ , this procedure consists of a standard implementation of the Simplex method for linear programming. It is aimed at finding  $X(B) = (x_1, \dots, x_k)$  as defined in Section 2, namely, the

rounded up optimum of the linear programming relaxation of problem (Q). Hence, the solution  $S = (B, X(B))$  is delivered, as well as its cost  $z(S) = C_1 \sum_{j=1}^k x_j + C_2 \times n_{>0}$ , where  $n_{>0}$  stands for the number of positive entries in  $X(B)$ . Recall that  $C_1$  and  $C_2$  are the unit costs for sheet printing and grid composing, respectively. Although the Simplex method has computational complexity  $O\left(\binom{k+m}{m}\right)$ , its general good behavior is a well known fact: the observed average of iterations is roughly  $1.5m$ .

INITIALSOL. An initial, feasible solution to CPP is built in two steps. First, an  $m$ -by- $m$  matrix  $B_o = (b_{ij}) \in \Omega$  is constructed, where  $b_{ij}$  is randomly generated in the set  $\{0, 1, \dots, t\}$ , for  $(i, j) \in M \times M$ , ensuring that  $B_o$  has no zero rows, and each of its columns sums up to  $t$ . It proceeds as follows: Lines (1) to (10) build an integer, nonnegative matrix with no zero rows, whose column sums are equal  $t$  or less; then lines (11) to (20) ensure that each column sums up to  $t$ . To end, procedure LINPROG( $B_o$ ) is called to determine  $X(B_o)$ , completing the construction of the initial solution  $S_o = (B_o, X(B_o))$ , and its cost  $z(S_o)$ .

The overall complexity of this procedure is  $O\left(m^2 + \binom{k+m}{m}\right)$ , namely, the same as that for LINPROG( $B$ ) (Algorithm 1).

RAND. This function delivers a random real number in the interval  $(0, 1)$ .

NEIGHBOR( $S$ ). Using the neighborhood  $\mathcal{N}$  as defined in the first paragraph of this section, this procedure—of computational complexity  $O(mk)$ —randomly produces a feasible solution in  $\mathcal{N}(S)$ .

RESIZE. This procedure takes a solution  $S = (B, X(B))$  as input and produces a solution  $S' = (B', X(B'))$  as follows. First  $B' \leftarrow B$  and let  $q$  be the number of zeros in  $X(B')$ . If  $q \geq 2$  then delete the first  $q - 1$  columns  $j$  such that  $x_j = 0$ . Otherwise, if  $q = 0$  and  $B'$  has less than  $m$  columns then a nonnegative integer column  $(\alpha_1, \dots, \alpha_m)$  is added to it, where

- (i)  $\sum_{i=1}^m \alpha_i = t$ ,
- (ii)  $|\alpha_r - \alpha_s| \leq 1$ , for all  $r, s \in [1, m]$ , and
- (iii)  $d_r < d_s$  implies  $\alpha_r \leq \alpha_s$ , for all  $r, s \in [1, m]$ .

To end, procedure LINPROG( $B'$ ) is called to determine  $X(B')$  and cost  $z(S')$ , where  $S' = (B', X(B'))$ . The computational complexity of this procedure is clearly  $O(m^2)$ . The rationale for this procedure is to allow the number of grills in the solutions found during the search to increase.

ANNEAL. This procedure basically consists of standard simulated annealing iterations where parameters  $\varphi, \lambda, T_o, T_f$ , denote, resp., the cooling factor, the internal cycle length, the initial temperature, and the final temperature (Algorithm 2).

The complexity of the algorithm is given by the number of times the two cycles are performed times the complexity of LINPROG( $S$ ). While the first cycle is performed until  $T$  reaches  $T_f$ , say,  $\kappa$  times, where  $\kappa = (\ln(T_f) - \ln(T_o)) / \ln(\theta)$  [20], the

```

(1)  $B_{ij} \leftarrow 0, \forall i, j \in M \times M;$ 
(2)  $i \leftarrow 1;$ 
(3) While  $i \leq m$  do
(4)    $ban \leftarrow false;$ 
(5)   While  $ban \neq true$  do
(6)      $j \leftarrow \lfloor RAND \times m \rfloor + 1;$ 
(7)     If  $\sum_{i \in M} B_{ij} < t$  then  $B_{ij} \leftarrow B_{ij} + 1; ban \leftarrow true;$ 
(8)     EndWhile;
(9)      $i \leftarrow i + 1;$ 
(10) EndWhile;
(11)  $j \leftarrow 1;$ 
(12) While  $j \leq m$  do
(13)    $cont \leftarrow \sum_{i \in M} B_{ij};$ 
(14)   While  $cont < t$  do
(15)      $i \leftarrow \lfloor RAND \times m \rfloor + 1;$ 
(16)      $B_{ij} \leftarrow B_{ij} + 1;$ 
(17)      $cont \leftarrow cont + 1;$ 
(18)   EndWhile
(19)    $j \leftarrow j + 1;$ 
(20) EndWhile;
(21)  $[z_0, X(B_0)] \leftarrow \text{LINPROG}(B_0).$ 

```

ALGORITHM 1: Procedure INITIALSOL.

```

INITIALSOL; [to generate  $S_0 = (B_0, X_0)$  and  $z(S_0)$ ]
 $S \leftarrow S_0;$ 
 $z_{\min} \leftarrow z(S_0);$ 
 $\theta \leftarrow T_0;$  [variable  $\theta$  holds the system temperature]
 $S_{act} \leftarrow \text{RESIZE}(S);$ 
 $S_{best} \leftarrow S_{act};$ 
While  $\theta \geq T_f$  do
   $w \leftarrow 0;$ 
  While  $w \leq \lambda$  do
     $S \leftarrow \text{NEIGHBOR}(S_{act});$ 
     $\text{LINPROG}(S);$ 
     $\delta \leftarrow z(S) - z(S_{act});$ 
    If  $\delta < 0$  or  $\text{RAND} < e^{-\delta/\theta}$  then
       $S_{act} \leftarrow S; S_{act} \leftarrow \text{RESIZE}(S);$ 
    EndIf
     $w \leftarrow w + 1;$ 
    If  $z(S_{act}) < z(S_{best})$  then
       $S_{best} \leftarrow S_{act}; w \leftarrow 0;$ 
    EndIf
  EndWhile
   $\theta \leftarrow \varphi\theta;$ 
EndWhile.

```

ALGORITHM 2: Procedure ANNEAL.

inner cycle is performed  $m \times t$  times, and  $\text{LINPROG}(S)$  has complexity  $\binom{k+m}{m}$ . So, the total time is  $O(\kappa m t \binom{k+m}{m})$ .

Above, we have pointed out that our procedure SALP is similar in essence to that of Teghem et al. [7]. Besides the fact that we use a different and larger neighborhood, the main difference between both approaches is that, along the search, in ours the number of grids equals the number of nonzero variables in the optimal solutions given by linear programming (LP), while in theirs it is just the opposite: the number of grids is determined by an element of the

neighborhood employed in simulated annealing (SA). We mention that [7] is one of the pioneering papers on the CPP. Among the small instances I001–I004 considered therein, the authors found the optimal solution of two of them; while in our present approach as well as in those in [5, 8, 11] the four instances were solved to optimality.

REFINEMENT. A local search procedure (LS) attempts to deterministically improve a given solution  $S^*$ . LS considers all the elements of  $\mathcal{N}(S^*)$  one at a time; if none improves on cost  $z(S^*)$ , the procedure stops; otherwise  $S^*$  is replaced by the first found solution of  $\mathcal{N}(S^*)$  with lower cost than  $S^*$ , and LS starts anew. The complexity of LS is that of entirely exploring the neighborhood  $\mathcal{N}(S^*)$  multiplied by  $\binom{k+m}{m}$ , namely,  $O(m^2 k^2 \times \binom{k+m}{m})$ .

Now, let us return to the toy example of Section 1 and consider the matrices  $B_0, \hat{B}$ , and  $B^*$ , in (1). Also, let  $C_1 = 1$  and  $C_2 = 100$ , and assume INITIALSOL generates  $S_0 = (B_0, X(B_0))$ , where  $X(B_0) = (11, 36, 0, 0, 7)$  is obtained by LINPROG. This is to say, solution  $S_0$  consists of three grids (columns 1, 2, and 5 of  $B_0$ ), imprinted 11, 36, and 7 times, respectively, yielding  $z(S_0) = 354$ .

Taking  $B_0$  as input, the procedure RESIZE produces  $\hat{B}$ ,  $X(\hat{B}) = (10, 20, 4, 9)$ , and  $z(\hat{B}) = 443$ . Then, from  $\hat{B}$  and after a few iterations, SALP successively obtains  $B^*, X(B^*) = (44, 12)$ ,  $S^* = (B^*, X(B^*))$ , and  $z(S^*) = 256$ , which turns out to be the global optimum of the toy example.

#### 4. Tabu Search/ $\mathcal{H}$ Approach

Our second approach consists of the hybridization of two heuristics: the Tabu Search technique (TS) and an ad hoc procedure called here  $\mathcal{H}$ , whose aim is to provide an initial solution to TS.

```

(1)  $S \leftarrow \mathcal{H}()$ ;
(2)  $z_{min} \leftarrow z_{act} \leftarrow z(S)$ ;
(3)  $count \leftarrow 0$ ;  $S_{best} \leftarrow S$ ;  $iter \leftarrow 0$ ;
(4)  $\mathcal{T} \leftarrow \emptyset$ ;  $S_{act} \leftarrow \text{RESIZE}(S)$ ;
(5) While  $count \leq \psi$  do
(6)    $S_{new} \leftarrow \text{NEIGH}(S)$ ;
(7)    $z_{new} \leftarrow z(S_{new})$ ;
(8)   If  $z_{new} < z_{min}$  then
(9)      $S_{best} \leftarrow S \leftarrow S_{new}$ ;
(10)     $z_{min} \leftarrow z \leftarrow z_{new}$ ;
(11)     $count \leftarrow 0$ ;
(12)   else
(13)      $S_{new} \leftarrow \text{NOTTABUNEIGH}(S)$ ;
(14)      $z \leftarrow z(S_{new})$ ;  $S \leftarrow S_{new}$ ;
(15)      $\text{UPDATE}(\mathcal{T})$ ;
(16)      $count \leftarrow count + 1$ ;
(17)   EndIf
(18)    $iter \leftarrow iter + 1$ ;
(19) EndWhile
(20) return  $B_{best}$ ;

```

ALGORITHM 3: Procedure TS/ $\mathcal{H}$ .

A thorough description of the  $\mathcal{H}$  heuristic can be found in [5]; however, we find it convenient to sketch here its main idea: at each of the outer iterations a solution is sought that minimizes the number of required grids subject to the constraint that for any cover the wastage is lower or equal to a given upper bound. This upper bound is initialized with the maximum demand  $d_m$  and is monotonically decreased in each outer iteration as long as new solutions are found or until it reaches zero. In the end, the solution with the minimum cost among those produced in the outer iterations is retained.

Successfully used to tackle combinatorial optimization problems galore (see, for example, [21] and the references therein), Tabu Search [22] is based on a tabu list of forbidden movements that is updated during the iterations to avoid cycling and get stuck in local minima. Our approach TS/ $\mathcal{H}$  considered an enlargement of the neighborhood  $\mathcal{N}$ —defined in Section 3—so as to include the feasible solutions that can be obtained by deleting any one grid; we denote by  $\mathcal{N}_{TS}$  this neighborhood. In the pseudocode TS/ $\mathcal{H}$  below the procedures NEIGH(S) and NOTTABUNEIGH(S) produce the best feasible solution, respectively, in  $\mathcal{N}_{TS}$  and in  $\mathcal{N}_{TS} \setminus \mathcal{T}$  (Algorithm 3).

In line (1) a solution  $S$  is heuristically produced by procedure  $\mathcal{H}$ , which is hereupon used as initial solution by the Tabu Search algorithm. In lines (2–4) several variables are initialized, among them is a nonnegative integer  $m$ -by- $m$  matrix  $\mathcal{T} = \{\mathcal{T}_{ij}\}$  to represent the tabu list (here,  $\emptyset$  denotes the zero matrix). The main cycle (5–19) is performed a maximum of  $\psi$  times without improving the best solution found. In our experiments we set  $\psi = 100$ . The procedure NEIGH (line 6) obtains a solution (using first-best strategy) in the neighborhood  $\mathcal{N}_{TS}$  improving the best solution found so far (if such solution exists). The solution  $S_{new}$  is evaluated in lines (8–11); then, if  $S_{new}$  improves on the so far best solution found, the latter is updated and the integer  $count$  is reset (lines 9–11); otherwise (lines 13–17) the best solution without tabu status is chosen as  $S_{new}$ .

In line 15 the function UPDATE( $\mathcal{T}$ ) makes  $\mathcal{T}_{ij} \leftarrow iter + 1 + \text{RAND}() * (m - 1)$  for each entry  $(i, j)$  of  $B$  that has been modified (recall that  $B$  is part of the solution  $S_{new}$ , as defined in Section 3). Thus, the entry  $(i, j)$  of matrix  $B$  cannot be changed as long as  $\mathcal{T}_{ij} \geq iter$ ; namely, it gets tabu status for a number of iterations randomly chosen in the range  $[1, m]$ .

The complexity of procedure TS/ $\mathcal{H}$  is the complexity of  $\mathcal{H}$  plus the TS complexity. From computational experimentation we found that, in general,  $3 \times \psi$  is a reasonable upper bound for the number of times the main cycle of TS is performed. As  $3 \times \psi$  is a constant, the complexity of Tabu Search comes to that of LS:  $O(m^2 k^2 \times \binom{k+m}{m})$ . As this dominates that of  $\mathcal{H}$ ,  $O(d_1 m^2 t \log(mt))$  [5], the overall complexity of TS/ $\mathcal{H}$  is  $O(m^2 k^2 \times \binom{k+m}{m})$ .

## 5. Numerical Results

The procedures described in Sections 3 and 4 were implemented on a computer with Xeon 3.5 GHz processor, 64 GB RAM, and g++ compiler in an Ubuntu Operating System. They were tested on all instances proposed in [8], available in [www.matcuer.unam.mx/~davidr/cpp.html](http://www.matcuer.unam.mx/~davidr/cpp.html). Also, to find the global optimum of some instances we used the exact algorithm  $\mathcal{E}$  described in [5].

As for the SALP parameters we selected  $T_o = z(S_o)$  and  $T_f = 1$ , because from our computational experiments they yielded, respectively, an average of accepted solutions close to 99%, and an average of accepted solutions of around 2.5%. Also, we experimentally found that in terms of solution quality and computational effort a desirable combination of cooling factor and cycle length was, respectively,  $\varphi = 0.99$  and  $\lambda = t \times m$ . Each considered instance was solved 100 times.

Tables 1 and 2 show the achievements of our procedures for instances I001–I009 and I013–I016, compared with the best found in the literature. Global optima are marked with an asterisk (\*). The Appendix provides a detailed description of our solutions for I013–I016. In Table 1, column  $\beta$  indicates the number of times the best solution was obtained by SA/LP. In the case of instance I016, one hundred runs were made, the lower and the higher costs found being 264 145 and 278 778, respectively. Each run took  $7 \times 10^5$  seconds on average.

To test several heuristics, a set of 15 instances, see [5, 8, 11], all with  $t = 4$ , has been used in combination with grid costs:  $4C_2, 2C_2, C_2, 0.5C_2, 0.25C_2$  ( $C_2 = 18\ 676$ ), yielding a total of 75 cases. Table 3 displays our results on this set, comparing them with the best found so far, namely, those obtained with GRASP [11] (excepting instance 50-4 with grid cost  $C_2$ ). The first portion of the instance name corresponds to the number  $m$  of covers. For each of the 75 cases the best results are shown in bold face. An asterisk (\*) marks global optima (when able to compute them). The number of times SA/LP obtains the best solution is 17 for instance (30-1,4 $C_2$ ), nine for instance (30-4,4 $C_2$ ), five for instance (30-3,4 $C_2$ ), two for instances (30-2,4 $C_2$ ), (30-5,4 $C_2$ ), (40-1,4 $C_2$ ), (40-2,4 $C_2$ ), (40-3,4 $C_2$ ), and (30-3,2 $C_2$ ), and once for the remaining 66 instances. On average, linear programming took 97.65% of the total time of each run.

We observe the dominance of SA/LP for instances 30- $x$  when the grid cost is in the range  $[0.5C_2, 4C_2]$ , for instances

TABLE 1: Best known results for instances  $I001$ – $I009$  and  $I013$ – $I015$ .

Instance	$m$	$t$	Best solution found	Obtained by	$\beta$	worst solution
$I001$	3	4	136 472*	[5–8, 11] & SA/LP & $\mathcal{H}/TS$	100	136 472*
$I002$	4	4	247 916*	[1, 5–8, 11] & SA/LP & $\mathcal{H}/TS$	100	247 916*
$I003$	5	4	1 851 948*	[5, 8, 11] & SA/LP	86	1 856 155
$I004$	8	4	264 348*	[1, 5, 6, 8, 11] & SA/LP & $\mathcal{H}/TS$	100	264 348*
$I005$	12	4	269 584*	[5, 8, 11] & SA/LP & $\mathcal{H}/TS$	33	286 384
$I006$	15	4	515 256*	[5, 8, 11] & SA/LP & $\mathcal{H}/TS$	4	530 180
$I007$	9	8	5 283.53*	[5] & SA/LP & $\mathcal{H}/TS$	73	5 465.95
$I008$	17	8	11 455.64 (11 400.27*)	SA/LP	8	11 839.52
$I009$	18	7	11 052.09 (11 047.96*)	SA/LP	1	11 601.24
$I013$	30	4	1 745 800	SA/LP	1	1 801 828
$I014$	40	4	2 557 828	SA/LP	1	2 615 480
$I015$	50	4	6 507 340	SA/LP	1	6 599 908

TABLE 2: Comparison of our results with the previous best for instance  $I016$ .

Instance	$m$	$t$	cost	GRASP [11]		SA/LP	
				wastage	cost	wastage	cost
$I016$	100	25	264 697	1.939%	264 145	1.705%	

40- $x$  when this cost is in the range  $[C_2, 4C_2]$  and for instances 50- $x$  when the cost is  $4C_2$ . Also, SA/LP seems the best choice when the grid cost is relatively high or the number of covers is low. In total, SA/LP finds the best result in 45 of the 75 instances (including ties).

As of heuristic  $\mathcal{H}/TS$ , it finds the best result for instance 30-4 when the cost is  $0.25C_2$  or  $0.5C_2$ , for instances 40-2, 40-3, and 40-4 when the cost is  $0.5C_2$ , for instance 40-4 when the cost is  $0.25C_2$ , and for instances 50- $x$  when the cost is in the range  $[0.25C_2, 0.5C_2]$ , with one case excepted. Thus,  $\mathcal{H}/TS$  seems the best when the number of covers is high and the grid cost is small. All in all,  $\mathcal{H}/TS$  finds the best result in 24 of the 75 instances (ties included).

For its part, GRASP dominates in instances 30- $x$  with cost  $0.25C_2$ , in instances 40- $x$  with cost in the range  $[0.25C_2, 0.5C_2]$ , and in instances 50- $x$  with several combinations of instance-cost; it obtains the best result four times. GRASP seems to be the best option when the grid cost is small and the number of covers is below 40. Altogether, GRASP finds the best results in 20 of the 75 instances (ties included).

In regard to computational effort it is difficult to assess our proposal, and not only because GRASP was run on a computer different from ours. In the GRASP implementation, the number of grids is fixed, so the reported average execution times correspond to a given number of grids. Conversely, in SA/LP and  $\mathcal{H}/TS$  the number of grids is a variable to be determined. In our experiments  $\mathcal{H}/TS$  was run only once for each instance, because its behavior is nearly deterministic, the sole random component being the amount of time each element stays in the tabu list.

Table 4 shows the computer time needed by our two heuristics for instances 30- $x$ , 40- $x$ , and 50- $x$ . The figures for SA/LP correspond to average times (in seconds) of one run. Clearly, none of the procedures has linear complexity; this largely explains the observed increase in time as  $m$  grows.

We also tested SA/LP and  $\mathcal{H}/TS$  on 60 instances  $E001, \dots, E060$  found in [www.matcuer.unam.mx/~davidr/cpp.html](http://www.matcuer.unam.mx/~davidr/cpp.html), whose optimum is known. Table 5 shows the obtained results, where it can be seen that SA/LP gets indeed the optimum for the first 30 instances, and it outperforms  $\mathcal{H}/TS$  for instances  $E031, \dots, E050$ . However,  $\mathcal{H}/TS$  obtains the best solutions for  $E051, \dots, E060$ .

It is worth mentioning that a similar behavior can be observed for instances in Table 3, where  $\mathcal{H}/TS$  tends to dominate as  $m$  grows and  $C_2$  decreases.

## 6. Conclusions

In this paper we have presented two competitive procedures for the cover printing problem, SA/LP and  $\mathcal{H}/TS$ , comparing their results with the best found for a selected set of 88 instances from the specialized literature. Our computer experiments show that our heuristics, taken together, although not dominating in all of these instances, allowed improving the known best solution of 68% of them, their success depending more on the characteristics of the problem treated.

A total of 60 better results were obtained in the chosen set: five in the group of 13 instances of type  $Ixxx$  and 55 in the large group of 75 instances.

It is worth mentioning that one of the advantages of our heuristics is that none of them requires presetting the number of grids.

For future work it is desirable to intensify the study of new, original approaches to this difficult combinatorial optimization problem, so as to better understand its intricacies and provide improved solutions.

## Appendix

Tables 6, 7, 8, 9, and 10 contain a detailed description of our solution to instances  $I013, I014, I015$ , and  $I016$ , as well as the

TABLE 3: Results for instances 30-x, 40-x, and 50-x with five distinct grid costs. All results obtained with GRASP come from [11].

Instance	Method	$4C_2$	$2C_2$	$C_2$	$0.5C_2$	$0.25C_2$
30-1	GRASP	<b>2 252 096*</b>	1 946 280	1 774 024	1 674 526	<b>1 623 167</b>
	$\mathcal{H}/\text{TS}$	<b>2 252 096*</b>	1 942 248	1 774 938	1 678 558	1 625 376
	SA/LP	<b>2 252 096*</b>	<b>1 933 512*</b>	<b>1 765 288</b>	<b>1 673 252</b>	1 625 820
30-2	GRASP	2 139 200	1 832 040	1 657 096	1 563 716	<b>1 516 991</b>
	$\mathcal{H}/\text{TS}$	2 140 544	1 832 040	1 659 784	1 566 404	1 519 714
	SA/LP	<b>2 137 856*</b>	<b>1 822 632*</b>	<b>1 654 548</b>	<b>1 563 044</b>	1 519 007
30-3	GRASP	2 396 576	2 083 368	1 903 048	1 802 878	<b>1 749 468</b>
	$\mathcal{H}/\text{TS}$	<b>2 392 544*</b>	2 093 728	1 908 424	1 806 840	<b>1 749 468</b>
	SA/LP	<b>2 392 544*</b>	<b>2 082 024</b>	<b>1 900 360</b>	<b>1 802 206</b>	1 750 175
30-4	GRASP	<b>2 930 816*</b>	2 618 000	2 427 740	2 324 280	2 266 873
	$\mathcal{H}/\text{TS}$	2 937 536	2 613 296	2 427 068	<b>2 323 272</b>	<b>2 266 201</b>
	SA/LP	<b>2 930 816*</b>	<b>2 604 560</b>	<b>2 425 864</b>	2 323 608	2 269 897
30-5	GRASP	<b>3 064 656</b>	2 728 488	2 537 733	2 429 112	<b>2 371 033</b>
	$\mathcal{H}/\text{TS}$	3 082 688	2 741 648	2 541 168	2 431 800	2 374 022
	SA/LP	<b>3 064 656</b>	<b>2 724 848</b>	<b>2 534 588</b>	<b>2 427 768</b>	2 373 014
40-1	GRASP	4 931 696	4 509 512	4 264 353	<b>4 133 621</b>	<b>4 062 366</b>
	$\mathcal{H}/\text{TS}$	4 931 696	4 520 824	4 273 304	4 141 088	4 066 384
	SA/LP	<b>4 912 880</b>	<b>4 490 304</b>	<b>4 262 552</b>	4 135 782	4 068 330
40-2	GRASP	4 226 096	3 800 552	3 557 764	<b>3 427 494</b>	<b>3 355 744</b>
	$\mathcal{H}/\text{TS}$	4 233 488	3 808 616	3 562 328	<b>3 427 494</b>	3 356 717
	SA/LP	<b>4 215 344</b>	<b>3 786 048</b>	<b>3 552 388</b>	3 428 838	3 365 418
40-3	GRASP	4 846 464	4 386 536	4 143 748	<b>4 005 414</b>	<b>3 935 379</b>
	$\mathcal{H}/\text{TS}$	4 836 944	4 389 224	4 145 484	<b>4 005 414</b>	3 937 024
	SA/LP	<b>4 822 832</b>	<b>4 378 752</b>	<b>4 141 592</b>	4 018 854	3 949 421
40-4	GRASP	6 053 376	5 605 152	5 359 928	<b>5 218 304</b>	5 139 533
	$\mathcal{H}/\text{TS}$	6 066 032	5 604 200	5 361 412	<b>5 218 304</b>	<b>5 138 826</b>
	SA/LP	<b>6 037 920</b>	<b>5 590 368</b>	<b>5 358 584</b>	<b>5 218 304</b>	5 149 960
40-5	GRASP	5 707 968	5 247 368	4 993 548	<b>4 848 032</b>	<b>4 769 898</b>
	$\mathcal{H}/\text{TS}$	5 717 936	5 262 152	5 000 268	4 848 634	4 770 234
	SA/LP	<b>5 684 336</b>	<b>5 239,976</b>	<b>4 991 000</b>	4 854 010	4 779 607
50-1	GRASP	6 850 592	6 301 568	<b>6 002 752</b>	5 849 914	5 764 388
	$\mathcal{H}/\text{TS}$	6 844 432	6 300 504	6 009 472	<b>5 849 102</b>	<b>5 760 391</b>
	SA/LP	<b>6 827 744</b>	<b>6 296 472</b>	6 004 628	5 860 526	5 778 360
50-2	GRASP	8 650 992	8 070 272	7 771 456	7 609 070	<b>7 518 308</b>
	$\mathcal{H}/\text{TS}$	8 649 648	<b>8 062 880</b>	<b>7 764 064</b>	<b>7 603 834</b>	7 520 324
	SA/LP	<b>8 619 296</b>	8 066 520	7 781 396	7 625 870	7 540 078
50-3	GRASP	9 599 184	9 038 904	8 720 040	8 547 854	8 455 006
	$\mathcal{H}/\text{TS}$	9 595 824	<b>9 030 840</b>	<b>8 716 008</b>	<b>8 545 096</b>	<b>8 450 122</b>
	SA/LP	<b>9 581 600</b>	9 035 264	8 732 136	8 567 874	8 471 325
50-4	GRASP	10 217 424	9 650 536	9 314 480	9 127 720	9 026 171
	$\mathcal{H}/\text{TS}$	10 235 008	<b>9 639 504</b>	<b>9 309 776</b>	<b>9 124 290</b>	<b>9 024 827</b>
	SA/LP	<b>10 199 280</b>	9 641 800	9 316 636	9 139 676	9 038 834
50-5	GRASP	10 628 688	<b>10 050 656</b>	<b>9 717 960</b>	9 536 296	9 440 795
	$\mathcal{H}/\text{TS}$	10 656 240	10 051 048	9 718 352	<b>9 531 592</b>	<b>9 437 505</b>
	SA/LP	<b>10 624 656</b>	10 051 048	9 730 905	9 554 300	9 458 869

TABLE 4: Average running time of our heuristics (in sec.). For SA/LP 100 runs were made.

Heuristic	Instance set		
	30-x	40-x	50-x
SA/LP (LP time)	45	155	436
SA/LP (Total time)	46	159	446
$\mathcal{H}/\text{TS}$	85	218	758

TABLE 5: Performance of SA/LP and  $\mathcal{H}/TS$ , for instances E0XX, with different  $m$  and  $t$ .

Instance	$m$	$t$	Optimum	SA/LP	$\mathcal{H}/TS$
E001	13	6	51 750	<b>51 750*</b>	<b>51 750*</b>
E002	13	6	51 606	<b>51 606*</b>	54 606
E003	13	6	56 686	<b>56 686*</b>	<b>56 686*</b>
E004	13	6	51 257	<b>51 257*</b>	53 384
E005	13	6	56 322	<b>56 322*</b>	<b>56 322*</b>
E006	13	6	51 803	<b>51 803*</b>	<b>51 803*</b>
E007	13	6	56 272	<b>56 272*</b>	<b>56 272*</b>
E008	13	6	59 104	<b>59 104*</b>	<b>59 104*</b>
E009	13	6	52 855	<b>52 855*</b>	<b>52 855*</b>
E010	13	6	53 778	<b>53 778*</b>	<b>53 778*</b>
E011	25	8	80 749	<b>80 749*</b>	80 889
E012	25	8	78 369	<b>78 369*</b>	<b>78 369*</b>
E013	25	8	83 369	<b>83 369*</b>	<b>83 369*</b>
E014	25	8	71 417	<b>71 417*</b>	75 756
E015	25	8	75 762	<b>75 762*</b>	78 580
E016	25	8	82 608	<b>82 608*</b>	88 970
E017	25	8	85 839	<b>85 839*</b>	<b>85 839*</b>
E018	25	8	74 488	<b>74 488*</b>	<b>74 488*</b>
E019	25	8	77 548	<b>77 548*</b>	82 128
E020	25	8	77 670	<b>77 670*</b>	77 848
E021	41	10	110 779	<b>110 779*</b>	111 319
E022	41	10	115 729	<b>115 729*</b>	116 516
E023	41	10	112 568	<b>112 568*</b>	116 691
E024	41	10	118 169	<b>118 169*</b>	118 927
E025	41	10	103 499	<b>103 499*</b>	103 777
E026	41	10	100 873	<b>100 873*</b>	101 669
E027	41	10	105 308	<b>105 308*</b>	110 826
E028	41	10	106 658	<b>106 658*</b>	107 383
E029	41	10	98 331	<b>98 331*</b>	98 475
E030	41	10	109 138	<b>109 138*</b>	109 148
E031	61	12	145 705	<b>146 549</b>	151 894
E032	61	12	149 415	<b>149 716</b>	158 751
E033	61	12	143 983	<b>144 296</b>	150 224
E034	61	12	143 305	<b>143 814</b>	144 108
E035	61	12	138 134	<b>138 521</b>	139 318
E036	61	12	151 318	<b>151 579</b>	153 755
E037	61	12	142 725	<b>142 776</b>	149 208
E038	61	12	138 358	<b>138 385</b>	138 689
E039	61	12	142 520	<b>142 520*</b>	143 646
E040	61	12	143 932	<b>144 473</b>	150 242
E041	85	14	184 266	<b>185 009</b>	188 739
E042	85	14	181 953	<b>182 731</b>	183 641
E043	85	14	180 216	<b>180 793</b>	184 897
E044	85	14	187 387	<b>188 392</b>	191 421
E045	85	14	162 654	<b>163 293</b>	164 242
E046	85	14	177 954	<b>178 031</b>	178 662
E047	85	14	183 011	184 006	<b>183 841</b>
E048	85	14	188 604	<b>189 230</b>	189 349
E049	85	14	190 843	<b>191 072</b>	191 507
E050	85	14	175 522	<b>176 270</b>	177 200
E051	113	16	230 934	240 057	<b>238 169</b>
E052	113	16	216 732	226 391	<b>223 205</b>
E053	113	16	239 502	254 266	<b>244 122</b>
E054	113	16	217 150	223 559	<b>218 052</b>
E055	113	16	220 353	229 621	<b>221 531</b>
E056	113	16	218 323	226 786	<b>224 245</b>
E057	113	16	209 618	219 285	<b>215 136</b>
E058	113	16	220 605	230 901	<b>223 054</b>
E059	113	16	232 452	242 181	<b>239 358</b>
E060	113	16	232 409	242 633	<b>239 624</b>

TABLE 6: Solution to instance I013 with 10 grids; its cost is 1 745 800.

Grid	Imprints	Cover (plates in the grid)			
1	22000	3(1)	6(1)	7(1)	8(1)
2	20000	9(1)	10(1)	11(1)	12(1)
3	17000	2(1)	4(1)	5(1)	13(1)
4	15000	1(2)	15(1)	16(1)	
5	13500	14(1)	17(1)	18(1)	19(1)
6	11000	2(1)	20(1)	21(1)	22(1)
7	9000	4(1)	5(1)	23(1)	24(1)
8	5000	3(1)	25(1)	26(1)	27(1)
9	2500	14(1)	25(1)	28(1)	29(1)
10	1000	6(1)	17(1)	26(1)	30(1)

TABLE 7: Solution to instance I014 with 13 grids; its cost is 2 557 828.

Grid	Imprints	Cover (plates in the grid)			
1	30750	2(1)	3(1)	4(1)	7(1)
2	27000	8(1)	9(1)	10(1)	11(1)
3	25000	1(2)	5(1)	12(1)	
4	16250	2(1)	6(2)	15(1)	
5	15000	3(1)	14(1)	16(1)	18(1)
6	13000	13(1)	19(1)	20(1)	21(1)
7	12000	5(1)	22(1)	23(1)	24(1)
8	10750	4(1)	25(1)	26(1)	27(1)
9	9100	13(1)	17(1)	28(1)	30(1)
10	5000	17(1)	29(1)	31(1)	32(1)
11	4300	14(1)	33(1)	34(1)	35(1)
12	3000	8(1)	29(1)	36(1)	37(1)
13	1100	38(2)	39(1)	40(1)	

TABLE 8: Solution to instance I015 with 19 grids; its cost is 6 507 340.

Grid	Imprints	Cover (plates in the grid)			
1	60650	1(1)	4(1)	5(1)	12(1)
2	61000	8(1)	9(1)	10(1)	11(1)
3	51150	2(1)	3(1)	16(1)	17(1)
4	39000	2(1)	19(1)	20(1)	21(1)
5	37000	7(1)	13(1)	22(1)	23(1)
6	34350	1(1)	3(1)	15(1)	24(1)
7	33500	6(1)	7(1)	18(1)	25(1)
8	30000	14(1)	26(1)	27(1)	29(1)
9	25500	14(1)	28(1)	30(1)	31(1)
10	21000	13(1)	32(1)	33(1)	34(1)
11	19400	4(1)	6(2)	35(1)	
12	11900	5(1)	15(1)	38(1)	39(1)
13	10000	18(1)	37(1)	40(1)	41(1)
14	8025	8(1)	36(2)	42(1)	
15	6000	9(1)	10(1)	15(1)	43(1)
16	4000	19(1)	37(1)	44(1)	45(1)
17	3000	5(1)	28(1)	46(1)	47(1)
18	1450	5(1)	32(1)	48(1)	49(1)
19	850	11(1)	30(1)	44(1)	50(1)

TABLE 9: Solution to instance I016 with 8 grids, yielding 1.705% of wastage.

Grid	Imprints	Cover (quantity of plates in the grid)											
1	42 163	4(1)	6(1)	15(2)	20(1)	21(1)	22(1)	28(1)	30(1)	37(1)	38(1)	52(1)	66(1)
		67(1)	71(1)	78(1)	80(2)	84(1)	87(2)	88(1)	90(1)	97(1)	99(1)		
2	39 501	1(1)	3(1)	11(1)	12(2)	24(1)	27(1)	39(1)	40(1)	47(2)	50(1)	53(2)	55(2)
		61(2)	63(1)	68(1)	69(1)	89(1)	95(1)	100(2)					
3	35 839	7(1)	13(1)	20(1)	22(1)	23(2)	24(1)	26(1)	29(1)	34(2)	46(1)	52(1)	54(1)
		60(1)	62(2)	66(1)	72(1)	78(1)	83(2)	92(1)	93(1)	98(1)			
4	33 395	5(1)	10(1)	14(1)	16(2)	28(1)	29(1)	35(1)	43(3)	46(1)	48(1)	49(1)	51(2)
		64(1)	70(1)	74(1)	82(1)	84(1)	88(1)	94(1)	96(1)	99(1)			
5	30 964	2(2)	8(1)	16(1)	17(1)	18(1)	21(1)	25(1)	41(2)	44(1)	50(1)	56(1)	57(1)
		58(1)	63(1)	64(1)	69(1)	74(1)	75(3)	89(1)	94(1)	98(1)			
6	23 823	7(2)	14(1)	21(1)	24(1)	31(1)	33(1)	35(1)	40(1)	45(1)	46(1)	49(1)	50(1)
		51(1)	57(2)	59(1)	72(1)	74(1)	85(1)	86(2)	91(1)	92(2)			
7	18 800	2(1)	4(2)	12(1)	26(1)	27(1)	30(2)	33(1)	36(2)	39(2)	59(1)	61(1)	70(1)
		71(1)	73(1)	76(1)	77(2)	79(1)	81(1)	83(1)	90(1)				
8	15 660	5(1)	8(1)	9(1)	11(1)	17(3)	19(1)	25(1)	28(1)	31(1)	32(1)	35(1)	39(1)
		42(1)	56(1)	59(1)	62(1)	65(1)	71(1)	82(1)	88(1)	92(1)	95(1)	97(1)	

TABLE 10: Worst solutions obtained by SA/LP for instances 30-x, 40-x, and 50-x, with five distinct grid costs.

Instance	$4C_t$	$2C_t$	$C_t$	$0.5C_t$	$0.25C_t$
30-1	2 329 488	2 003 792	1 830 332	1 709 596	1 646 275
30-2	2 168 768	1 873 984	1 706 964	1 600 732	1 541 680
30-3	2 455 824	2 142 896	1 960 700	1 848 434	1 772 845
30-4	3 007 648	2 698 752	2 483 768	2 361 632	2 299 766
30-5	3 172 400	2 797 704	2 582 020	2 466 622	2 390 685
40-1	5 099 808	4 635 008	4 372 060	4 201 498	4 102 513
40-2	4 338 544	3 890 712	3 631 936	3 484 383	3 390 575
40-3	4 955 552	4 498 200	4 228 924	4 072 874	3 974 604
40-4	6 197 296	5 718 272	5 438 524	5 282 004	5 197 387
40-5	5 851 216	5 388 712	5 071 418	4 905 764	4 811 954
50-1	7 039 088	6 435 128	6 129 060	5 928 524	5 842 748
50-2	8 899 072	8 232 112	7 907 284	7 706 909	7 584 024
50-3	9 830 688	9 222 864	8 854 132	8 636 600	8 507 212
50-4	10 416 560	9 798 544	9 422 532	9 208 378	9 071 832
50-5	10 891 216	10 255 840	9 846 816	9 621 153	9 493 418

worst solutions obtained in one hundred runs for instances 30-x, 40-x, and 50-x. All results were found with SA/LP.

### Data Availability

We are willing to give all the instances and our solutions to the persons who request them.

### Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

### References

- [1] A. Ekici, Ö. Ergun, P. Keskinocak, and M. G. Lagoudakis, "Optimal job splitting on a multi-slot machine with applications in the printing industry," *Naval Research Logistics*, vol. 57, no. 3, pp. 237–251, 2010.
- [2] Y.-C. Hsieh and P.-S. You, "An immune evolutionary approach for the label printing problem," *International Journal of Computational Intelligence Systems*, vol. 7, no. 3, pp. 515–523, 2014.
- [3] K. F. C. Yiu, K. L. Mak, and H. Y. K. Lau, "A heuristic for the label printing problem," *Computers & Operations Research*, vol. 34, no. 9, pp. 2576–2588, 2007.

- [4] S. R. Mohan, S. K. Neogy, A. Seth, N. K. Garg, and S. Mittal, "An optimization model to determine master designs and runs for advertisement printing," *Journal of Mathematical Modelling and Algorithms*, vol. 6, no. 2, pp. 259–271, 2007.
- [5] D. Romero and F. Alonso-Pecina, "Ad hoc heuristic for the cover printing problem," *Discrete Optimization*, vol. 9, no. 1, pp. 17–28, 2012.
- [6] S. Elaoud, J. Teghem, and B. Bouaziz, "Genetic algorithms to solve the cover printing problem," *Computers & Operations Research*, vol. 34, no. 11, pp. 3346–3361, 2007.
- [7] J. Teghem, M. Pirlot, and C. Antoniadis, "Embedding of linear programming in a simulated annealing algorithm for solving a mixed integer production planning problem," *Journal of Computational and Applied Mathematics*, vol. 64, no. 1-2, pp. 91–102, 1995.
- [8] D. Tuytens and A. Vandaele, "Using a greedy random adaptive search procedure to solve the cover printing problem," *Computers & Operations Research*, vol. 37, no. 4, pp. 640–648, 2010.
- [9] C. V. Calderón, *Aplicación de métodos de optimización en la solución de un problema de la industria editorial (Graduate Thesis)*, University of Puerto Rico, Mayagüez, Porto Rico, Brazil, 2010.
- [10] L. E. Carrera, *Algoritmo para encontrar el óptimo a una simplificación de un problema combinatorio presente en la industria editorial (Graduate Thesis)*, Mathematics Department, Cinvestav, Mexico City, Mexico, 2006.
- [11] D. Tuytens and A. Vandaele, "Towards an efficient resolution of printing problems," *Discrete Optimization*, vol. 14, pp. 126–146, 2014.
- [12] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "Hybrid metaheuristics in combinatorial optimization: a survey," *Applied Soft Computing*, vol. 11, no. 6, pp. 4135–4151, 2011.
- [13] A. M. Gomes and J. F. Oliveira, "Solving irregular strip packing problems by hybridising simulated annealing and linear programming," *European Journal of Operational Research*, vol. 171, no. 3, pp. 811–829, 2006.
- [14] M. Caserta and S. Voss, "Metaheuristics: hybridizing metaheuristics and mathematical programming," in *Metaheuristics: Intelligent Problem Solving*, V. Maniezzo, T. Stützle, and S. Voss, Eds., pp. 1–38, Springer, Berlin, Germany, 2009.
- [15] Z. N. Popović, V. D. Kerleta, and D. S. Popović, "Hybrid simulated annealing and mixed integer linear programming algorithm for optimal planning of radial distribution networks with distributed generation," *Electric Power Systems Research*, vol. 108, pp. 211–222, 2014.
- [16] M. López-Ibáñez, F. Mascia, M.-E. Marmion, and T. Stützle, "A template for designing single-solution hybrid metaheuristics," in *Proceedings of the GECCO Comp '14*, pp. 1423–1426, New York, NY, USA, July 2014.
- [17] M. Mestria, "New hybrid heuristic algorithm for the clustered traveling salesman problem," *Computers & Industrial Engineering*, vol. 116, pp. 1–12, 2018.
- [18] E. Aarts, J. Kort, and W. Michiels, "Simulated annealing," in *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, E. K. Burke and G. Kendall, Eds., Springer Science+Business Media, New York, NY, USA, 2014.
- [19] R. J. Vanderbei, *Linear Programming. Foundations and Extensions*, vol. 196 of *International Series in Operations Research & Management Science*, Springer, Boston, MA, USA, 2014.
- [20] H. Sanvicente-Sánchez and J. Frausto-Solís, "A method to establish the cooling scheme in simulated annealing like algorithms," in *Proceedings of the International Conference on Computational Science and Its Applications*, pp. 755–763, Springer, Berlin, Heidelberg, 2004.
- [21] E. Taillard, "Tabu search," in *Metaheuristics*, P. Siarry, Ed., Springer, 2016.
- [22] F. Glover, "Tabu search—part I," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989.

## Research Article

# An Improved Lagrangian Relaxation Algorithm for the Robust Generation Self-Scheduling Problem

Ping Che <sup>1</sup>, Zhenhao Tang <sup>2</sup>, Hua Gong <sup>3</sup>, and Xiaoli Zhao<sup>3</sup>

<sup>1</sup>The Department of Mathematics, Northeastern University, Shenyang 110819, China

<sup>2</sup>School of Automation Engineering, Northeast Electric Power University, Jilin 132012, China

<sup>3</sup>School of Science, Shenyang Ligong University, Shenyang 110159, China

Correspondence should be addressed to Ping Che; [cocoche ping@163.com](mailto:cocoche ping@163.com)

Received 9 March 2018; Accepted 8 July 2018; Published 22 July 2018

Academic Editor: Guillermo Cabrera-Guerrero

Copyright © 2018 Ping Che et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The robust generation self-scheduling problem under electricity price uncertainty is usually solved by the commercial solver, which is limited in computation time and memory requirement. This paper proposes an improved Lagrangian relaxation algorithm for the robust generation self-scheduling problem where the quadratic fuel cost and the time-dependent exponential startup cost are considered. By using the optimal duality theory, the robust generation self-scheduling problem, which has a max-min structure, is reformulated as a minimization mixed integer nonlinear programming (MINLP) problem. Upon the reformulation, the Lagrangian relaxation algorithm is developed. To obtain a solvable relaxed problem, the variable splitting technique is introduced before the relaxation. The obtained relaxed problem is decomposed into a linear programming-type subproblem and multiple single-unit subproblems. Each single-unit subproblem is solved optimally by a two-stage backward dynamic programming procedure. The special cases of the problem are discussed and a two-stage algorithm is proposed. The proposed algorithms are tested on test cases of different sizes and the numerical results show that the algorithms can find near-optimal solutions in a reasonable time.

## 1. Introduction

In the competitive electricity market, the generation self-scheduling problem plays a key role in the planning and operation of electric power systems. An effective generation self-scheduling can help the power generation company to decide the bidding strategies and make generation plans for the subsequent time horizon. The generation self-scheduling problem is to determine the operation of generation units, including the on/off statuses of generation units and the output level of committed generation units, according to electricity prices over the scheduling horizon. The objective is to maximize the total generation profit. Because the formulation of the generation self-scheduling problem includes quadratic fuel cost function, exponential startup cost function, and integer variables indicating the on/off statuses of generation units, the problem is a nonconvex MINLP problem. As there could be a large number of generation units and a long scheduling horizon, the generation self-scheduling problem can

become large-scale and computationally challenging. This motivates us to study the solution method of the generation self-scheduling problem.

The decision-making of the generation self-scheduling is affected by electricity prices, which are volatile and full of uncertainty in the practical spot market. The uncertainty may be caused by the volatility of fuel prices and the changes in government subsidies, industry regulations, and local weather conditions. Therefore, electricity prices cannot be forecasted accurately and need to be treated as uncertain parameters. Two approaches are usually applied in dealing with uncertain parameters: stochastic programming and robust optimization. The stochastic programming approach relies on the probability distribution of uncertain parameters and typically expresses the possible realizations of uncertain parameters by a finite number of scenarios according to the probability distribution. Examples of using the approach can be seen in [1–7]. The drawback of the approach is that it is

difficult to obtain the exact distribution of uncertain parameters in the practice. The robust optimization approach is distribution-free. The approach represents uncertain parameters with a deterministic uncertainty set that contains all possible realizations or an adequate realization range of uncertain parameters. The obtained decision is robust against the variation of uncertain parameters within the uncertainty set. Examples of using the approach can be seen in [8–13]. To make the decision more practical, in this paper, we use the robust optimization approach to model uncertain electricity prices in the generation self-scheduling problem.

The solution method of the generation self-scheduling problem has been widely investigated, with the majority concentrated on the deterministic generation self-scheduling problem and the stochastic generation self-scheduling problem. For the deterministic generation self-scheduling problem, the mixed integer linear programming approach was used in [14–16], a Lagrangian relaxation algorithm was proposed in [17], the Lagrangian relaxation method and the MIP method were compared in [18], a particle swarm optimization algorithm was proposed in [19], a genetic algorithm was proposed in [20], a new memetic algorithm was proposed in [21], an ant colony optimization algorithm was proposed in [22], an evolutionary algorithm was proposed in [23], and a survey of solution methods was presented in [24]. For the stochastic generation self-scheduling problem where uncertain electricity prices are modeled by the stochastic programming approach, the mixed integer linear programming approach was used in [1–4], the typical Lagrangian relaxation algorithm was applied in [5], and a sample average approximation algorithm was proposed in [6].

For the robust generation self-scheduling problem where uncertain electricity prices are modeled by the robust optimization approach, the solution algorithm is not widely investigated. A critical review of the robust generation self-scheduling was presented in [25]. To provide a better sketch for the research of the robust generation self-scheduling problem, a comparison of some relevant literature is presented in Table 1. From Table 1, we can observe that the robust generation self-scheduling problem was usually solved by commercial solvers. In order to make the problem solvable by the solvers, the model was simplified by piecewise linearly approximating the quadratic fuel cost function [8, 10, 11], reducing the time-dependent exponential startup cost function as a constant [8, 10], omitting the startup cost [9, 12], or omitting the unit commitment decision [12]. The solution approach has two disadvantages. First, the use of commercial solvers is limited in computation time and memory requirement [18]. When the robust generation self-scheduling problem is large-scale, it is not practical to solve the problem by commercial solvers. Second, the simplification of the model will reduce the effect of the solution. Therefore, an effective solution algorithm should be developed to solve the robust generation self-scheduling problem.

In this paper, we develop a Lagrangian relaxation algorithm for the robust generation self-scheduling problem. Compared with the existing robust generation self-scheduling literature, this paper provides the following contributions:

(1) We consider the quadratic fuel cost and the exponential startup cost in the robust generation self-scheduling problem and develop an effective Lagrangian relaxation algorithm to solve the problem. The algorithm decomposes the problem into a linear programming-type subproblem and multiple single-unit subproblems. Each single-unit subproblem is solved by a two-stage backward dynamic programming procedure. The feasible solution is constructed by a heuristic algorithm.

(2) Unlike the typical Lagrangian relaxation algorithm that may lead to an unbounded relaxed problem for the considered problem, we introduce the variable splitting technique to improve the algorithm.

(3) A numerical comparison between the proposed algorithm and the MILP solver is reported. Numerical results demonstrate the effectiveness of the proposed algorithm.

The rest of the paper is organized as follows. The mathematical formulation of the robust generation self-scheduling problem is presented in Section 2. The solution methodology is proposed in Section 3. Numerical experiments are carried out in Section 4. The research is concluded in Section 5.

## 2. Problem Description and Formulation

We consider the robust generation self-scheduling problem under electricity price uncertainty from the viewpoint of a price-taking power producer. The objective is to maximize the total generation profit under the worst-case scenario within the considered range of electricity prices. The fuel cost is a quadratic function of the power generation level and the startup cost is an exponential function of the down time of the generation unit. The schedule needs to satisfy generation unit operation constraints and system electricity demand constraints. We take an hour as the scheduling period in the generation self-scheduling problem under consideration.

The remainder of the section is organized as follows. The notations used in the problem formulation are presented in Section 2.1. The modeling of uncertain electricity prices is described in Section 2.2. An MINLP model for the robust generation self-scheduling problem is formulated in Section 2.3.

### 2.1. Notations

#### Parameters

$n$ : Number of generation units

$T$ : Scheduling horizon

$RD_i/RU_i$ : Ramp-down/ramp-up rate of generation unit  $i$

$SD_i/SU_i$ : Shutdown/startup ramp rate of generation unit  $i$

$P_i^L/P_i^U$ : Minimum/maximum power output of generation unit  $i$

$L_i/U_i$ : Minimum down-time/up-time of generation unit  $i$

$D_t$ : Electricity demand in hour  $t$

$FC_i(\cdot, \cdot)$ : Fuel cost function of generation unit  $i$

TABLE I: A comparison of some relevant literature.

No.	Price Robust	Thermal Unit Commitment	Quadratic Fuel Cost	Exponential Startup Cost <sup>a</sup>	Solution	Reference
1	Y	Y	N	N	MILP Solver	[8]
2	Y	Y	Y	-	MIQP Solver	[9]
3	Y	Y	N	N	MILP Solver	[10]
4	Y	Y	N	N	MILP Solver	[11]
5	Y	N	Y	-	QP Solver	[12]
6	Y	Y	Y	Y	LR <sup>b</sup>	Proposed

<sup>a</sup> “-” indicates that the startup cost was not considered.

<sup>b</sup> Lagrangian relaxation algorithm.

$a_{0i}, a_{1i}, a_{2i}$ : Coefficients of the quadratic fuel cost function of generation unit  $i$

$SC_i(\cdot, \cdot)$ : Startup cost function of generation unit  $i$

$b_{1i}, b_{2i}, \tau_i$ : Coefficients of the startup cost function of generation unit  $i$

$\lambda_t^L/\lambda_t^U$ : Lower/upper bound of the electricity price in hour  $t$

$\lambda_t$ : Random parameter representing the electricity price in hour  $t$

$\Gamma$ : Budget lower bound of the sum of the electricity price  $\lambda_t$  within the scheduling horizon

$y_{i0}$ : Number of hours generation unit  $i$  being up or down at the end of hour 0

$u_{i0}$ : Binary parameter to indicate the initial on/off status of generation unit  $i$

$p_{i0}$ : Power generation level of generation unit  $i$  in hour 0

$t_i^d$ : Last hour of the time periods during which the on/off statuses of generation unit  $i$  must be the same to its initial on/off status

$t_i^c$ : Earliest hour that generation unit  $i$  can be committed in the scheduling horizon

$\gamma$ : Weight for the artificial variable introduced in the algorithm

#### Decision Variables

$u_{it}$ : Binary variable to indicate the on/off status of generation unit  $i$  in hour  $t$

$v_{it}$ : Binary variable to indicate if generation unit  $i$  is started up in hour  $t$

$y_{it}$ : State variable to indicate the number of hours generation unit  $i$  being up or down at the end of hour  $t$

$p_{it}$ : Power generation level of generation unit  $i$  in hour  $t$

$\omega_0, \omega_t$ : Dual variables

$z_t$ : Introduced variable in the solution method

$q_t$ : Artificial variable

$\xi_{1it}, \xi_{2it}, \mu_t, \eta_t$ : Lagrangian multipliers

2.2. *Modeling of Uncertain Electricity Prices.* According to the robust optimization approach, we model uncertain electricity prices with the following uncertainty set:

$$\Lambda := \{(\lambda_1, \dots, \lambda_T):$$

$$\lambda_t^L \leq \lambda_t \leq \lambda_t^U, \quad t = 1, \dots, T \quad (1a)$$

$$\sum_{t=1}^T \lambda_t \geq \Gamma\}. \quad (1b)$$

Constraints (1a) restrict the electricity price for each hour between a lower bound and an upper bound, which can be set to the 2.5% and 97.5% quantiles of the electricity price forecast, respectively. Constraint (1b) presents a budget lower bound  $\Gamma$  for the sum of the electricity price  $\lambda_t$  over the scheduling horizon. The budget value  $\Gamma$  is used to control the level of conservatism of the robust optimization approach. The smaller the value of  $\Gamma$  is, the more conservative the approach is. System operators can choose the budget value according to their requirements.

Within the above uncertainty set, there are infinite electricity price scenarios. For a fixed generation self-schedule, the best-case electricity price scenario is the scenario corresponding to the maximum electricity sales revenue, while the worst-case electricity price scenario is the scenario corresponding to the minimum electricity sales revenue. It can be observed that the best-case electricity price scenario is at  $\lambda_t = \lambda_t^U$  for all  $t$  for any fixed power generation level, while the worst-case electricity price scenario is related to the fixed power generation level and the budget value  $\Gamma$  [26].

2.3. *The Robust Generation Self-Scheduling Model.* The robust generation self-scheduling problem for a price-taking power producer, denoted by (RSS), is formulated as follows:

$$\begin{aligned} \max_{u,p} \min_{\lambda \in \Lambda} & \sum_{t=1}^T \lambda_t \sum_{i=1}^n p_{it} \\ & - \sum_{i=1}^n \sum_{t=t_i^c}^T [FC_i(p_{it}, u_{it}) + SC_i(y_{i,t-1}, v_{it})] \end{aligned} \quad (2)$$

$$\text{s.t. } u_{it} = u_{i0}, \quad i = 1, \dots, n, \quad t = 1, \dots, t_i^d \quad (3)$$

$$v_{it} = 0, \quad i = 1, \dots, n, \quad t = 1, \dots, t_i^d \quad (4)$$

$$u_{it} - u_{i,t-1} \leq u_{ik},$$

$$i = 1, \dots, n, k = t + 1, \dots, \min\{T, U_i + t - 1\}, \quad (5)$$

$$t = t_i^d + 1, \dots, T$$

$$u_{i,t-1} - u_{it} \leq 1 - u_{ik},$$

$$i = 1, \dots, n, k = t + 1, \dots, \min\{T, L_i + t - 1\}, \quad (6)$$

$$t = t_i^d + 1, \dots, T$$

$$u_{it} - u_{i,t-1} \leq v_{it},$$

$$i = 1, \dots, n, t = t_i^d + 1, \dots, T \quad (7)$$

$$y_{it}$$

$$= u_{it} \max\{1, y_{i,t-1} + 1\}$$

$$+ (1 - u_{it}) \min\{-1, y_{i,t-1} - 1\}, \quad (8)$$

$$i = 1, \dots, n, t = 1, \dots, T$$

$$u_{it} P_i^L \leq p_{it} \leq u_{it} P_i^U,$$

$$i = 1, \dots, n, t = 1, \dots, T \quad (9)$$

$$-RD_i u_{i,t+1} - SD_i (1 - u_{i,t+1}) \leq p_{i,t+1} - p_{it}$$

$$\leq RU_i u_{it} + SU_i (1 - u_{it}), \quad (10a)$$

$$i = 1, \dots, n, t = t_i^c - 1$$

$$-RD_i u_{i,t+1} - SD_i (1 - u_{i,t+1}) \leq p_{i,t+1} - p_{it}$$

$$\leq RU_i u_{it} + SU_i (1 - u_{it}), \quad (10b)$$

$$i = 1, \dots, n, t = t_i^c, \dots, T - 1$$

$$\sum_{i=1}^n p_{it} \leq D_t, \quad t = 1, \dots, T \quad (11)$$

$$p_{it} \geq 0, \quad i = 1, \dots, n, t = 1, \dots, T \quad (12)$$

$$u_{it}, v_{it} \in \{0, 1\}, \quad i = 1, \dots, n, t = 1, \dots, T \quad (13)$$

$$y_{it} : \text{nonzero integer}, \quad (14)$$

$$i = 1, \dots, n, t = 1, \dots, T$$

In the above formulation, the objective function (2) is to maximize the generation profit under the worst-case scenario within the uncertainty set of electricity prices. Under the max-min decision rule, the obtained generation schedule

is robust against the variation of electricity prices within the uncertainty set. The generation profit is determined by the electricity sales revenue, the fuel cost, and the startup cost over the scheduling horizon where the fuel cost is  $FC_i(p_{it}, u_{it}) = a_{0i}u_{it} + a_{1i}p_{it} + a_{2i}p_{it}^2$  and the startup cost is  $SC_i(y_{i,t-1}, v_{it}) = [b_{1i}(1 - \exp(y_{i,t-1}/\tau_i)) + b_{2i}]v_{it}$  according to [27]. Equations (3) and (4) show the impact of the initial statuses of units on decision-making. Inequalities (5) and (6) represent the minimum up-time and down-time requirements, respectively. Inequalities (7) represent the startup status. Equations (8) represent the relationship between the state variables in adjacent hours. Inequalities (9) represent the power generation capacity of units. Inequalities (10a) and (10b) represent the ramping-up and ramping-down rate limits. Inequalities (11) represent the electricity demand constraints. Constraints (12)-(14) show the value field of the decision variables.

### 3. Solution Methodology

As the objective function (2) is max-min-type and includes nonlinear cost functions, (RSS) is a max-min MINLP problem. To solve the problem, we first reformulate (RSS) as a minimization model using the idea provided in [26] and then develop an improved Lagrangian relaxation algorithm upon the reformulation.

*3.1. Reformulation of (RSS).* According to constraints (1a), let

$$\lambda_t = \lambda_t^U - (\lambda_t^U - \lambda_t^L) z_t, \quad t = 1, \dots, T \quad (15)$$

where  $z_t, t = 1, \dots, T$ , are introduced variables satisfying

$$0 \leq z_t \leq 1, \quad t = 1, \dots, T. \quad (16)$$

Then we can reformulate  $\min_{\lambda \in \Lambda} \sum_{t=1}^T \lambda_t \sum_{i=1}^n p_{it}$  as

$$\min \sum_{t=1}^T \lambda_t^U \sum_{i=1}^n p_{it} - \sum_{t=1}^T (\lambda_t^U - \lambda_t^L) z_t \sum_{i=1}^n p_{it} \quad (17)$$

$$\text{s.t.} \quad \sum_{t=1}^T (\lambda_t^U - \lambda_t^L) z_t \leq \sum_{t=1}^T \lambda_t^U - \Gamma. \quad (18)$$

By dualizing constraints (16) and (18), we can transform (16)-(18) as follows:

$$\max \sum_{t=1}^T \lambda_t^U \sum_{i=1}^n p_{it} - \left( \sum_{t=1}^T \lambda_t^U - \Gamma \right) \omega_0 - \sum_{t=1}^T \omega_t \quad (19)$$

$$\text{s.t.} \quad (\lambda_t^U - \lambda_t^L) \omega_0 + \omega_t \geq (\lambda_t^U - \lambda_t^L) \sum_{i=1}^n p_{it}, \quad (20)$$

$$t = 1, \dots, T$$

$$\omega_t \geq 0, \quad t = 1, \dots, T \quad (21)$$

$$\omega_0 \geq 0 \quad (22)$$

where  $\omega_t$ ,  $t = 1, \dots, T$ , and  $\omega_0$  are the dual variables for constraints (16) and (18), respectively.

Based on the above transformation, (RSS) can be reformulated as

(RSS1)

$$\begin{aligned} \max \quad & \sum_{t=1}^T \lambda_t^U \sum_{i=1}^n p_{it} \\ & - \sum_{i=1}^n \sum_{t=t_i^c}^T [FC_i(p_{it}, u_{it}) + SC_i(y_{i,t-1}, v_{it})] \\ & - \left( \sum_{t=1}^T \lambda_t^U - \Gamma \right) \omega_0 - \sum_{t=1}^T \omega_t \end{aligned} \quad (23)$$

s.t. constraints (3)-(14), and (20)-(22)

or equivalently  
(RSS2)

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{t=t_i^c}^T [FC_i(p_{it}, u_{it}) + SC_i(y_{i,t-1}, v_{it})] \\ & - \sum_{t=1}^T \lambda_t^U \sum_{i=1}^n p_{it} + \left( \sum_{t=1}^T \lambda_t^U - \Gamma \right) \omega_0 + \sum_{t=1}^T \omega_t \end{aligned} \quad (24)$$

s.t. the same constraints in (RSS1).

### 3.2. The Lagrangian Relaxation Algorithm

(1) *Variable Splitting-Based Lagrangian Relaxation.* When the Lagrangian relaxation algorithm is used to solve a generation scheduling problem, constraints that couple different units are typically relaxed to make the relaxed problem separable in units. In this paper, constraints that couple different units include constraints (11) and (20). If constraints (20) are relaxed, the resulting subproblem that contains dual variables  $\omega_0$  and  $\omega_t$ ,  $t = 1, \dots, T$ , will be as follows:

$$\begin{aligned} \min \quad & \left( \sum_{t=1}^T \lambda_t^U - \Gamma - \sum_{t=1}^T \eta_t (\lambda_t^U - \lambda_t^L) \right) \omega_0 \\ & + \sum_{t=1}^T (1 - \eta_t) \omega_t \end{aligned} \quad (25)$$

s.t. constraints (21) and (22)

where  $\eta_t$ ,  $t = 1, \dots, T$ , are nonnegative Lagrangian multipliers. Note that the above subproblem is unbounded when either  $\sum_{t=1}^T \lambda_t^U - \Gamma - \sum_{t=1}^T \eta_t (\lambda_t^U - \lambda_t^L)$  or  $1 - \eta_t$  is negative. Consequently, using the typical Lagrangian relaxation algorithm may lead to an unbounded relaxed problem.

To obtain a bounded relaxed problem, we reserve constraints (20) and introduce the variable splitting technique into the algorithm. The variable splitting technique is to duplicate certain decision variables by adding some artificial variables and variable copy constraints that link the decision

variables and the added artificial variables into the problem [28]. This technique is usually used to obtain a stronger lower bound for a minimization problem [29–33], but we use it to construct a solvable relaxed problem in this paper. Based on the technique, we add artificial variables  $q_t$ ,  $t = 1, \dots, T$ , and the following variable copy constraints:

$$\sum_{i=1}^n p_{it} = q_t, \quad t = 1, \dots, T \quad (26)$$

into (RSS2). We also introduce a parameter  $\gamma \in [0, 1]$  to represent the weight of the added artificial variables in the objective function. The resulting problem, denoted by (RSS3), is equivalent to (RSS2) and described as follows:

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{t=t_i^c}^T [FC_i(p_{it}, u_{it}) + SC_i(y_{i,t-1}, v_{it})] \\ & - \sum_{t=1}^T \lambda_t^U \left[ (1 - \gamma) \sum_{i=1}^n p_{it} + \gamma q_t \right] \\ & + \left( \sum_{t=1}^T \lambda_t^U - \Gamma \right) \omega_0 + \sum_{t=1}^T \omega_t \end{aligned} \quad (27)$$

s.t. constraints (3)-(10a), (10b), (12)-(14), (21), (22), (26)

$$(\lambda_t^U - \lambda_t^L) \omega_0 + \omega_t \geq (\lambda_t^U - \lambda_t^L) q_t, \quad t = 1, \dots, T \quad (28)$$

$$q_t \leq D_t, \quad t = 1, \dots, T \quad (29)$$

$$q_t \geq 0, \quad t = 1, \dots, T \quad (30)$$

where the value of  $\gamma$  will be discussed in Section 4.

We relax constraints (10b) and (26) and incorporate them into the objective function (27) by introducing Lagrangian multipliers  $\{\xi_{1it} \geq 0, \xi_{2it} \geq 0\}$ ,  $i = 1, \dots, n$ ,  $t = t_i^c, \dots, T - 1$ ,  $\pi_t \geq 0$ ,  $t = 1, \dots, T$ , and  $\mu_t \in \mathbb{R}$ ,  $t = 1, \dots, T$ . The obtained relaxed problem, denoted by (RP), is as follows:

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{t=t_i^c}^T [FC_i(p_{it}, u_{it}) + SC_i(y_{i,t-1}, v_{it})] \\ & - \sum_{t=1}^T \lambda_t^U \left[ (1 - \gamma) \sum_{i=1}^n p_{it} + \gamma q_t \right] + \left( \sum_{t=1}^T \lambda_t^U - \Gamma \right) \omega_0 \\ & + \sum_{t=1}^T \omega_t \\ & + \sum_{i=1}^n \sum_{t=t_i^c}^{T-1} \{ \xi_{1it} [p_{i,t+1} - p_{it} - RU_i u_{it} - SU_i (1 - u_{it})] \} \end{aligned}$$

$$\begin{aligned}
& +\xi_{2it} [p_{it} - p_{i,t+1} - RD_i u_{i,t+1} - SD_i (1 - u_{i,t+1})] \\
& + \sum_{t=1}^T \mu_t \left( \sum_{i=1}^n p_{it} - q_t \right)
\end{aligned} \tag{31}$$

s.t. constraints (3)-(10a), (12)-(14), (21), (22), and (28)-(30).

(2) *Solution of the Relaxed Problem.* Given the Lagrangian multipliers, (RP) can be decomposed into  $n + 1$  independent

$$\begin{aligned}
\min \quad & \sum_{t=t_i^c}^T \{FC_i(p_{it}, u_{it}) + [\mu_t - (1 - \gamma)\lambda_t^U] p_{it} + SC_i(y_{i,t-1}, v_{it})\} \\
& + (\xi_{2it_i^c} - \xi_{1it_i^c}) p_{it_i^c} + \xi_{1it_i^c} (SU_i - RU_i) u_{it_i^c} + \sum_{t=t_i^c+1}^{T-1} \{(\xi_{1i,t-1} - \xi_{1it} + \xi_{2it} - \xi_{2i,t-1}) p_{it} \\
& + [\xi_{1it} (SU_i - RU_i) + \xi_{2i,t-1} (SD_i - RD_i)] u_{it}\} \\
& + (\xi_{1i,T-1} - \xi_{2i,T-1}) p_{iT} + \xi_{2i,T-1} (SD_i - RD_i) u_{iT}
\end{aligned} \tag{33}$$

s.t. constraints (3)-(10a) and (12)-(14).

(RP1) is a linear programming model and can be solved optimally by using a commercial solver. Specially, if  $\Gamma = \sum_{t=1}^T \lambda_t^U$ , we can achieve the following optimal solution directly without calling the commercial solver:

$$q_t = \begin{cases} D_t, & \text{if } \gamma\lambda_t^U + \mu_t \geq 0, \\ 0, & \text{otherwise,} \end{cases} \quad t = 1, \dots, T, \tag{34}$$

$$\hat{p}_{it} = \begin{cases} \arg \min \{C_{it}(p_{it}) : p_{i,t-1} - RD_i \leq p_{it} \leq p_{i,t-1} + RU_i u_{i,t-1} + SU_i (1 - u_{i,t-1}), P_i^L \leq p_{it} \leq P_i^U\}, & \text{if } t = t_i^c, \\ \arg \min \{C_{it}(p_{it}) : P_i^L \leq p_{it} \leq P_i^U\}, & \text{if } t > t_i^c. \end{cases} \tag{37}$$

In the second stage, determine the on/off statuses of the unit using the backward dynamic programming. The state transition equations are

$$\begin{aligned}
f_{i,T+1}(y_{iT}) &= 0, \\
f_{it}(y_{i,t-1}) &= \min_{(u_{it}, v_{it}) \in \Omega(y_{i,t-1})} \{SC_i(y_{i,t-1}, v_{it}) \\
& + C_{it}(\hat{p}_{it}) u_{it} + f_{i,t+1}(y_{it})\}, \quad t = T, T-1, \dots, t_i^c,
\end{aligned} \tag{38}$$

where  $y_{it}$  is subject to constraint (8) and  $f_{it}(y_{i,t-1})$  is the optimal value function.

The obtained relaxation solution can provide a lower bound for the optimal objective function value of (RSS3).

(3) *Construction of the Feasible Solution.* Because constraints (10b) and (26) are relaxed, the optimal solution to the relaxed

subproblems which are denoted by (RP1) and (RP2<sub>*i*</sub>),  $i = 1, \dots, n$ , and expressed as follows:

$$\min \quad \left( \sum_{t=1}^T \lambda_t^U - \Gamma \right) \omega_0 + \sum_{t=1}^T \omega_t - \sum_{t=1}^T (\gamma\lambda_t^U + \mu_t) q_t \tag{32}$$

s.t. constraints (21), (22), and (28)-(30).  
(RP2<sub>*i*</sub>)

$$\omega_0 = \text{any value no less than } \max_{1 \leq t \leq T} \{q_t\}, \tag{35}$$

$$\omega_t = 0, \quad t = 1, \dots, T. \tag{36}$$

Each (RP2<sub>*i*</sub>) corresponds a single-unit subproblem. According to the feature of the subproblem, we solve it using a two-stage backward dynamic programming procedure [34, 35]. In the first stage, define  $C_{it}(p_{it})$  as the generalized generation cost of unit  $i$  in hour  $t$  if unit  $i$  is on in hour  $t$  in (RP2<sub>*i*</sub>) and determine the corresponding optimal generation level

problem is generally infeasible for (RSS3). To obtain a feasible solution, we propose the following heuristic method based on the current relaxation solution.

*Step 0.* Initialize  $t = 1$ .

*Step 1.* For each unit  $i$ , adjust  $p_{it}$  according to  $p_{i,t-1}$  to meet constraints (10b) corresponding to hour  $t$  without violating constraints (9).

*Step 2.* Decrease  $p_{it}$  for some units or shut off some units to meet constraint (11) corresponding to hour  $t$  without violating constraints (5), (6), (9), and (10b).

*Step 3.* If  $t \geq T$ , go to Step 4. Otherwise, set  $t = t + 1$  and go to Step 1.

*Step 4.* Check if constraints (20) are satisfied. If constraints (20) are satisfied, stop. Otherwise, solve the following linear programming problem to determine  $\omega_0$  and  $\omega_t, t = 1, \dots, T$ .

$$\min \left( \sum_{t=1}^T \lambda_t^U - \Gamma \right) \omega_0 + \sum_{t=1}^T \omega_t \quad (39)$$

s.t. constraints (20)-(22).

The obtained feasible solution can provide an upper bound for the optimal objective function value of (RSS3).

(4) *Updating of the Lagrangian Multipliers.* We initialize the Lagrangian multipliers by zero and update them according to the subgradient algorithm [36]. The iteration is stopped when the maximum number of iterations is reached or the relative duality gap  $(Z^U - Z^L)/|Z^L| \times 100\%$  is smaller than a certain threshold, where  $Z^L$  and  $Z^U$  are the best lower bound and upper bound obtained so far for (RSS3), respectively.

3.3. *Discussions on Special Cases.* (RSS) can be simplified in the following two special cases:  $\Gamma = \sum_{t=1}^T \lambda_t^L$  and  $\Gamma = \sum_{t=1}^T \lambda_t^U$ . In the case of  $\Gamma = \sum_{t=1}^T \lambda_t^L$ , we have  $\lambda_t = \lambda_t^L$  for all  $t$  according to constraints (1a) and (1b) and (RSS) can be simplified as a deterministic multi-unit generation self-scheduling problem. Similar result can be obtained in the case of  $\Gamma = \sum_{t=1}^T \lambda_t^U$ . Therefore, we propose the following two-stage algorithm for the two special cases:

*Stage 1.* Simplify (RSS) by determining  $\lambda_t, t = 1, \dots, T$ , according to the budget value  $\Gamma$ . If  $\Gamma = \sum_{t=1}^T \lambda_t^L$ , then  $\lambda_t = \lambda_t^L, t = 1, \dots, T$ . If  $\Gamma = \sum_{t=1}^T \lambda_t^U$ , then  $\lambda_t = \lambda_t^U, t = 1, \dots, T$ .

*Stage 2.* Solve the simplified problem by using the typical Lagrangian relaxation algorithm where constraints (10b) and (11) are relaxed.

Since both the variable splitting-based Lagrangian relaxation algorithm and the two-stage algorithm proposed above can solve (RSS) in the two special cases, we will choose the more effective one for each special case based on the numerical test in Section 4.

## 4. Numerical Results

In this section, we consider test cases of different sizes to implement the numerical experiments. For convenience, the variable splitting-based Lagrangian relaxation algorithm is denoted by A1 and the two-stage algorithm for the two special cases is denoted by A2. The organization of the section is as follows. First, the generation of the test cases is described in detail. Second, we discuss the impact of the weight  $\gamma$  on the performance of algorithm A1 and determine the value of  $\gamma$  in the experiments. Third, we test the performance of algorithms A1 and A2, respectively. Finally, we discuss the effect of the budget value  $\Gamma$  on the self-scheduling.

We use Visual C++ to implement the proposed algorithms on a PC with 2.83 GHz and 3.25-GB memory. The linear programming involved is solved by calling CPLEX 12.5.

TABLE 2: Range of values for parameters associated with units and demands.

Parameter	Range of values
$P_i^L$ (MW)	[40, 400]
$P_i^U (\leq 1200)$ (MW)	$[3P_i^L, 4P_i^L]$
$RD_i/RU_i$ (MW)	$[0.3P_i^U, 0.6P_i^U]$
$U_i/L_i$ (h)	$P_i^U < 600$ [1, 3]
	$P_i^U \geq 600$ [2, 5]
$y_{i0}$ (h)	On (0.4 chance) [1, 10]
	Off (0.6 chance) [-10, -1]
$x_{i0}$ if $u_{i0} = 1$ (MW)	$[P_i^L, P_i^U]$
$D_t$ (MW)	$[0.7 \sum_{i=1}^n P_i^U, \sum_{i=1}^n P_i^U]$
$a_{oi}$ (\$)	$[0.8P_i^U, 1.2P_i^U]$
$a_{1i}$ (\$/MW)	$P_i^U < 600$ [16.0, 17.5]
	$P_i^U \geq 600$ [18.0, 19.5]
$a_{2i}$ (\$/(MW) <sup>2</sup> )	$P_i^U < 400$ [0.03, 0.06]
	$P_i^U \geq 400$ [0.01, 0.03]
$b_{1i}$ (\$)	$P_i^U < 600$ $[4P_i^U, 6P_i^U]$
	$P_i^U \geq 600$ $[2P_i^U, 4P_i^U]$
$b_{2i}$ (\$)	$[0.4b_{1i}, 0.6b_{1i}]$
$\tau_i$ (h)	$P_i^U < 600$ [1, 4]
	$P_i^U \geq 600$ [3, 6]

4.1. *Test Cases.* Parameters for the test cases are presented as follows. The number of units is set to 10, 50, and 100, respectively. The scheduling horizon is set to 24, 96, and 168 hours, respectively. The combination of the two parameters forms nine problem sizes, in which  $10 \times 24, 10 \times 96,$  and  $50 \times 24$  are corresponding to small-sized problems,  $10 \times 168, 50 \times 96,$  and  $100 \times 24$  are corresponding to medium-sized problems, and  $50 \times 168, 100 \times 96,$  and  $100 \times 168$  are corresponding to large-sized problems according to the number of decision variables and the number of constraints included in the problem. For each problem size, ten test cases are generated randomly and tested. Therefore, a total of ninety cases are tested in the experiments.

For each test case, we let  $SD_i = RD_i + P_i^L, SU_i = RU_i + P_i^L,$  and  $RD_i = RU_i$  for convenience. Value ranges for parameters associated with units are partially based on those in [27] and shown in Table 2. We use Pennsylvania–New Jersey–Maryland (PJM) Interconnection Real Time data from 2005 to 2006 to forecast price ranges and set  $\lambda_t^L$  and  $\lambda_t^U$  to the endpoints of the following confidence interval at the 95% confidence level, respectively:

$$\left( \bar{\lambda}_t - \frac{t_{0.975}(m-1)}{\sqrt{m}} s_t, \bar{\lambda}_t + \frac{t_{0.975}(m-1)}{\sqrt{m}} s_t \right) \quad (40)$$

In (40),  $\bar{\lambda}_t$  is the sample mean,  $s_t$  is the sample standard deviation,  $m$  is the sample size, and  $t_{0.975}(m-1)$  is the 97.5% quantile of Student's  $t$ -distribution with  $m-1$  degrees of freedom. The forecasted price data for 168 hours are provided in Figure 1. The budget lower bound  $\Gamma$  is set to  $\sum_{t=1}^T \lambda_t^L + budget$   $\sum_{t=1}^T (\lambda_t^U - \lambda_t^L)$  where *budget* is allowed to vary within the set  $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$ . Note that *budget* = 0 and *budget* = 1 are corresponding to the two special cases

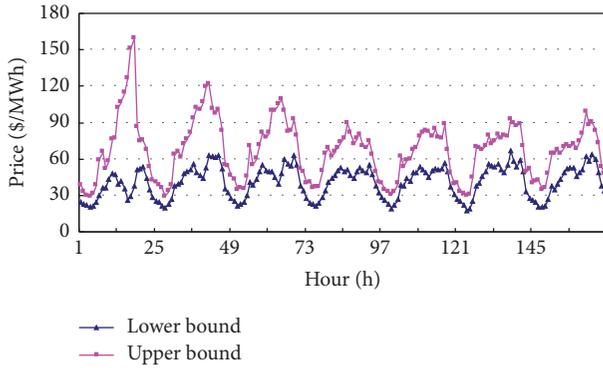


FIGURE 1: Price data for 168 hours (\$/MWh).

discussed in Section 3.3. The convergence threshold of the relative duality gap is set to 1%.

#### 4.2. Computational Results

(1) *Discussion on Weight  $\gamma$* . We allow  $\gamma$  to vary within the set  $\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$  and discuss the impact of the weight  $\gamma$  on the performance of algorithm A1 as follows. First, we carry out the algorithm for each test case under various *budget* and  $\gamma$  settings. Then, corresponding to each  $\gamma$  setting, we calculate the average relative duality gap and computation time over all test cases and report the results in Table 3. From Table 3, we can make the following observations:

- (1) The average relative duality gap varies between 0.90% and 0.99%. The results show that algorithm A1 can meet the convergence requirement under all  $\gamma$  settings.
- (2) The average computation time varies between 0.61 s and 0.99 s and reaches the minimum when  $\gamma$  is set to 0.4. The results indicate that algorithm A1 shows the fastest convergence behavior with  $\gamma$  setting to 0.4.

Based on the above discussion, we set  $\gamma$  to 0.4 for algorithm A1 in the following experiments.

(2) *Performance of the Algorithms*. We use the relative duality gap, the computation time, and the number of cases, denoted by  $N$ , in which the algorithm stops before reaching the maximum number of iterations to measure the performance of the proposed algorithms. The numerical results of algorithm A1 for all problem sizes and *budget* settings are reported in Tables 4 and 5. The numerical results of algorithm A2 for the two special cases are reported in Table 6. The reported relative duality gap and computation time are both the average of ten test cases in the same problem size.

From Tables 4 and 5, we have the following observations:

- (1) The average relative duality gap is 0.92% and the maximum relative duality gap is 0.96%. The computation time increases linearly with the increase of the problem size. The average computation time is 0.61 s and the maximum computation time is 3.74 s.

The results demonstrate that algorithm A1 can find a solution very close to the optimal one in a reasonable time for the cases in all sizes.

- (2) The average number of cases in which algorithm A1 stops before reaching the iteration limit is 10. The result shows that algorithm A1 can meet the convergence requirement within the iteration limit for almost all cases and indicates good convergence behavior of the algorithm.
- (3) The computation time shows a decreasing trend when *budget* increases from 0 to 1. This is because the conservatism of the solution decreases as the budget lower bound increases.

From Table 6, we can observe that algorithm A2 can meet the convergence requirement before reaching the iteration limit for all test cases. For *budget* = 0, the average relative duality gap is 0.89% and the average computation time is 0.81 s. For *budget* = 1, the average relative duality gap is 0.95% and the average computation time is 0.22 s. A comparison between the results of algorithms A1 and A2 indicates that algorithm A2 converges faster than algorithm A1 for the special cases. This is because (RSS) is simplified as a deterministic generation self-scheduling problem in algorithm A2.

Based on the numerical results in Tables 4–6, we suggest using algorithm A2 to solve (RSS) in the special cases and algorithm A1 to solve the problem in other cases.

As a comparison, we also solve (RSS2) by using the MILP approach where the quadratic fuel function is approximated by a ten-piece piecewise linear function, the exponential startup cost function is linearized, and the resulting MILP model is solved by calling CPLEX MILP solver. For each test case, the time limit is set to 1800 s. The numerical results are reported in Table 7. In Table 7, columns 2-7 report the numbers of solvable cases in which the MILP approach can reach the optimality within the time limit, denoted by  $NS$ , under various problem sizes and *budget* settings and columns 9-14 report the average computation times over the solvable cases. For the unsolvable cases, the MILP approach either cannot be optimally solved within the time limit or runs out of memory. If all cases cannot be optimally solved, the average computation time is not given.

From Table 7, we can have the following observations:

- (1) For the problem in small size, all cases can be optimally solved. For the problem in medium size, only partial cases can be optimally solved. For the problem in large size, none of the cases can be optimally solved. The largest size of the cases that the MILP approach can solve within the acceptable time is  $100 \times 24$ .
- (2) For the solvable cases, the computation time of the MILP approach grows exponentially as the problem size increases.

The observations show the computation limit of the MILP approach and imply the importance of proposing an effective solution algorithm. Because the proposed algorithm can find the near-optimal solutions within the acceptable time for the

TABLE 3: Average relative duality gap and computation time under various  $\gamma$  settings.

$\gamma$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Relative duality gap (%)	0.93	0.93	0.94	0.93	0.92	0.91	0.90	0.94	0.96	0.99	0.98
Computation time (s)	0.99	0.96	0.87	0.71	0.61	0.63	0.75	0.86	0.90	0.96	0.98

TABLE 4: Relative duality gap of algorithm A1.

Size	Relative duality gap (%)						Avg.
	<i>budget</i>						
	0	0.2	0.4	0.6	0.8	1	
10 × 24	0.92	0.94	0.92	0.96	0.92	0.88	0.92
10 × 96	0.92	0.95	0.94	0.92	0.93	0.93	0.93
10 × 168	0.95	0.96	0.96	0.91	0.92	0.93	0.94
50 × 24	0.92	0.86	0.94	0.91	0.88	0.90	0.90
50 × 96	0.92	0.92	0.94	0.95	0.92	0.90	0.92
50 × 168	0.93	0.90	0.95	0.96	0.93	0.90	0.93
100 × 24	0.92	0.86	0.92	0.92	0.89	0.86	0.90
100 × 96	0.93	0.86	0.91	0.92	0.84	0.91	0.90
100 × 168	0.95	0.87	0.91	0.94	0.89	0.91	0.91
Avg.	0.93	0.90	0.93	0.93	0.90	0.90	0.92

TABLE 5: Numerical results of algorithm A1.

Size	<i>N</i>							Computation time (s)						
	<i>budget</i>							<i>budget</i>						
	0	0.2	0.4	0.6	0.8	1	Avg.	0	0.2	0.4	0.6	0.8	1	Avg.
10 × 24	10	9	10	10	10	10	10	0.30	0.26	0.27	0.24	0.17	0.01	0.21
10 × 96	10	10	10	10	10	10	10	0.37	0.26	0.25	0.22	0.17	0.04	0.22
10 × 168	10	10	10	10	10	10	10	0.52	0.35	0.31	0.30	0.25	0.11	0.31
50 × 24	10	10	10	10	10	10	10	0.52	0.22	0.28	0.27	0.20	0.03	0.25
50 × 96	10	10	10	10	10	10	10	1.26	0.51	0.45	0.39	0.34	0.20	0.53
50 × 168	10	10	10	10	10	10	10	2.68	1.07	0.83	0.72	0.69	0.52	1.09
100 × 24	10	10	10	10	10	10	10	0.55	0.27	0.24	0.27	0.23	0.06	0.27
100 × 96	10	10	10	10	10	10	10	1.51	0.84	0.69	0.66	0.60	0.39	0.78
100 × 168	10	10	10	10	10	10	10	3.74	2.55	1.39	1.24	1.17	1.05	1.86
Avg.	10	10	10	10	10	10	10	1.27	0.70	0.52	0.48	0.43	0.27	0.61

TABLE 6: Numerical results of algorithm A2 for the two special cases.

Size	Relative duality gap (%)		<i>N</i>		Computation time (s)	
	<i>budget</i> = 0	<i>budget</i> = 1	<i>budget</i> = 0	<i>budget</i> = 1	<i>budget</i> = 0	<i>budget</i> = 1
10 × 24	0.65	0.96	10	10	0.02	0.01
10 × 96	0.81	0.98	10	10	0.10	0.08
10 × 168	0.96	0.98	10	10	0.29	0.18
50 × 24	0.76	0.96	10	10	0.04	0.03
50 × 96	0.99	0.94	10	10	0.69	0.16
50 × 168	0.99	0.95	10	10	1.58	0.43
100 × 24	0.86	0.96	10	10	0.14	0.06
100 × 96	0.99	0.93	10	10	1.37	0.29
100 × 168	0.98	0.93	10	10	3.07	0.75
Avg.	0.89	0.95	10	10	0.81	0.22

TABLE 7: Number of solvable cases<sup>a</sup> and average computation time of the MILP approach.

Size	NS							Computation time (s)						
	<i>budget</i>							<i>budget</i>						
	0	0.2	0.4	0.6	0.8	1	Avg.	0	0.2	0.4	0.6	0.8	1	Avg.
10 × 24	10	10	10	10	10	10	10	1.65	0.39	0.37	0.37	0.37	0.36	0.59
10 × 96	10	10	10	10	10	10	10	70.21	8.79	8.52	8.45	8.52	8.42	18.82
10 × 168	0	10	10	10	10	10	8	-	42.04	41.64	41.46	41.36	44.69	42.24
50 × 24	10	10	10	10	10	10	10	551.79	4.52	4.36	4.14	3.83	4.01	95.44
50 × 96	0	0	0	0	0	0	0	-	-	-	-	-	-	-
50 × 168	0	0	0	0	0	0	0	-	-	-	-	-	-	-
100 × 24	0	10	10	10	10	10	8	-	19.23	18.63	18.84	18.49	17.23	-
100 × 96	0	0	0	0	0	0	0	-	-	-	-	-	-	-
100 × 168	0	0	0	0	0	0	0	-	-	-	-	-	-	-
Avg.	3	6	6	6	6	6	5	-	-	-	-	-	-	-

<sup>a</sup> Corresponding to cases that can be optimally solved by using the MILP approach within the time limit.

TABLE 8: Average generation level under various *budget* settings.

Size	Generation level (MW)						
	<i>budget</i>						
	0	0.2	0.4	0.6	0.8	1	Avg.
10 × 24	81858	115014	121810	124072	126819	127648	116203
10 × 96	398015	511554	536327	548467	559971	565509	519974
10 × 168	703614	885812	912438	934035	951012	958648	890927
50 × 24	422552	593972	616302	635507	644835	652059	594205
50 × 96	2046938	2592646	2659243	2713494	2763548	2787158	2593838
50 × 168	3570388	4489274	4600006	4681392	4759769	4800147	4483496
100 × 24	858882	1209880	1248902	1283975	1310584	1324268	1206082
100 × 96	3945357	4989315	5127241	5223253	5321735	5341850	4991459
100 × 168	7137883	8959433	9198064	9372351	9531833	9606864	8967738
Avg.	2129498	2705211	2780037	2835172	2885567	2907128	2707102

TABLE 9: Average objective function value under various *budget* settings.

Size	Generation profit (\$)						
	<i>budget</i>						
	0	0.2	0.4	0.6	0.8	1	Avg.
10 × 24	891716	1520921	2481063	3529387	4638719	5854005	3152635
10 × 96	5746889	7767879	11021736	14558938	18378060	22535395	13334816
10 × 168	10738997	13711115	18570527	23872184	29601911	35815388	22051687
50 × 24	4569506	7826961	12692856	17991655	23687784	29867163	16105988
50 × 96	29829539	40525389	56480048	73811914	92586379	112876879	67685024
50 × 168	54385184	69585348	93813824	120348245	149010436	180186711	111221625
100 × 24	9213887	15795275	25797137	36504533	47984444	60668555	32660638
100 × 96	57959284	78591223	109414984	142950970	179071718	218221036	131034869
100 × 168	108862841	139454061	188030093	241240640	298751036	360937693	222879394
Avg.	31355316	41642019	57589141	74978718	93745610	114106980	68902964

cases in all sizes, it is more effective than the MILP approach, especially for the problem in medium or large size.

(3) *Effect of the Budget Value on the Self-Scheduling.* To show the effect of the budget value on the self-scheduling, we compare the generation levels and the objective function values

under various *budget* settings, respectively. The numerical results are reported in Tables 8 and 9. From Tables 8 and 9, we can observe that given the problem size, both the generation level and the generation profit increase as *budget* increases. This is because the problem becomes less conservative with the increase of *budget*.

## 5. Conclusions

In this paper, we propose an improved Lagrangian relaxation algorithm for the robust generation self-scheduling problem under electricity price uncertainty in the deregulated electricity market. The problem includes quadratic fuel cost, exponential startup cost, unit operation constraints, and electric demand constraints. To avoid obtaining an unbounded relaxed problem, variable splitting is introduced into the algorithm. For the special cases of the budget value, we also propose a two-stage algorithm. Numerical results demonstrate the good performance of the proposed algorithms. Future research can be focused on generalizing the proposed algorithm to solve other robust generation self-scheduling problems.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research is partly supported by the National Natural Science Foundation of China (71402021, 61503072) and Science and Technology Project of Shenyang (17231131).

## References

- [1] M. A. Plazas, A. J. Conejo, and F. J. Prieto, "Multimarket optimal bidding for a power producer," *IEEE Transactions on Power Systems*, vol. 20, no. 4, pp. 2041–2050, 2005.
- [2] H. Haghghat, H. Seifi, and A. R. Kian, "On the self-scheduling of a power producer in uncertain trading environments," *Electric Power Systems Research*, vol. 78, no. 3, pp. 311–317, 2008.
- [3] T. Li, M. Shahidehpour, and Z. Li, "Risk-constrained bidding strategy with stochastic unit commitment," *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 449–458, 2007.
- [4] R. Laia, H. M. I. Pousinho, R. Melico, and V. M. F. Mendes, "Self-scheduling and bidding strategies of thermal units with stochastic emission constraints," *Energy Conversion and Management*, vol. 89, pp. 975–984, 2015.
- [5] G. P. Granelli, P. Marannino, M. Montagna, and F. Zanellini, "Monte Carlo based unit commitment procedures for the deregulated market environment," *International Journal of Electrical Power & Energy Systems*, vol. 28, no. 10, pp. 712–722, 2006.
- [6] Q. Wang, J. Wang, and Y. Guan, "Price-based unit commitment with wind power utilization constraints," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 2718–2726, 2013.
- [7] P. Che, L. Tang, and J. Wang, "Two-stage minimax stochastic unit commitment," *IET Generation, Transmission & Distribution*, vol. 12, no. 4, pp. 947–956, 2018.
- [8] L. Baringo and A. J. Conejo, "Offering strategy via robust optimization," *IEEE Transactions on Power Systems*, vol. 26, no. 3, pp. 1418–1425, 2011.
- [9] A. Soroudi, "Robust optimization based self scheduling of hydro-thermal Genco in smart grids," *Energy*, vol. 61, pp. 262–271, 2013.
- [10] H. M. I. Pousinho, J. Contreras, P. Pinson, and V. M. F. Mendes, "Robust optimisation for self-scheduling and bidding strategies of hybrid CSP-fossil power plants," *International Journal of Electrical Power & Energy Systems*, vol. 67, pp. 639–650, 2015.
- [11] S. Dehghan, N. Amjady, B. Vatani, and H. Zareipour, "A new hybrid stochastic-robust optimization approach for self-scheduling of generation companies," *International Transactions on Electrical Energy Systems*, vol. 26, no. 6, pp. 1244–1259, 2016.
- [12] X. Luo, C. Chung, H. Yang, and X. Tong, "Robust optimization-based generation self-scheduling under uncertain price," *Mathematical Problems in Engineering*, vol. 2011, Article ID 497014, 17 pages, 2011.
- [13] R. Jiang, J. Wang, and Y. Guan, "Robust unit commitment with wind power and pumped storage hydro," *IEEE Transactions on Power Systems*, vol. 27, no. 2, pp. 800–810, 2012.
- [14] J. M. Arroyo and A. J. Conejo, "Optimal response of a thermal unit to an electricity spot market," *IEEE Transactions on Power Systems*, vol. 15, no. 3, pp. 1098–1104, 2000.
- [15] S. Liu, J. Chevallier, and B. Zhu, "Self-scheduling of a power generating company: carbon tax considerations," *Computers & Operations Research*, vol. 66, pp. 384–392, 2016.
- [16] C. K. Simoglou, P. N. Biskas, and A. G. Bakirtzis, "Optimal self-scheduling of a thermal producer in short-term electricity markets by MILP," *IEEE Transactions on Power Systems*, vol. 25, no. 4, pp. 1965–1977, 2010.
- [17] L. Tang and P. Che, "Generation scheduling under a CO<sub>2</sub> emission reduction policy in the deregulated market," *IEEE Transactions on Engineering Management*, vol. 60, no. 2, pp. 386–397, 2013.
- [18] T. Li and M. Shahidehpour, "Price-based unit commitment: A case of Lagrangian relaxation versus mixed integer programming," *IEEE Transactions on Power Systems*, vol. 20, no. 4, pp. 2015–2025, 2005.
- [19] I. Jacob Raglend, C. Raghuvver, G. Rakesh Avinash, N. P. Padhy, and D. P. Kothari, "Solution to profit based unit commitment problem using particle swarm optimization," *Applied Soft Computing*, vol. 10, no. 4, pp. 1247–1256, 2010.
- [20] P. S. Georgilakis, "Genetic algorithm model for profit maximization of generating companies in deregulated electricity markets," *Applied Artificial Intelligence*, vol. 23, no. 6, pp. 538–552, 2009.
- [21] D. K. Dimitroulas and P. S. Georgilakis, "A new memetic algorithm approach for the price based unit commitment problem," *Applied Energy*, vol. 88, no. 12, pp. 4687–4699, 2011.
- [22] C. C. Columbus, K. Chandrasekaran, and S. P. Simon, "Nodal ant colony optimization for solving profit based unit commitment problem for GENCOs," *Applied Soft Computing*, vol. 12, no. 1, pp. 145–160, 2012.
- [23] M. J. Ghadi, A. I. Karin, and A. Baghrmian, "Optimal power scheduling of thermal units considering emission constraint for GENCOs' profit maximization," *International Journal of Electrical Power & Energy Systems*, vol. 82, pp. 124–135, 2016.
- [24] N. P. Padhy, "Unit commitment—a bibliographical survey," *IEEE Transactions on Power Systems*, vol. 19, no. 2, pp. 1196–1205, 2004.
- [25] B. Vatani, B. Chowdhury, S. Dehghan, and N. Amjady, "A critical review of robust self-scheduling for generation companies under electricity price uncertainty," *International Journal of Electrical Power & Energy Systems*, vol. 97, pp. 428–439, 2018.

- [26] D. Bertsimas and M. Sim, "Robust discrete optimization and network flows," *Mathematical Programming*, vol. 98, no. 1–3, pp. 49–71, 2003.
- [27] J. F. Bard, "Short-term scheduling of thermal-electric generators using Lagrangian relaxation," *Operations Research*, vol. 36, no. 5, pp. 756–766, 1988.
- [28] M. Guignard and S. Kim, "Lagrangean decomposition: a model yielding stronger Lagrangean bounds," *Mathematical Programming*, vol. 39, no. 2, pp. 215–228, 1987.
- [29] P. Barcia and K. Jörnsten, "Improved Lagrangean decomposition: an application to the generalized assignment problem," *European Journal of Operational Research*, vol. 46, no. 1, pp. 84–92, 1990.
- [30] J. Batut and A. Renaud, "Daily generation scheduling optimization with transmission constraints: a new class of algorithms," *IEEE Transactions on Power Systems*, vol. 7, no. 3, pp. 982–989, 1992.
- [31] O. Nilsson and D. Sjelvgren, "Variable splitting applied to modelling of start-up costs in short term hydro generation scheduling," *IEEE Transactions on Power Systems*, vol. 12, no. 2, pp. 770–775, 1997.
- [32] A. Belloni, A. L. Diniz Souto Lima, M. E. Maceira, and C. A. Sagastizabal, "Bundle relaxation and primal recovery in unit commitment problems. The Brazilian case," *Annals of Operations Research*, vol. 120, pp. 21–44, 2003.
- [33] L. Tang, P. Che, and J. Wang, "Corrective unit commitment to an unforeseen unit breakdown," *IEEE Transactions on Power Systems*, vol. 27, no. 4, pp. 1729–1740, 2012.
- [34] X. Guan, P. B. Luh, H. Yan, and J. A. Amalfi, "An optimization-based method for unit commitment," *International Journal of Electrical Power & Energy Systems*, vol. 14, pp. 9–17, 1992.
- [35] A. Frangioni and C. Gentile, "Solving nonlinear single-unit commitment problems with ramping constraints," *Operations Research*, vol. 54, no. 4, pp. 767–775, 2006.
- [36] M. L. Fisher, "The Lagrangian relaxation method for solving integer programming problems," *Management Science*, vol. 27, no. 1, pp. 1–18, 1981.

## Research Article

# Multiple-Try Simulated Annealing Algorithm for Global Optimization

Wei Shao <sup>1</sup> and Guangbao Guo <sup>2</sup>

<sup>1</sup>*School of Management, Qufu Normal University, Rizhao, Shandong 276826, China*

<sup>2</sup>*Department of Statistics, Shandong University of Technology, Zibo 255000, China*

Correspondence should be addressed to Wei Shao; [wshao1031@gmail.com](mailto:wshao1031@gmail.com)

Received 18 March 2018; Revised 23 May 2018; Accepted 29 May 2018; Published 17 July 2018

Academic Editor: Guillermo Cabrera-Guerrero

Copyright © 2018 Wei Shao and Guangbao Guo. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Simulated annealing is a widely used algorithm for the computation of global optimization problems in computational chemistry and industrial engineering. However, global optimum values cannot always be reached by simulated annealing without a logarithmic cooling schedule. In this study, we propose a new stochastic optimization algorithm, i.e., simulated annealing based on the multiple-try Metropolis method, which combines simulated annealing and the multiple-try Metropolis algorithm. The proposed algorithm functions with a rapidly decreasing schedule, while guaranteeing global optimum values. Simulated and real data experiments including a mixture normal model and nonlinear Bayesian model indicate that the proposed algorithm can significantly outperform other approximated algorithms, including simulated annealing and the quasi-Newton method.

## 1. Introduction

Since the 21<sup>st</sup> century, the modern computers have greatly expanded the scientific horizon by facilitating the studies on complicated systems, such as computer engineering, stochastic process, and modern bioinformatics. A large volume of high dimensional data can easily be obtained, but their efficient computation and analysis present a significant challenge.

With the development of modern computers, Markov chain Monte Carlo (MCMC) methods have enjoyed a enormous upsurge in interest over the last few years [1, 2]. During the past two decades, various advanced MCMC methods have been developed to successfully compute different types of problems (e.g., Bayesian analysis, high dimensional integral, and combinatorial optimization). As an extension of MCMC methods, the simulated annealing (SA) algorithm [1–3] has become increasingly popular since it was first introduced by Kirkpatrick et al. (1983). As Monte Carlo methods are not sensitive to the dimension of data sets, the SA algorithm plays an important role in molecular physics, computational chemistry, and computer science. It has also been successfully applied to many complex optimization problems.

Several improved optimization methods have been proposed recently [4, 5] and successfully applied to polynomial and vector optimization problems, min–max models, and so on [6–10]. Although Sun and Wang (2013) discussed the error bound for generalized linear complementarity problems, all of these improved methods were designed for special optimization problems and not for global optimization problems. The SA algorithm is a global optimization algorithm that can obtain global optimization results with slowly decreasing temperature schedule. However, Holley et al. (1989) pointed out that only with the use of a “logarithmic” cooling schedule could the SA algorithm converge to the global minimum with probability one [11, 12]. Liang et al. (2014) improved the SA algorithm by introducing the simulated stochastic approximation annealing (SAA) algorithm [13, 14]. Such algorithm can work with a square-root cooling schedule in which the temperature can decrease much faster than that in a “logarithmic” cooling schedule. Karagiannis et al. (2017) extended the SAA algorithm by using population Monte Carlo ideas and introduced the parallel and interacting stochastic approximation annealing (PISAA) algorithm [15].

In the present study, we propose a variation of the SA algorithm, i.e., the multiple-try Metropolis based simulated

annealing (MTMSA) algorithm, for global optimization. The MTMSA algorithm is a combination of the SA algorithm and multiple-try Metropolis (MTM) algorithm [16]. The MTM algorithm, which allows several proposals from different proposal distributions in the multiple-try step simultaneously, achieves a higher rate of convergence than the standard Metropolis algorithm (e.g., random walk Metropolis algorithm) [1, 2, 17]. Thus, the MTMSA algorithm can guarantee that global minima are reached with a rapidly decreasing cooling schedule. Comparing with PISAA, which should run on multicore computer to their advantage, the MTMSA often owns high convergent rate by use of the efficient vector operation with MATLAB or R on personal and super computer. Simulation and real data examples show that, under the framework of the multiple-try algorithm, the MTMSA algorithm can reach global minima under a rapidly decreasing cooling schedule relative to that of the SA algorithm.

The remainder of this paper is organized as follows. Section 2 describes the framework of the MTMSA algorithm. Section 3 illustrates the comparison of the MTMSA algorithm with other optimization methods in a mixture normal model. Section 4 presents the application of the MTMSA algorithm to Bayesian analysis and half-space depth computation through real data sets. Finally, Section 5 summarizes the conclusions derived from the study.

## 2. MTMSA Algorithm

**2.1. Overview of SA Algorithm.** The SA algorithm originates from the annealing process, which is a thermodynamic process used to attain a low energy state in condensed matter physics [1]. The process comprises two steps. The first state is the high temperature state, in which solids transform into liquid and particles move freely to ward ideal. Then the temperature drops to zero slowly and the movement of the particles become restricted such that the desired structure is achieved. Realizing that the Metropolis algorithm [18] can be used to simulate the movements of particles, Kirkpatrick et al. (1983) proposed a computer simulation based physical annealing process, i.e., the SA algorithm.

Suppose our goal is to find the minimum value of  $h(\mathbf{x})$ ,  $\mathbf{x} \in D$ . This goal is equivalent to the search for the maximum value of  $\exp\{-h(\mathbf{x})/T\}$ ,  $\mathbf{x} \in D$ , with any positive temperature  $T$ . Let  $T_1 > T_2 > \dots > T_k > \dots$  be a decreasing temperature sequence, with large  $T_1$  and  $\lim_{k \rightarrow +\infty} T_k = 0$ . In every temperature  $T_k$ , we use the Metropolis-Hastings (MH) algorithm (or Gibbs sampling algorithm [19]) to update the Markov chain  $N_k$  times, with  $\pi_k(\mathbf{x}) \propto \exp\{-h(\mathbf{x})/T_k\}$  as its stationary distribution. When  $k$  is increasing, an increasing number of samples concentrate in the maximum value nearby. The SA algorithm can be summarized as follows:

- (1) Initialize  $\mathbf{x}^{(0)}$  with starting temperature  $T_1$ .
- (2) At current temperature  $T_k$ , update the Markov chain  $N_k$  times, with  $\pi_k(\mathbf{x})$  as its stationary distribution, and transmit the last state  $\mathbf{x}$  to the next temperature.
- (3) Update  $k$  to  $k + 1$ .

The SA algorithm can reach the global optimum when the temperature sequence decreases slowly (e.g., the inverse logarithmic rate, i.e., the order of  $O(\log(L_k)^{-1})$ , where  $L_k = N_1 + \dots + N_k$  [1, 2, 11, 12]). However, no one can afford to use such a slow cooling schedule. Various improved SA algorithms have thus been designed and to overcome the excessively slow cooling schedule and to successfully resolve various optimization problems in industries and commerce [14, 20–23]. Realizing that the MTM algorithm can overcome the “local-trap” problem and enjoy a high convergence rate, we propose the MTMSA algorithm, which is a combination of the MTM algorithm and the SA algorithm.

**2.2. MTMSA Algorithm.** The Metropolis algorithm is the first iterative sampling algorithm of the MCMC algorithm. Hastings extended this algorithm by allowing the proposal distribution to be an asymmetric distribution, i.e., the MH algorithm [24].

The MH algorithm is an iterative MCMC sampling algorithm, whose iterative points  $\{x_0, x_1, \dots, x_n, \dots\}$  show a limiting distribution  $\pi(x)$ . The challenge of using the MH algorithm is that it tends to suffer from the “local-trap” problem when the target distribution function  $\pi(x)$  is a multimodal distribution [2, 19]. It eventually impedes the convergence of the SA algorithm. The MTM algorithm can overlap the “local-trap” problem. The MTM algorithm allows several proposal points simultaneously and selects the best one as the next sampling point while keeping the stationary distribution unchanged. Thus, combining the MTM algorithm and SA algorithm yields the MTMSA algorithm.

Suppose the global optimization problem is

$$\min_{\mathbf{x} \in D} h(\mathbf{x}). \quad (1)$$

The whole procedure of the MTMSA algorithm for problem (1) is summarized below.

- (1) Set the temperature parameters  $T_{max}, T_{min}, a \in (0, 1)$ , the length of the Markov chain  $N_k$ , and the number of multiple-try  $m$  in the MTM algorithm. Initialize the state of the Markov chain  $\mathbf{x}$ , and set  $k = 1$ .
- (2) At current temperature  $T_k = T_{max} \times a^k$ , let

$$\pi_k(\mathbf{x}) \propto \exp\left\{-\frac{h(\mathbf{x})}{T_k}\right\} \quad (2)$$

be the stationary distribution. For  $l = 1, 2, \dots, N_k$ , use the MTM algorithm to update the Markov chain  $N_k$  times.

- (2.1) Propose a “proposal set” of size  $m$  from  $T(\mathbf{x}, \cdot)$ , denoted as  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$ , where  $\mathbf{x}_i \in D$  and  $T(\mathbf{x}, \cdot)$  is any symmetric proposal transform distribution.
- (2.2) Randomly choose a proposal state  $\mathbf{x}^*$  from  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$  with probability  $\{\pi_k(\mathbf{x}_1), \pi_k(\mathbf{x}_2), \dots, \pi_k(\mathbf{x}_m)\}$ .
- (2.3) Propose a “reference set” of size  $m$ , denoted as  $\{\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \dots, \bar{\mathbf{x}}_m\}$ , where  $\{\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_{m-1}\}$  is proposed from  $T(\mathbf{x}^*, \cdot)$ , and set  $\bar{\mathbf{x}}_m = \mathbf{x}$ .

```

Input:  $T_{max}, T_{min}, a, N_k, m, h(\mathbf{x}), T(\mathbf{x}, \mathbf{y})$ 
Output:  $\mathbf{x}$ 
(1) Initialize:  $\mathbf{x} = \mathbf{1}, k = 1, T_1 = T_{max}$ 
(2) while  $T_k > T_{min}$  do
(3)   set  $\pi_k(\mathbf{x}) \propto \exp\{-h(\mathbf{x})/T_k\}$ ;
(4)   for  $l = 1$  to  $N_k$  do
(5)      $s_p = 0$ ;
(6)      $s_r = 0$ ;
(7)     for  $i = 1$  to  $m$  do
(8)       sample  $\mathbf{x}_i$  from  $T(\mathbf{x}, \cdot)$ ;
(9)        $p_i = \pi_k(\mathbf{x}_i)$ ;
(10)       $s_p = s_p + p_i$ ;
(11)     choose  $\mathbf{x}^*$  from  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$  with probability  $\{p_1, p_2, \dots, p_m\}$ ;
(12)     for  $j = 1$  to  $m - 1$  do
(13)       sample  $\tilde{\mathbf{x}}_j$  from  $T(\mathbf{x}^*, \cdot)$ ;
(14)        $s_r = s_r + \pi_k(\tilde{\mathbf{x}}_j)$ ;
(15)      $s_r = s_r + \pi_k(\mathbf{x})$ ;
(16)     set  $r = \min\{s_p/s_r, 1\}$ ;
(17)     sample  $u$  from uniform distribution in  $[0, 1]$ ;
(18)     if  $u < r$  then
(19)       set  $\mathbf{x} = \mathbf{x}^*$ ;
(20)        $k = k + 1$ ;
(21)        $T_k = T_{max} \times a^k$ ;
(22)   return  $\mathbf{x}$ ;

```

ALGORITHM 1: MTMSA algorithm used to detect the minimum of  $h(\mathbf{x}), \mathbf{x} \in D$ .

(2.4) Calculate the generalized Metropolis ratio.

$$r = \min \left\{ \frac{\pi_k(\mathbf{x}_1) + \pi_k(\mathbf{x}_2) + \dots + \pi_k(\mathbf{x}_m)}{\pi_k(\tilde{\mathbf{x}}_1) + \pi_k(\tilde{\mathbf{x}}_2) + \dots + \pi_k(\tilde{\mathbf{x}}_m)}, 1 \right\}. \quad (3)$$

Then update the current state of the Markov chain with probability  $r$ . Set  $\mathbf{x} = \mathbf{x}^*$ ; otherwise, reject it, and keep  $\mathbf{x}$  unchanged.

- (3) If  $T_k < T_{min}$ , output the last solution  $\mathbf{x}$  and the minimum value of (1) of the whole procedure; otherwise, update  $k$  to  $k + 1$ , and proceed to step (2).

Furthermore, **Algorithm 1** gives the pseudocode of MTMSA algorithm for the computation of problem (1).

The convergence of the MTMSA algorithm can be obtained from the stationary distribution of the MTM algorithm (i.e., the detailed balance condition of the MTM algorithm [1]). Theoretically, when  $T_k$  approaches zero and the step number of the MTM algorithm is sufficiently large, all samples drawn from  $\pi_k$  would be in the vicinity of the global minimum of  $h(\mathbf{x})$  in  $D$ .

The next proposition gives the computation complex of the MTMSA algorithm.

**Proposition 1.** *The computation complex of the MTMSA algorithm is*

$$O(Nnm), \quad (4)$$

where  $N$  is the length of decreasing cooling temperature,  $n$  is the frequency of the Markov chain update, and  $m$  is the number of multiple-try points.

*Proof.* The proof of the proposition directly follows the procedure of the MTMSA algorithm described above. The decreasing cooling temperature and the length of the Markov chain are the external loops of the MTMSA algorithm. By combining the external loops with the internal loop of the multiple-try model, we then complete the proof of the proposition.  $\square$

The proposition indicates that the computation complex of the MTMSA algorithm is a polynomial in  $N$  and  $m$ . Given the computation complex of a stationary distribution  $\pi_k(\mathbf{x})$ , the computation complex of the MTMSA algorithm is not greater than the polynomial in  $d$  (where  $d$  is the dimension of  $\mathbf{x}$ ).

The MTMSA algorithm has many advantages over other approximation algorithms. Compared with traditional optimization algorithms (such as the Nelder-Mead (NM) method [25] and the quasi-Newton (QN) method [26]), which are local optimization methods, the MTMSA algorithm gets more accurate results often, as shown in our simulated multimodal experiment. In practice, the speed of the MTMSA algorithm is generally high, particularly for an efficient vector operation (or parallel computing) with MATLAB or R in the evaluation of multiple-try points. The MTMSA algorithm clearly outperforms the SA algorithm in our experiment. Furthermore, by setting the number of multiple-try points  $m = 1$ , we can obtain a special case of the MTMSA algorithm, that is, the SA algorithm. Simulated and real data examples in subsequent sections show the advantage of the MTMSA over other approximated algorithms.

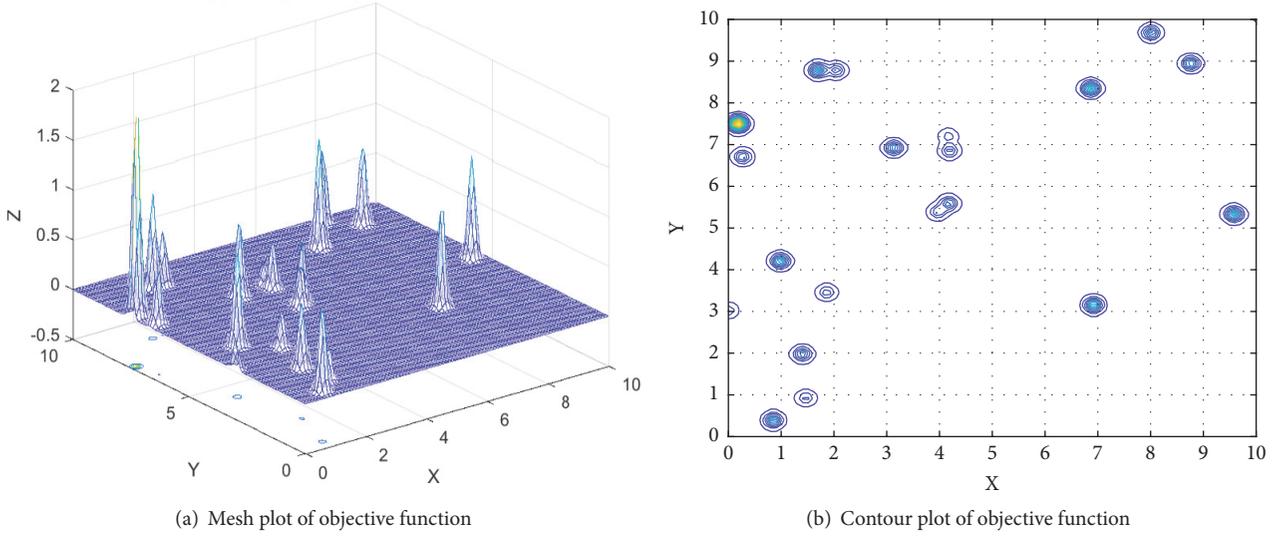


FIGURE 1: Mesh and contour plot of the objective function.

### 3. Simulated Experiment Results

This section presents a simulation example (i.e., mixture normal model). The main purpose of this example is to demonstrate that the MTMSA algorithm could compute the optimization problem in the case of multiple local maxima and outperform its counterparts in terms of accuracy and efficiency. All results are obtained using R language (version X64 2.3.4) and MATLAB (version R2017a) on a Dell OptiPlex7020MT desktop computer with Intel(R) Core(TM) i7-4790 CPU @ 3.6 GHz, RAM 8.00 GB, and Windows 7 Ultimate with Service Pack 1 (x64). The R and MATLAB codes in this work are available upon request to the corresponding author.

In this simulation example, we consider a two-dimensional multimodal mixture normal model modified from [27, 28]. In this model, the objective function is the probability density function, which is the combination of 20 normal models

$$f(\mathbf{x}) \propto \sum_{i=1}^{20} \frac{\omega_i}{2\pi\sigma_i^2} \exp^{-\frac{1}{2\sigma_i^2}(\mathbf{x}-\boldsymbol{\mu}_i)'(\mathbf{x}-\boldsymbol{\mu}_i)}, \quad (5)$$

where  $\sigma_1 = \sigma_2 = \dots = \sigma_{20} = 0.1$  and  $(\omega_1, \omega_2, \dots, \omega_{20})$ , which are the weights of the 20 normal models, are chosen to be the arithmetic progression from 1 to 5, except the last one  $\omega_{20} = 10$ . The 20 mean vectors are independently sampled from the uniform distribution from  $[0, 10]$ .

Figure 1 illustrates the mesh and contour plots of (5), which contains 20 modes in this objective function. This example poses a serious challenge for optimization because many classical optimization methods may converge on the local optimum in this multimodal example. Clearly, the global maximum point is the last mode  $(0.183, 7.501)$  with the maximum value of 2.449.

Four methods are used to find the global maximum point of this optimization problem: the NM method, modified QN method, SA algorithm, and MTMSA algorithm. The NM and QN methods are commonly applied numerical optimization

algorithms, and the QN method allows box constraints. The SA and its improved version, i.e., the MTMSA algorithm, are stochastic optimization algorithms. Apart from the minimum temperature  $T_{min}$ , another commonly used parameter that controls the degree of decreasing temperature is  $N_{temper}$ :

$$T_k = T_{max} \cdot \alpha^{k-1}, \quad k = 1, 2, \dots, N_{temper}. \quad (6)$$

For the SA algorithm, we set the degree of decreasing temperature  $N_{temper} = 75$ , the starting temperature  $T_{max} = 10$ , the decreasing parameter  $\alpha = 0.9$ , the length of the Markov chain  $N_{mc} = 100$ , and the proposal variance of the Metropolis algorithm  $v_{pro} = 2$ . For the MTMSA algorithm, we set the number of multiple-tries  $N_{mtm} = 100$ . The other parameters are  $N_{temper} = 25$ ,  $\alpha = 0.8$ ,  $T_{max} = 1$ ,  $N_{mc} = 100$ , and  $v_{pro} = 2$ . With different  $N_{temper}$  (75 and 25) and  $\alpha$  (0.9 and 0.8, respectively) values, the SA and MTMSA algorithms have similar  $T_{min}$ .

We tested these four algorithms (NM, QN, SA, and MTMSA algorithms) to compute the optimization problem and repeated this computation 50 times. The computation results are summarized in Figure 2 and Table 1.

The mean value, standard deviation (sd), mean square error (MSE), total CPU time (in seconds), and average CPU time for one accurate result (in seconds) of the results from different algorithms are summarized in Table 1, where  $mean = (1/R) \sum_{i=1}^R V_i$ ,  $sd = \sqrt{(1/(R-1)) \sum_{i=1}^R (V_i - mean)^2}$ ,  $MSE = (1/R) \sum_{i=1}^R (V_i - V_e)^2$ , and  $V_e = 2.449$  is the exact maximum value in the model (5). Figure 2 illustrates the boxplot of 50 computation results from these four algorithms.

Suffering from the ‘‘local-trap’’ problem, NM and QN algorithms cannot find the global maximum successfully in 50 computations (they often find other local modes in (5)). Compared with the MTMSA algorithm, the SA algorithm uses a slowly decreasing temperature schedule ( $N_{temper} = 75$ ) and consumes more CPU time. However, only 10 results of 50 repetitions from the SA algorithm converge to the

TABLE 1: Computation results (mean, sd, MSE, consumed total CPU time, and average CPU time (in seconds)) of the 50 computations from different algorithms.

	DM	QN	SA	MTMSA
mean	0.3447	0.0311	1.7523	2.4392
sd	0.4563	0.1355	0.6427	0.0086
MSE	4.6319	5.8635	0.8901	0.0001
total CPU time	4.1732	0.9121	1305.6	528.1
average CPU time	$+\infty$	$+\infty$	130.56	10.562

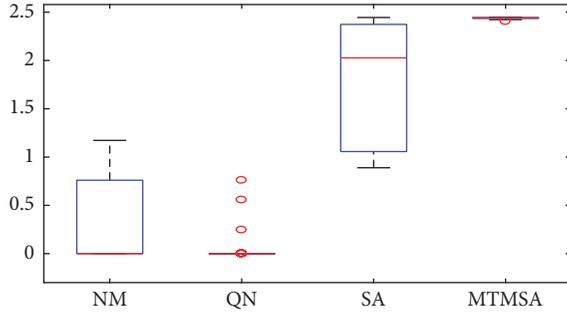


FIGURE 2: Boxplot of 50 results computed from the NM method, QN method, SA algorithm, and MTMSA algorithm.

global maximum point (0.183, 7.501), and the mean of the SA algorithm is 1.7523. By contrast, the MTMSA algorithm functions with a rapidly decreasing temperature schedule. The MTMSA algorithm consumes minimal CPU time (only about 8 min), but it yields highly accurate results (all 50 results converge to the global maximum). Furthermore, the MTMSA algorithm only needs approximately 10 seconds to compute one accurate result, whereas the SA algorithm requires about 130 seconds. All results from NM and QN algorithms suffer from the “local-trap” problem.

We compared the differences in the decreasing temperature schedules used in SA and MTMSA algorithms. “The slower the better” was found to be applicable to the decreasing temperature schedules. A rapidly decreasing temperature schedule may result in the “local-trap” problem. In the next simulation, we set the temperature schedules to decrease from 10 to  $5 \times 10^{-3}$ , and the length of decreasing temperature was set to 500, 75 and 25 for SA and MTMSA algorithms. Each computation was repeated 50 times.

The length of decreasing temperature of the SA algorithm is set to 75, 500, and denoted as the SA1 and SA2 algorithm, respectively. The SA2 algorithm shows the slowest decreasing schedule. It uses 500 steps to drop from the highest temperature to the lowest one. Thus almost all 50 results converge to the global maximum (about 94% percent of computation results escape from local optima and reaches the global maximum). The SA1 algorithm uses a rapidly decreasing schedule, and only about half of the 50 results converge to the global maximum (about 54% percent of computation results escape from local optima and reaches the global maximum). By contrast, the MTMSA algorithm only uses 25 steps in decreasing temperature, but all of the 50 results converge to the global maximum.

TABLE 2: Biochemical oxygen demand versus time.

Time (days)	BOD (mg/I)
1	8.3
2	10.3
3	19.0
4	16.0
5	15.6
7	19.8

Figure 3 shows the decreasing schedules and convergence paths of the three algorithms. We find that when the temperature decreases to about 0.02 (corresponding to the 50th, 400th, and 20th steps in SA1, SA2, and MTMSA), all sample paths from the three algorithms converge to their local and global optima. All the sample paths of MTMSA converge to the global optima, and lots of sample paths of SA1 and SA2 converge to the local optima because the average sample path of MTMSA in Figure 3 is the highest and at the level about 2.43. The MTMSA algorithm uses the rapidly decreasing schedule and achieves the fastest convergence rate. Therefore, the MTMSA algorithm is the most efficient and accurate in this simulation example.

## 4. Real Data Examples

4.1. *Bayesian Analysis Using MTMSA.* In this section, we illustrate the application of the MTMSA algorithm in Bayesian analysis with real data from [29]. In this example, we fit a nonlinear model derived from exponential decay

$$y_i = \theta_1 (1 - \exp\{-\theta_2 x_i\}) + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2), \quad (7)$$

with a fixed rate that is constant to a real data set [30] (Table 2).

The variables BOD (mg/I) and time (days) in Table 2 are the response and control variables in model (7) (denoted as the BOD problem) with a constant variance  $\sigma^2$  for independent normal errors. The likelihood for the BOD problem is

$$L(\theta_1, \theta_2, \sigma^2 | X, Y) \propto \exp \left\{ -6 \log \sigma - \frac{1}{2} \frac{\sum_{i=1}^6 (y_i - \theta_1 (1 - \exp\{-\theta_2 x_i\}))^2}{\sigma^2} \right\}, \quad (8)$$

where  $X = (x_1, \dots, x_6)$  and  $Y = (y_1, \dots, y_6)$ .

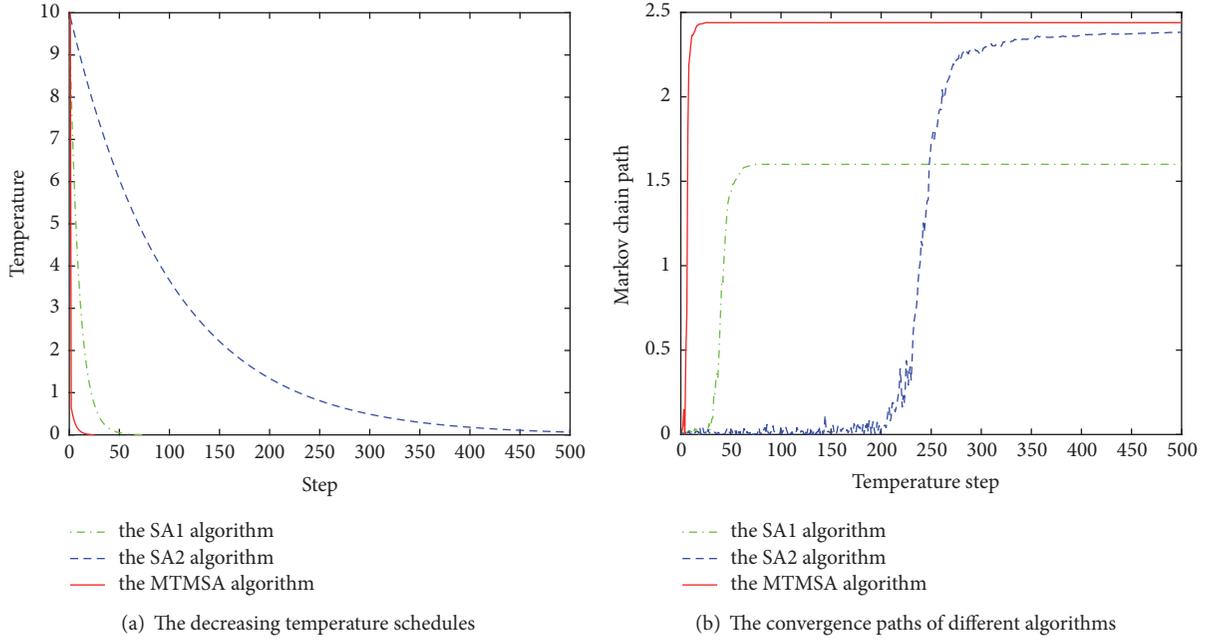


FIGURE 3: Decreasing temperature schedules (a) and convergence paths (b) of the SA1 algorithm, SA2 algorithm, and MTMSA algorithm. The convergence paths are the average of 50 paths.

TABLE 3: Computation results (mean ( $10^{-5}$ ), sd ( $10^{-5}$ )) in special temperature steps (1, 5, 10, 15, 20, 25) from 20 repetitions.

	step 1	step 5	step 10	step 15	step 20	step 25
mean	1.05	1.39	0.91	3.92	145	148
sd	3.57	5.16	2.20	12.2	3.04	0.24

While choosing the flat prior for the parameters  $\sigma^2$  and  $(\theta_1, \theta_2)$  (i.e., the uniform distribution in  $(0, +\infty)$  and  $[-20, 50] \times [-2, 6]$ , respectively) and integrating out  $\sigma^2$ , we obtain the following (improper) posterior distribution of  $(\theta_1, \theta_2)$ :

$$p(\theta_1, \theta_2 | X, Y) \propto \left[ \sum_{i=1}^6 (y_i - \theta_1 (1 - \exp\{-\theta_2 x_i\}))^2 \right]^{-2} \cdot I_{[-20, 50] \times [-2, 6]}(\theta_1, \theta_2), \quad (9)$$

where

$$I_{[-20, 50] \times [-2, 6]}(\theta_1, \theta_2) = \begin{cases} 1, & (\theta_1, \theta_2) \in [-20, 50] \times [-2, 6] \\ 0, & (\theta_1, \theta_2) \notin [-20, 50] \times [-2, 6]. \end{cases} \quad (10)$$

For a Bayesian analysis, one often treats the parameters  $(\theta_1, \theta_2)$  as random variables. In this work, we use the posterior distribution of  $(\theta_1, \theta_2)$  for their statistical inference and use the posterior mode of (9) as the estimation of  $(\theta_1, \theta_2)$ , which coincides with the maximum likelihood estimation. The Bayesian statistical inference of the parameters  $(\theta_1, \theta_2)$  is

translated to the global optimization problem in  $[-20, 50] \times [-2, 6]$ .

$$\sup_{(\theta_1, \theta_2) \in [-20, 50] \times [-2, 6]} p(\theta_1, \theta_2 | X, Y). \quad (11)$$

In addition, we use the MTMSA algorithm to compute the global optimization problem (11). The parameters of the MTMSA algorithm are set to be  $N_{mtm} = 20$ ,  $N_{temper} = 25$ ,  $\alpha = 0.6$ ,  $T_{max} = 1$ , and  $N_{mc} = 1000$ . The computation is then repeated 20 times. Figure 4 and Table 3 illustrate the decreasing temperature schedule and the convergence paths of 20 repetitions from the MTMSA algorithm. After 20 steps, all 20 computation paths become convergent to  $1.48 \times 10^{-3}$ , which has the largest mean and smallest sd.

Figure 5 shows the mesh (a) and contour (b) plots of the posterior distribution (9). Figure 6 and Table 4 show the locations of the scatters  $(\theta_1, \theta_2)$  from 20 repetitions at different temperature steps. With the temperature decreasing from  $0.6$  to  $2.8 \times 10^{-6}$ , all scatters converge to the optimization point  $(19.15, 0.53)$ .

**4.2. Half-Space Depth Computation Using MTMSA.** As a powerful tool for nonparametric multivariate analysis, half-space depth (HD also known as Tukey depth) has been eliciting increased interest since it was introduced by Tukey [31, 32]. HD, which extends univariate order-related statistics to multivariate settings, provides a center-outward ordering

TABLE 4: Location results of  $(\theta_1, \theta_2)$  at different temperature levels.

Level	0.6	$2.8 \times 10^{-4}$	$1.6 \times 10^{-4}$	$2.8 \times 10^{-6}$
mean	(20.22, 1.80)	(16.63, 1.06)	(18.25, 1.05)	(19.15, 0.53)
sd	(21.71, 2.39)	(17.64, 1.92)	(9.58, 1.56)	(0.11, 0.01)

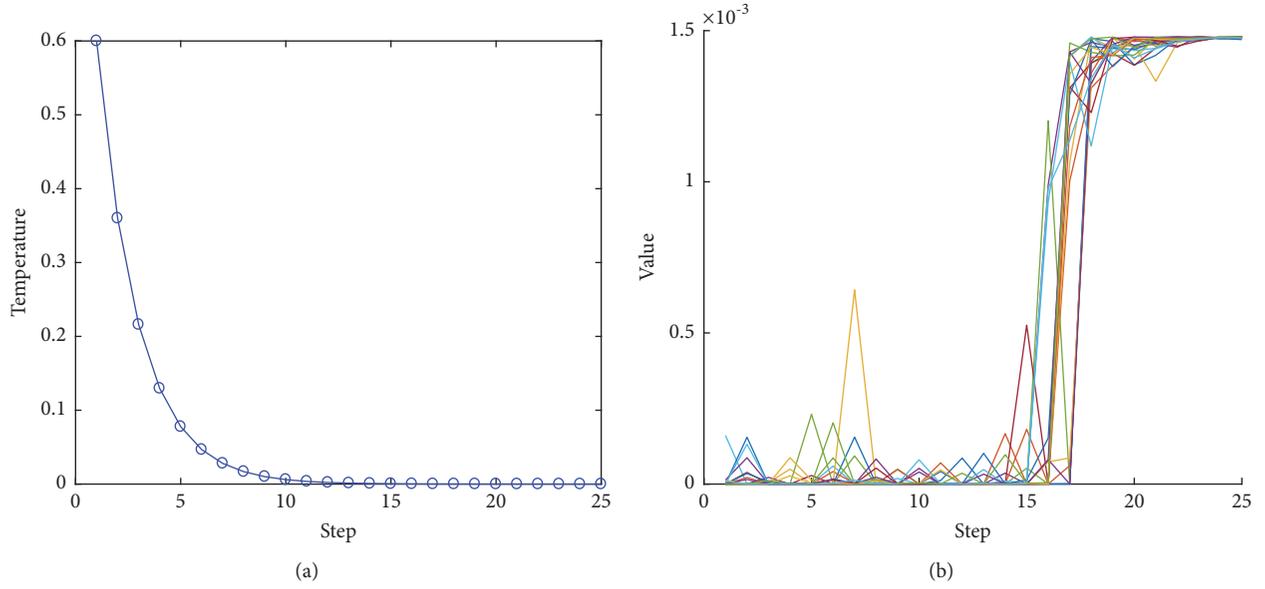


FIGURE 4: The decreasing temperature schedule (a) and the convergence paths of 20 repetitions (b) from the MTMSA algorithm.

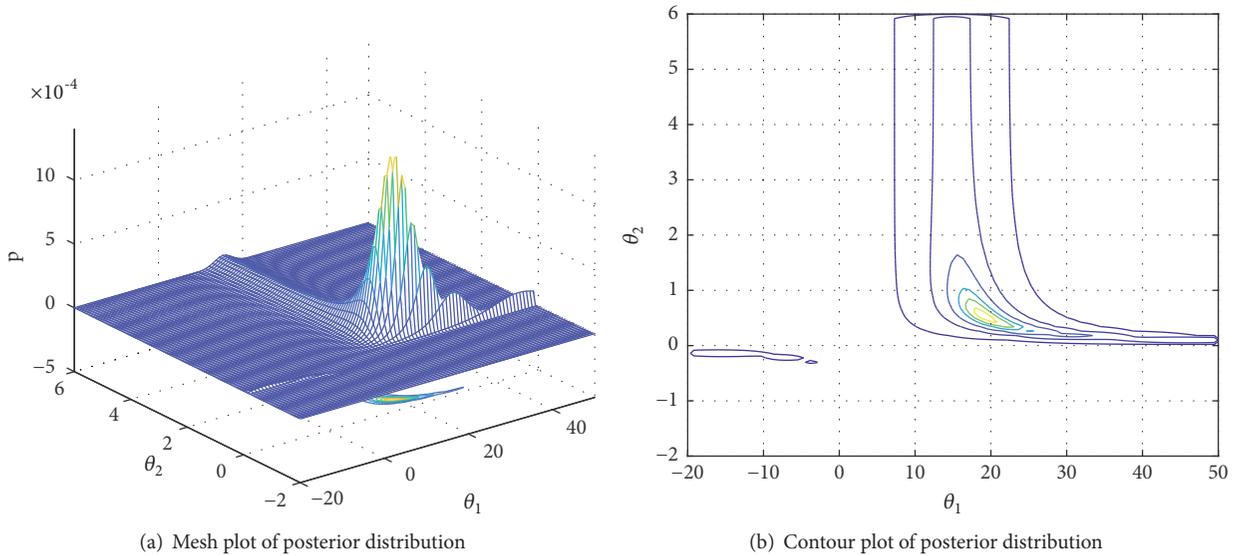


FIGURE 5: Exact mesh (a) and contour (b) plots of the posterior distribution (9).

of multivariate samples and visualizes data in high dimensional cases [33, 34]. However, the computation of HD is challenging, and the exact algorithm is often inefficient, especially when the dimension is high [35]. In this subsection, we use MTMSA to compute HD and compared MTMSA with other approximated and exact algorithms.

Given a sample data set of size  $n$   $\mathbf{X}^n = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n\}$  in  $\mathbb{R}^d$ ,  $\mathbf{x}$  is a point in  $\mathbb{R}^d$ , and the HD of  $\mathbf{x}$  with respect to (w.r.t.)  $\mathbf{X}^n$  is defined by

$$HD(\mathbf{x}, \mathbf{X}^n) = \min_{\mathbf{u} \in \mathbb{S}^{d-1}} \frac{1}{n} \# \{i \mid \mathbf{u}^T \mathbf{X}_i \geq \mathbf{u}^T \mathbf{x}, i \in \mathcal{N}\}, \quad (12)$$

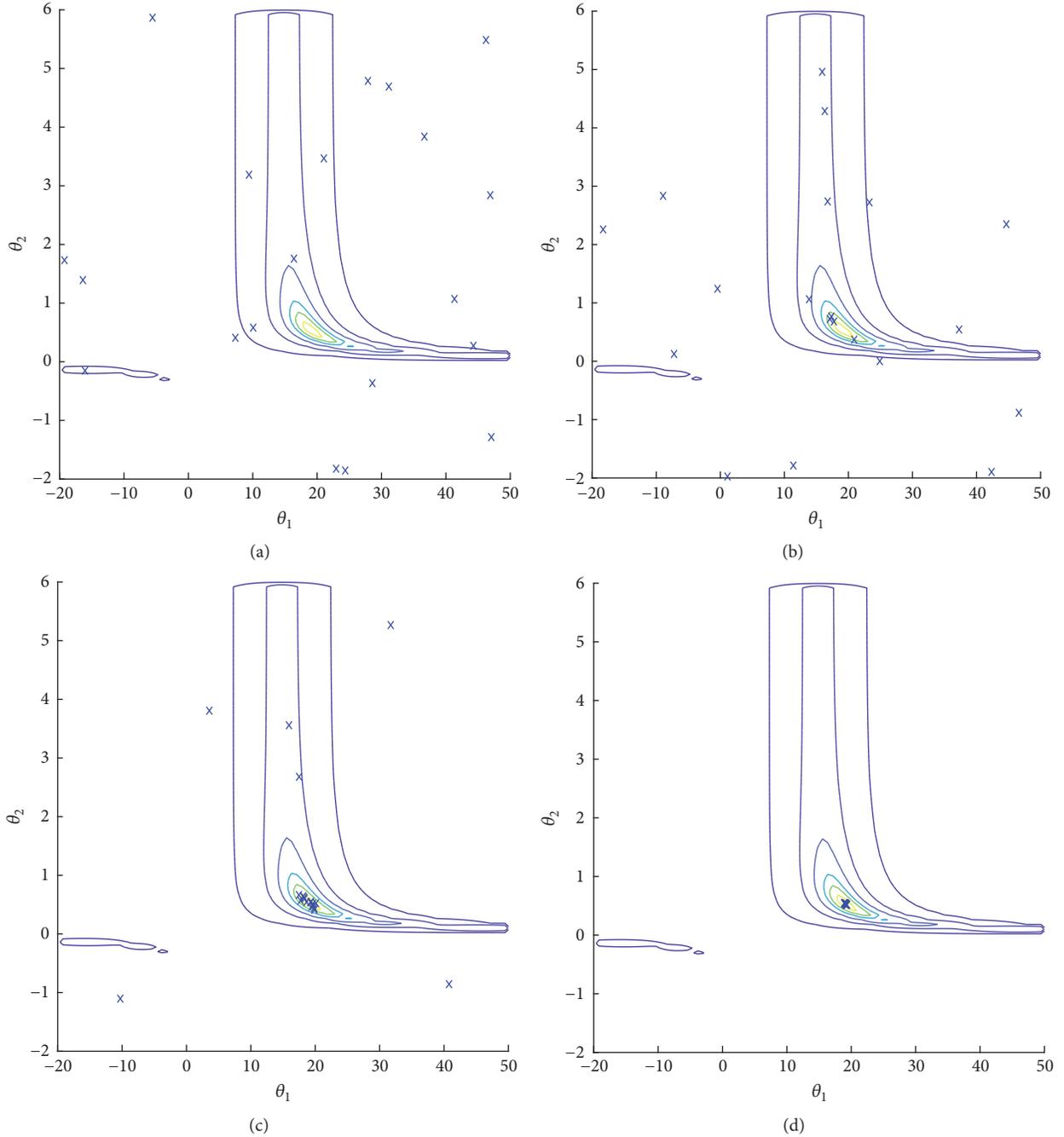


FIGURE 6: Locations of  $(\theta_1, \theta_2)$  from 20 repetitions at different temperature steps  $(0.6, 2.8 \times 10^{-4}, 1.6 \times 10^{-4}, 2.8 \times 10^{-6}, .)$ .

where  $\mathbb{S}^{d-1} = \{u \in \mathbb{R}^d \mid \|u\| = 1\}$ ,  $\mathcal{N} = \{1, 2, \dots, n\}$ , and  $\#\{\cdot\}$  denotes the counting measure. Then, the computation of HD (12) is a global optimization problem in  $\mathbb{S}^{d-1}$ .

Next, we considered a concrete data set (Table 6) obtained from [35] and can be found in the Records Office of the Laboratory School of the University of Chicago. The original data consisted of 64 subjects' scores obtained from eighth-grade levels to eleventh-grade levels. Then, we compared MTMSA with three approximated algorithms (NM, QN, and SA) and the exact algorithm from [35] for the HD computation of the first data point w.r.t. the data set.

We tested two sets of parameters for the SA algorithm. The first is  $N_{temper} = 20$ ,  $N_{mc} = 50$ ,  $T_{max} = 1$ , and  $a = 0.7$  and denoted as the SA1 algorithm. The second one is  $N_{temper} = 20$ ,  $N_{mc} = 200$ ,  $T_{max} = 1$ , and  $a = 0.7$  and denoted as the SA2 algorithm. For the MTMSA algorithm, we set the parameter to be  $N_{temper} = 20$ ,  $m = 100$ ,  $N_{mc} = 30$ ,  $T_{max} = 1$ , and  $a = 0.7$ . The three algorithms (SA1, SA2, and MTMSA) use the same decreasing temperature schedule. Then, we used the six algorithms (exact, NM, QN, SA1, SA2, and MTMSA) for this computation and repeated the computation 50 times. Figure 7 and Table 5 show the computation results.

TABLE 5: Computation results (mean, sd, MSE, consumed total CPU time, and average CPU time (in seconds)) of the 50 computations from different algorithms.

	exact	NM	QN	SA1	SA2	MTMSA
mean	0.2344	0.3653	0.3841	0.2609	0.2425	0.2344
sd	0	0.0519	0.0485	0.0243	0.0079	0
MSE	0	0.0199	0.0247	0.0013	0.0001	0
total CPU time	2450	0.06	0.05	5.7410	23.103	15.87
average CPU time	49	$+\infty$	$+\infty$	0.9570	0.9626	0.3174

TABLE 6: Concrete data set.

subject	Grade 8	Grade 9	Grade 10	Grade 11
1	1.75	2.60	3.76	3.68
2	0.90	2.47	2.44	3.43
3	0.80	0.93	0.40	2.27
4	2.42	4.15	4.56	4.21
5	-1.31	-1.31	-0.66	-2.22
6	-1.56	1.67	0.18	2.33
7	1.09	1.50	0.52	2.33
8	-1.92	1.03	0.50	3.04
9	-1.61	0.29	0.73	3.24
10	2.47	3.64	2.87	5.38
11	-0.95	0.41	0.21	1.82
12	1.66	2.74	2.40	2.17
13	2.07	4.92	4.46	4.71
14	3.30	6.10	7.19	7.46
15	2.75	2.53	4.28	5.93
16	2.25	3.38	5.79	4.40
17	2.08	1.74	4.12	3.62
18	0.14	0.01	1.48	2.78
19	0.13	3.19	0.60	3.14
20	2.19	2.65	3.27	2.73
21	-0.64	-1.31	-0.37	4.09
22	2.02	3.45	5.32	6.01
23	2.05	1.80	3.91	2.49
24	1.48	0.47	3.63	3.88
25	1.97	2.54	3.26	5.62
26	1.35	4.63	3.54	5.24
27	-0.56	-0.36	1.14	1.34
28	0.26	0.08	1.17	2.15
29	1.22	1.41	4.66	2.62
30	-1.43	0.80	-0.03	1.04
31	-1.17	1.66	2.11	1.42
32	1.68	1.71	4.07	3.30
33	-0.47	0.93	1.30	0.76
34	2.18	6.42	4.64	4.82
35	4.21	7.08	6.00	5.65
36	8.26	9.55	10.24	10.58
37	1.24	4.90	2.42	2.54
38	5.94	6.56	9.36	7.72
39	0.87	3.36	2.58	1.73

TABLE 6: Continued.

subject	Grade 8	Grade 9	Grade 10	Grade 11
40	-0.09	2.29	3.08	3.35
41	3.24	4.78	3.52	4.84
42	1.03	2.10	3.88	2.81
43	3.58	4.67	3.83	5.19
44	1.41	1.75	3.70	3.77
45	-0.65	-0.11	2.40	3.53
46	1.52	3.04	2.74	2.63
47	0.57	2.71	1.90	2.41
48	2.18	2.96	4.78	3.34
49	1.10	2.65	1.72	2.96
50	0.15	2.69	2.69	3.50
51	-1.27	1.26	0.71	2.68
52	2.81	5.19	6.33	5.93
53	2.62	3.54	4.86	5.80
54	0.11	2.25	1.56	3.92
55	0.61	1.14	1.35	0.53
56	-2.19	-0.42	1.54	1.16
57	1.55	2.42	1.11	2.18
58	0.04	0.50	2.60	2.61
59	3.10	2.00	3.92	3.91
60	-0.29	2.62	1.60	1.86
61	2.28	3.39	4.91	3.89
62	2.57	5.78	5.12	4.98
63	-2.19	0.71	1.56	2.31
64	-0.04	2.44	1.79	2.64

Figure 7 and Table 5 show that the exact algorithm consumed the most CPU time (about 2450 seconds) and obtained exact computation results (0.2344). MTMSA also obtained the exact results but consumed only 15.87 seconds. The SA algorithms (SA1 and SA2) consumed suitable CPU time (5.741 and 23.103 seconds, respectively) but obtained only 6 and 24 exact results, respectively. The results of NM and QN fell into the local optima because all of them were larger than the exact result. With regard to the average CPU time, MTMSA used only 0.3174 for the computation of one exact result, which is the least amount of time compared with the time for the other exact and approximated algorithms. Hence, MTMSA outperformed the other algorithms in this experiment example.

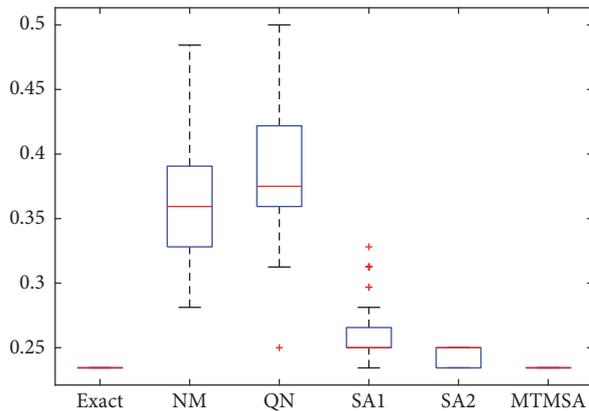


FIGURE 7: Boxplot of the results computed from the exact, NM, QN, SA1, SA2, and MTMSA algorithms.

## 5. Conclusions

We developed the MTMSA algorithm for global optimization problems in the fields of mathematical/biological sciences, engineering, Bayesian data analysis, operational research, life sciences, and so on. The MTMSA algorithm is a combination of the SA algorithm and the MTM algorithm. Using simulated and real data examples, it demonstrated that, relative to the QN and SA algorithm, the MTMSA algorithm can function with a rapidly decreasing cooling schedule while guaranteeing that the global energy minima are reached.

Several directions can be taken for future work. First, combined with the quasi-Monte Carlo method, the low-discrepancy sequences and experimental design [36–39] can be used to accelerate the convergence of the SA algorithm. Second, aside from the MTM algorithm, the MTMSA algorithm can also be implemented with several parallel interacting Markov chains to improve the SA algorithm by making full use of modern multicore computer [40, 41]. Third, we anticipate that a parallel SA algorithm can be used efficiently for variable selection in high dimensional cases [42–45] because the variable selection problem is a special case of the optimization problem. Finally, data depth [32, 33, 35, 46] is an important tool for multidimensional data analysis, but the computation of data depth in high dimensional cases is challenging. The example of half-space depth computation in Section 4 shows the advantage of the MTMSA algorithm in low dimensional case. Hence, we believe that the MTMSA algorithm can be successfully applied to compute highly complex data depths (e.g., projection and regression depths) in high dimensional cases. Further analysis along these directions would be interesting.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

## Acknowledgments

The research was partially supported by the National Natural Science Foundation of China (11501320, 71471101, and 11426143), the Natural Science Foundation of Shandong Province (ZR2014AP008), and the Natural Science Foundation of Qufu Normal University (bsqd20130114).

## References

- [1] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer, 2001.
- [2] F. Liang, C. Liu, and R. J. Carroll, *Advanced Markov Chain Monte Carlo Methods: Learning From Past Samples*, John Wiley & Sons, 2011.
- [3] S. Kirkpatrick, J. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *American Association for the Advancement of Science: Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [4] Y. Wang, L. Qi, S. Luo, and Y. Xu, "An alternative steepest direction method for the optimization in evaluating geometric discord," *Pacific Journal of Optimization*, vol. 10, no. 1, pp. 137–149, 2014.
- [5] C. Wang and Y. Wang, "A superlinearly convergent projection method for constrained systems of nonlinear equations," *Journal of Global Optimization*, vol. 44, no. 2, pp. 283–296, 2009.
- [6] Y. Wang, L. Caccetta, and G. Zhou, "Convergence analysis of a block improvement method for polynomial optimization over unit spheres," *Numerical Linear Algebra with Applications*, vol. 22, no. 6, pp. 1059–1076, 2015.
- [7] L. Qi, X. Tong, and Y. Wang, "Computing power system parameters to maximize the small signal stability margin based on min-max models," *Optimization and Engineering*, vol. 10, no. 4, pp. 465–476, 2009.
- [8] H. Chen, Y. Chen, G. Li, and L. Qi, "A semidefinite program approach for computing the maximum eigenvalue of a class of structured tensors and its applications in hypergraphs and copositivity test," *Numerical Linear Algebra with Applications*, vol. 25, no. 1, 2018.
- [9] G. Wang and X. X. Huang, "Levitin-Polyak well-posedness for optimization problems with generalized equilibrium constraints," *Journal of Optimization Theory and Applications*, vol. 153, no. 1, pp. 27–41, 2012.
- [10] G. Wang, "Levitin-Polyak well-posedness for vector optimization problems with generalized equilibrium constraints," *Pacific Journal of Optimization*, vol. 8, no. 3, pp. 565–576, 2012.
- [11] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, 1984.
- [12] R. A. Holley, S. Kusuoka, and D. W. Stroock, "Asymptotics of the spectral gap with applications to the theory of simulated annealing," *Journal of Functional Analysis*, vol. 83, no. 2, pp. 333–347, 1989.
- [13] F. Liang, C. Liu, and R. J. Carroll, "Stochastic approximation in Monte Carlo computation," *Journal of the American Statistical Association*, vol. 102, no. 477, pp. 305–320, 2007.
- [14] F. Liang, Y. Cheng, and G. Lin, "Simulated stochastic approximation annealing for global optimization with a square-root cooling schedule," *Journal of the American Statistical Association*, vol. 109, no. 506, pp. 847–863, 2014.

- [15] G. Karagiannis, B. A. Konomi, G. Lin, and F. Liang, "Parallel and interacting stochastic approximation annealing algorithms for global optimisation," *Statistics and Computing*, vol. 27, no. 4, pp. 927–945, 2017.
- [16] J. S. Liu, F. Liang, and W. H. Wong, "The multiple-try method and local optimization in metropolis sampling," *Journal of the American Statistical Association*, vol. 95, no. 449, pp. 121–134, 2000.
- [17] R. Casarin, R. Craiu, and F. Leisen, "Interacting multiple try algorithms with different proposal distributions," *Statistics and Computing*, vol. 23, no. 2, pp. 185–200, 2013.
- [18] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [19] W. Shao, G. Guo, F. Meng, and S. Jia, "An efficient proposal distribution for metropolis-hastings using a b-splines technique," *Computational Statistics & Data Analysis*, vol. 57, pp. 465–478, 2013.
- [20] W. Shao and Y. Zuo, "Simulated annealing for higher dimensional projection depth," *Computational Statistics & Data Analysis*, vol. 56, no. 12, pp. 4026–4036, 2012.
- [21] W. Shao, G. Guo, G. Zhao, and F. Meng, "Simulated annealing for the bounds of kendall's tau and spearman's rho," *Journal of Statistical Computation and Simulation*, vol. 84, no. 12, pp. 2688–2699, 2014.
- [22] Y. Luo, B. Zhu, and Y. Tang, "Simulated annealing algorithm for optimal capital growth," *Physica A: Statistical Mechanics and its Applications*, vol. 408, pp. 10–18, 2014.
- [23] O. S. Sariyer and C. Güven, "Simulated annealing algorithm for optimal capital growth," *Physica A: Statistical Mechanics and its Applications*, vol. 408, pp. 10–18, 2014.
- [24] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.
- [25] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.
- [26] R. H. Byrd, P. Lu, J. Nocedal, and C. Y. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on Scientific Computing*, vol. 16, no. 5, pp. 1190–1208, 1995.
- [27] S. C. Kou, Q. Zhou, and W. H. Wong, "Equi-energy sampler with applications in statistical inference and statistical mechanics," *The Annals of Statistics*, vol. 34, no. 4, pp. 1581–1652, 2006.
- [28] F. Liang and W. H. Wong, "Real-parameter evolutionary Monte Carlo with applications to Bayesian mixture models," *Journal of the American Statistical Association*, vol. 96, no. 454, pp. 653–666, 2001.
- [29] C. Ritter and M. A. Tanner, "Facilitating the Gibbs sampler: The Gibbs stopper and the Griddy-Gibbs sampler," *Journal of the American Statistical Association*, vol. 87, no. 419, pp. 861–868, 1992.
- [30] D. M. Bates and D. G. Watts, *Nonlinear Regression Analysis and Its Applications*, John Wiley & Sons, New York, NY, USA, 1988.
- [31] J. W. Tukey, "Mathematics and the picturing of data," *Proceedings of the International Congress of Mathematicians*, vol. 2, pp. 523–531, 1975.
- [32] Y. Zuo and R. Serfling, "General notions of statistical depth function," *The Annals of Statistics*, vol. 28, no. 2, pp. 461–482, 2000.
- [33] X. Liu, Y. Zuo, and Q. Wang, "Finite sample breakdown point of Tukey's halfspace median," *Science China Mathematics*, vol. 60, no. 5, pp. 861–874, 2017.
- [34] X. Liu, "Fast implementation of the Tukey depth," *Computational Statistics*, vol. 32, no. 4, pp. 1395–1410, 2017.
- [35] X. Liu and Y. Zuo, "Computing halfspace depth and regression depth," *Communications in Statistics—Simulation and Computation*, vol. 43, no. 5, pp. 969–985, 2014.
- [36] Z. Li, S. Zhao, and R. Zhang, "On general minimum lower order confounding criterion for s-level regular designs," *Statistics & Probability Letters*, vol. 99, pp. 202–209, 2015.
- [37] J. Wang, Y. Yuan, and S. Zhao, "Fractional factorial split-plot designs with two- and four-level factors containing clear effects," *Communications in Statistics—Theory and Methods*, vol. 44, no. 4, pp. 671–682, 2015.
- [38] S. Zhao, D. K. Lin, and P. Li, "A note on the construction of blocked two-level designs with general minimum lower order confounding," *Journal of Statistical Planning and Inference*, vol. 172, pp. 16–22, 2016.
- [39] S.-L. Zhao and Q. Sun, "On constructing general minimum lower order confounding two-level block designs," *Communications in Statistics—Theory and Methods*, vol. 46, no. 3, pp. 1261–1274, 2017.
- [40] G. Guo, W. You, G. Qian, and W. Shao, "Parallel maximum likelihood estimator for multiple linear regression models," *Journal of Computational and Applied Mathematics*, vol. 273, pp. 251–263, 2015.
- [41] G. Guo, W. Shao, L. Lin, and X. Zhu, "Parallel tempering for dynamic generalized linear models," *Communications in Statistics—Theory and Methods*, vol. 45, no. 21, pp. 6299–6310, 2016.
- [42] M. Wang and G.-L. Tian, "Robust group non-convex estimations for high-dimensional partially linear models," *Journal of Nonparametric Statistics*, vol. 28, no. 1, pp. 49–67, 2016.
- [43] M. Wang, L. Song, and G.-L. Tian, "Scad-penalized least absolute deviation regression in high-dimensional models," *Communications in Statistics—Theory and Methods*, vol. 44, no. 12, pp. 2452–2472, 2015.
- [44] G.-L. Tian, M. Wang, and L. Song, "Variable selection in the high-dimensional continuous generalized linear model with current status data," *Journal of Applied Statistics*, vol. 41, no. 3, pp. 467–483, 2014.
- [45] M. Wang and X. Wang, "Adaptive Lasso estimators for ultrahigh dimensional generalized linear models," *Statistics & Probability Letters*, vol. 89, pp. 41–50, 2014.
- [46] P. J. Rousseeuw and M. Hubert, "Regression depth," *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 388–433, 1999.

## Research Article

# Reconstruction of Medical Images Using Artificial Bee Colony Algorithm

Nur 'Afifah Rusdi , Zainor Ridzuan Yahya,  
Nurshazneem Roslan , and Wan Zuki Azman Wan Muhamad 

*Institute of Engineering Mathematics, Universiti Malaysia Perlis, 02600 Pauh, Perlis, Malaysia*

Correspondence should be addressed to Nur 'Afifah Rusdi; [afifahrusdi@unimap.edu.my](mailto:afifahrusdi@unimap.edu.my)

Received 2 March 2018; Accepted 17 June 2018; Published 9 July 2018

Academic Editor: Guillermo Cabrera-Guerrero

Copyright © 2018 Nur 'Afifah Rusdi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The goal of this study is to assess the efficiency of Artificial Bee Colony (ABC) algorithm in finding the optimal solution of curve fitting problem specifically for medical images. Data of Computed Tomography (CT) images from two different patients were collected. The procedure of curve fitting for medical images include conversion of Digital and Communications in Medicine (DICOM) images to binary images, boundary and corner point detection, parameterization, and curve reconstruction by using ABC algorithm. Then, Sum Square Error (SSE) was used to calculate the distance of the fitted Cubic Bezier curve with the boundary of the original images. Based on the calculation and parameter tuning that had been done, the smallest error of both skulls is 57.5754 and 28.8628, respectively. The finding of this study illustrated that the proposed method had efficiently produced fitted Bezier curve that resemble the original medical images. In addition, the used of Douglas Peucker algorithm helps to improve the performance of the proposed method since computational time can be minimized. This study had shown that the proposed method can be used as an alternative method in order to reconstruct or redesigned the medical images since it produces a small error. For future work, we are planning to explore and applied the ABC algorithm to reconstruct the missing part of the skull since it can reduce the time taken to produce the skull implant as well as reducing the cost of producing it.

## 1. Introduction

The process of representing an existing object geometrically in form of computer aided design (CAD) model is called reverse engineering where it referred to the process of computing a curve close to the data point set of the bitmap images [1]. The basic procedure of curve fitting process involved boundary and corner point detections, parameterization, and curve fitting [2]. Several studied had been done in this area [1–7]. However, because of the demand in the industries, new curve fitting algorithm is always being highlighted to acquire curves that satisfy different conditions.

Artificial Bee Colony (ABC) algorithm is one of the many swarm based algorithms that attract the attention of most researchers because of its ability and advantages in dealing with various problems including optimization problems, object recognition, data mining, and image clustering curve fitting [2, 4, 8]. A simple main algorithm and implementation

of ABC are found to be very effective in the studies above since they produce very good results at low computational cost. There are three main categories of research regarding ABC that had been discussed in the literatures which includes comparison and modifications, hybrid models, and applications [9]. Numbers of researches had been done to access the ability of ABC algorithm by comparing with other optimization algorithms such as Genetic Algorithm (GA), Particle Swarm Algorithm (PSO), Differential Evolution (DE), Evolutionary Algorithm (EA), and Ant Colony Optimization (ACO) [9–12]. The computational results of above studies showed that ABC outperforms the other algorithms. Not only that, but also several researches had combined ABC algorithm with other optimization algorithm so that it become more powerful. This combination is called hybridized ABC. For example, Xiaohu Shi et al. in 2010 had proposed an integrated algorithm based on ABC and PSO. Numerical result of that study shows that this hybrid algorithm is

effective and perform better than ABC or PSO alone [13]. Other than that, a novel hybrid approach had been discussed whereby the combination of GA and ABC had been used for tuning Proportional Integral (PI) speed controller in a vector-controlled Permanent Magnet Synchronous Motor (PMSM) Drive. The simulation results of this hybrid algorithm then are compared with other methods such as gradient descent method, GA, and ABC. The computational results of the hybrid approach works better compare to other methods [14].

Other than that, ABC algorithm also has been used to solve application problems such as signal processing applications, optimization problems encountered in electrical engineering, wireless sensor networks, and others [9]. Recently, numbers of researchers had explored the use of optimization algorithms in handling medical problems. For instance, the reconstruction of craniofacial fracture by using rational cubic Ball curve had been proposed in the literature. In that study, GA is used as an optimization algorithm to handle craniofacial reconstruction problem [15]. The finding shows the proposed interpolant work well for solving this problem. Not only that, but also few studies related to the medical images have been discussed by researchers [16, 17]. Hence, in this study, we will access the ability of ABC algorithm in handling curve fitting problems focusing on medical images which is a skull.

Choosing an appropriate curve also affect the performance of curve fitting. Numerous curves had been chosen by the researchers such as B Spline, cubic Spline, cubic Bézier curve, and quartic Bézier curve for solving curve fitting problems [2–7, 18–22]. Apart from all those curves, Bézier curve grabs the attention of most researchers because of its advantages and interesting properties which make it differ from other curves [21]. This paper can be divided into six sections. Section 2 discussed the Bézier curve and the Artificial Bee Colony in general. Then it was followed by explaining the process involved in Artificial Bee Colony algorithm. Next, explanation on how the algorithm works for skull reconstruction had been done in Section 5. Lastly, the paper has been concluded in Conclusion section.

## 2. Bézier Curve

Bézier curve was developed by Paul de Casteljaun in 1959 and by Pierre Étienne Bézier in about 1962. The mathematical theory of this curve is based on the concept of Bernstein polynomials [23]. Bézier curve is defined as a parametric curve  $R(t)$  which can be in any degree  $n$  with  $n + 1$  control points [3]. Generally, it can be written as

$$R(t) = \sum_{i=0}^n B_i^n(t) P_i \quad (1)$$

where  $P_i$  are called control points and  $B_i^n(t)$  are blending functions, known as Bernstein polynomial [24]. In most of curve fitting problems, cubic Bézier curve has been used as the proposed curve as well as these studies [3, 4, 18, 19, 25]. Cubic Bézier curve is generated with four control points,

$P_i(x, y)$ , where  $i$  varies from zero to three. From (1), we can illustrate cubic Bézier curve as

$$R(t) = B_0^3(t) P_0 + B_1^3(t) P_1 + B_2^3(t) P_2 + B_3^3(t) P_3 \quad (2)$$

As shown in (2), control points  $P_0$  and  $P_3$  are the end points while  $P_1$  and  $P_2$  are two middle points [4]. This study focuses on how to estimate those middle control points. In general, control point estimation is defined as a process of searching the most suitable position of  $P_1$  and  $P_2$  that will produce a better curve. Hence, Artificial Bee Colony optimization technique has been applied to optimize those points.

As has been mentioned in the introduction, numbers of researchers choose Bézier curve in dealing with reconstruction problems because of its advantages and interesting properties which make it differ from other curves. For instance, in medical application, Bézier curve had been applied for the modification of human face image for personal identification [26]. In this work, the outline of the facial features was obtained from some key points. Then, it will be connected in the form of Bézier curve before being simulated by the system. Other than that, Bézier curve has also been used in prosthesis modelling problem whereby the curve had been applied to create the contours descriptors [27]. Therefore, in this study, cubic Bézier curve will be applied to reconstruct the medical images.

## 3. Artificial Bee Colony

Artificial Bee Colony (ABC) is one of the swarm intelligence-based algorithms that grab the attention of many researchers especially in solving optimization problems. ABC was proposed by Karaboga in 2005 which has been inspired by the behaviour of honey bees [8]. This method consists of three essential components, namely, food sources, employed foragers, or known as employed bees and unemployed foragers or named as unemployed bees. Unemployed foragers can be classified into two main groups which are known as onlooker bees and scout bees [10]. All these foragers play different but important roles in ABC algorithm.

Before selecting a food source, there are several elements that need to be alert to ensure the value of the food sources such as its closeness to the hive, the taste of the nectar, the ease of extracting the nectar, and the profitability of the food sources [11]. Employed bees are assigned to carry all that information and share with the onlooker bees via waggle dance. Then, they will try to find the food source based on the information gathered by the employed bee. If the onlooker bee found a better food source, the new food will replace the old ones and onlooker bee become an employed bee again. But, if the better food source could not be found at a certain limit, then the old food source will be abandoned and the bee will become scout bee. The scout bee will go to search a new food source randomly [28]. The next section explained how ABC algorithm works which resembles the foraging behaviour of bees.

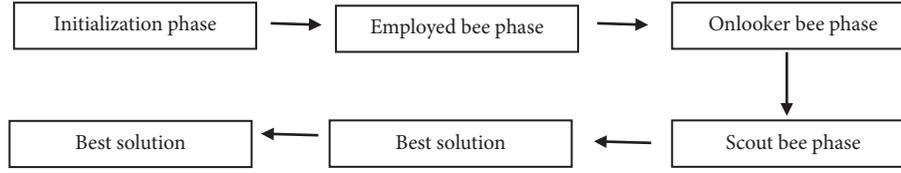


FIGURE 1: Flowchart of ABC algorithm.

#### 4. ABC Algorithm

There are four main phases involved in this algorithm, namely, initialization phase, employed bee phase, onlooker bee phase, and lastly scout bee phase. In ABC algorithm, each food source referred to as a position of a feasible solution and objective function represents the quality of the food source discovered by the bees [29]. During the initialization process, the number of employed bees are set to be equal to the number of onlooker bees and also equal to the number of food sources. Not only that, but also maximum cycle number, population size, and limit will be set. Beginning of the process, a set of initial populations of solution,  $x_i$  ( $i = 1, 2, \dots, S$ ), is generated randomly by using expression (3) as follows:

$$x_i^j = x_{\min}^j + \text{rand}(0, 1) \times (x_{\max}^j - x_{\min}^j) \quad (3)$$

where  $i$  varies from 1 to  $N$ ,  $j$  varies from 1 to  $D$ , and  $\text{rand}(0, 1)$  is a random number within the range  $[0, 1]$ .  $N$  and  $D$  are referred to as number of employed bees and dimensional of solution space, respectively. After having the initial solution, the calculation of the objective function is done by using any specific formula depending on the problem [4]. Then, the evaluation of the fitness function of each food source is calculated by using

$$\text{fit } i = \begin{cases} \frac{1}{(1 + fi)} & \text{if } fi \geq 0 \\ 1 + \text{abs}(fi) & \text{if } fi < 0 \end{cases} \quad (4)$$

where  $f_i$  is the objective function that was mentioned before. After the initialization process, each of the employed bee,  $x_i$ , explore a new food source known as  $v_{ij}$  in the neighbourhood of its present position as follows:

$$v_{i,j} = x_{i,j} + \Phi(x_{i,j} - x_{k,j}) \quad (5)$$

As shown in expression (5),  $k$  represent a random neighbourhood that has to be different from  $i$  whereby  $j$  is a randomly selected dimension and  $\Phi$  is a real number in the range of  $[-1, 1]$  [30]. Then, the fitness value for a new solution,  $v_{ij}$ , is calculated by using expression (4) and will be compared with the fitness value of the old one,  $x_i$ . This selection process is known as greedy selection whereby if  $v_{ij}$  is better than  $x_i$ ,  $v_{ij}$  will replace  $x_i$  as a new solution and the old one is discarded. Otherwise,  $x_i$  is kept as it is [31–33].

After all the employed bees completed the searching process, they will share the information as mention in the

previous section to the onlooker bees. Then, each of the onlooker bees were assigned to make a decision either to select the food source advertised by the employed bees or not. Hence, the probability of each of the food source to be selected can be calculated by using expression (6) where  $fit_i$  is the fitness value of the  $i$ th food source and  $\sum fit_i$  referred to the total fitness value [34]. Next, the modification of the food source is done by updating the position of the food source in order to find a new and better solution. This modification can be done by using expression (5) and again the greedy selection process will be applied to choose the best food source.

The last phase in ABC algorithm is known as scout bee phase. During this phase, if the food source cannot be improved through a limited cycle, then the food source is assumed to be abandoned. Hence, new food source is randomly determined by using expression (3). Notice that the value of predetermine number of cycles which is known as limit is an important control parameter in this algorithm for the purpose of abandonment [35]. At first, this limit is set to be zero. If the solution could not be improved, the trial value will be increased by one or else it will be reset to zero [36]. The process will be repeated until the termination criterion is satisfied or it achieved the maximum cycle number that has been set during the initialization process. The summarizing process of ABC algorithm is illustrated as in Figure 1.

#### 5. ABC for Curve Fitting

The first phase of curve fitting process is called boundary and corner point detections. Before it can be done, the selected images must be in form of bitmap or digitized images. For this study, the data of Computed Tomography (CT) images of skulls from two different patients were collected and the data is in Digital and Communication in Medicine (DICOM) images. Therefore, the conversion of DICOM images to bitmap images had been done. Figure 2 demonstrated the bitmap images of the skulls.

Next, the boundary of the medical images was obtained by using a built in MATLAB function called boundaries. The boundary extraction process was done to obtain the object's shape in graphical or nonscalar representation [21] and this process is important in order to preserve the complete shape of an object [1]. After boundary extraction had completed, the corner points will be detected by using SAM algorithm. Corner points are classified as points that partition the boundary into several segments. This step is important since

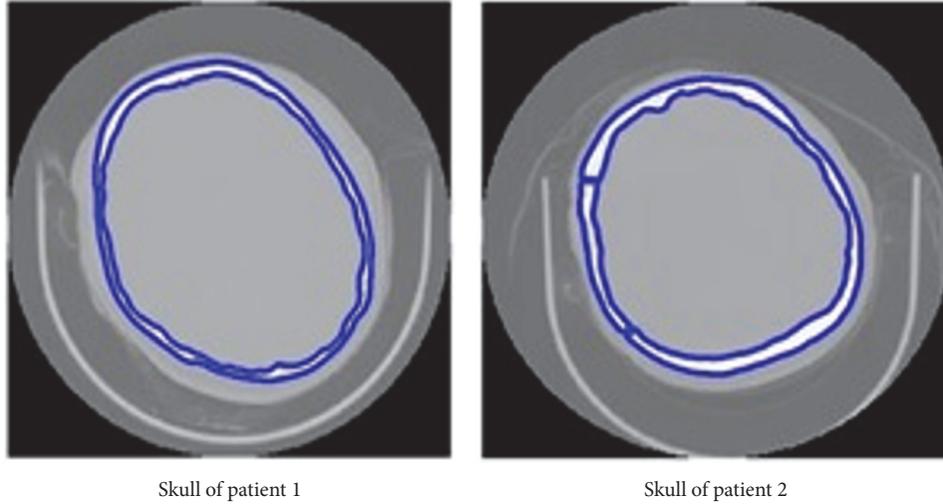


FIGURE 2: Bitmap images of skull.

it gives important clues for the shape representation analysis [21]. At this phase, Douglas Peucker algorithm had been applied to minimize the data points so that it can reduce the computational time. Boundary of medical images and detected corners of the boundary can be represented as in Table 1.

The next stage in curve fitting process is called parameterization. As is mentioned above, cubic Bézier curve is generated with four control points. The expansion of (2) can be represented in matrix form as follows:

$$R(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 3 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \quad (6)$$

Before we can estimate the intermediate points  $P_1$  and  $P_2$ , we need to estimate the values of  $t$  as in (6). For this study, chord length parameterization had been applied to estimate the value of  $t$  associated with each point by using the following.

$$t_i = \begin{cases} 0 & \text{if } i = 1 \\ \frac{|p_1p_2| + |p_2p_3| + \dots + |p_i p_{i+1}|}{|p_1p_2| + |p_2p_3| + \dots + |p_{n-1}p_n|} & \text{if } i \leq n-1 \\ 1 & \text{if } i = n \end{cases} \quad (7)$$

The climax of this process is called curve fitting. At this phase, ABC algorithm was used to estimate the two middle points  $P_1$  and  $P_2$ . There are several parameters involved in this process which are population size, number of iterations, limit, and dimension. In this study, population size is set to be 50 with 100 number of iterations. The middle points  $P_1$  and  $P_2$  consist of  $(x_1, y_1)$  and  $(x_2, y_2)$ , respectively. Therefore, the dimension is set to be four since it represented  $x_1, y_1, x_2,$  and

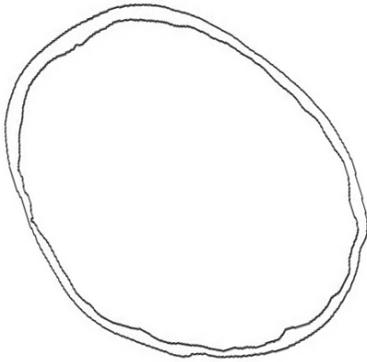
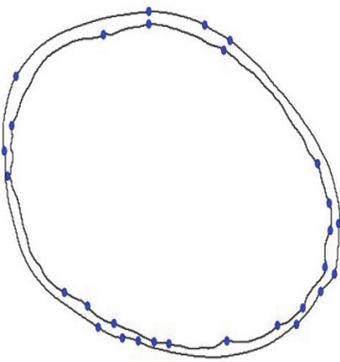
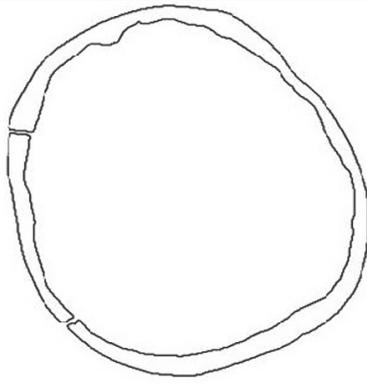
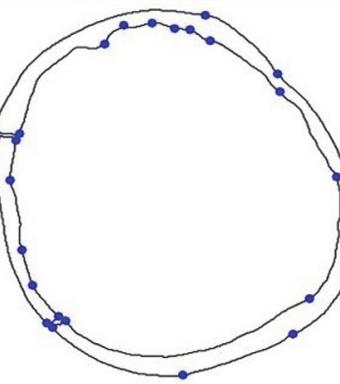
$y_2$ . Lastly, the limit is set to be 100. As has been mentioned in previous section, this limit is very important for the purpose of abundant. When the limit reached 100, if the solution is not improved, the solution will be abandoned and new solution will be searched.

After having the best value of  $P_1$  and  $P_2$ , the fitted cubic Bézier curve is obtained. Note that the main objective of this study is minimizing the distance between boundary of the original images and fitted cubic Bézier curve obtained by using the propose method. Hence, Sum Square Error (SSE) had been chosen as the objective function,  $f(x)$ , to calculate the error given by those two curves. In addition, for this study, the stopping criterion has been decided in which every process will be repeated ten times and the mean error had been recorded. Figure 3 illustrated the demonstration of the fitted cubic Bézier curve (black line) over boundary of original images (blue dot) and the error given by those two curves. The error given by skull of patients 1 and 2 is 57.5754 and 28.8628, respectively. As shown in Figure 3, most of the fitted cubic Bézier curve is on the boundary of original images. In a nutshell, it can be concluded that the proposed method can be an alternative method to reconstruct the curve.

## 6. Conclusion

This study focused on the reconstruction of medical images by using Artificial Bee Colony Algorithm. Findings of this study show that the proposed method can be used as an alternative method in solving curve fitting problems since it produced an accurate representation with the original images and less error. The use of Douglas Peucker algorithm also affects the result since it reduced the number of data points. This study can be extended by exploring how this algorithm works when dealing with reconstruction of the missing skull since it can reduce the time taken to produce the skull implant as well as reducing the cost of producing it.

TABLE 1: Boundary and corner point detections of medical images.

	Boundary	Corner
Skull of patient 1		
Skull of patient 2		

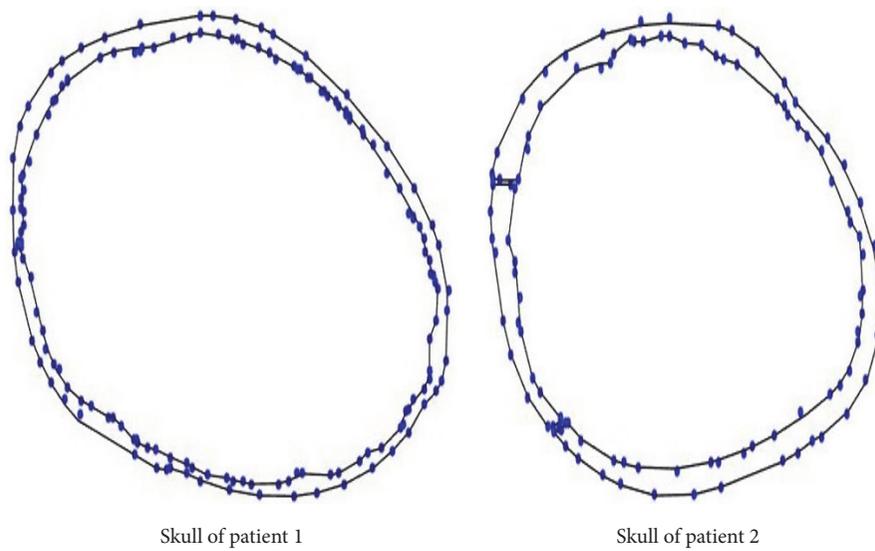


FIGURE 3: Fitted cubic Bézier curve (black line) over boundary of original images (blue dot) and the error given by those two curves.

## Data Availability

Data obtained from Digital and Communications in Medicine (DICOM) images have been used for the purposes of optimization.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This study was fully supported by Fundamental Research Grant Scheme (FRGS) 2017 [9003-00627].

## References

- [1] M. Sarfraz, M. Z. Hussain, and M. Irshad, "Reverse engineering of digital curve outlines using genetic algorithm," *International Journal of Computers*, vol. 7, no. 1, pp. 1–10, 2013.
- [2] N. A. Rusdi and Z. R. Yahya, "Reconstruction of arabic font using artificial bee colony algorithm," *ARPN Journal of Engineering and Applied Sciences*, vol. 11, no. 18, pp. 10761–10767, 2016.
- [3] N. A. Rusdi and Z. R. Yahya, "Reconstruction of generic shape with cubic Bézier using least square method," *AIP Conference Proceedings*, Article ID 050004, 2015.
- [4] N. A. Rusdi and Z. R. Yahya, "Artificial Bee Colony algorithm for curve reconstruction," *AIP Conference Proceedings*, Article ID 030077, 2016.
- [5] N. A. Rusdi and Z. R. Yahya, "Pembinaan Semula Fon Arab menggunakan Lengkung Bézier Kuartik," *Sains Malaysiana*, vol. 44, no. 8, pp. 1209–1216, 2015.
- [6] N. Roslan and Z. R. Yahya, "Reconstruction of egg shape using B-spline," in *Proceedings of The International Conference on Mathematics, Engineering and Industrial Applications 2014 (Icomeia 2014)*, p. 050016, Penang, Malaysia, 2015.
- [7] N. Roslan and Z. R. Yahya, "Reconstruction of Font with Cubic Bézier Using Differential Evolution (Pembinaan Semula Fon dengan Bézier Kubik menggunakan Evolusi Pembezaan)," *Sains Malaysiana*, vol. 44, no. 8, pp. 1203–1208, 2015.
- [8] K. Marcus, "A comprehensive review of Artificial Bee Colony Algorithm," *International Journal of Computers & Technology*, vol. 5, no. 1, pp. 15–28, 2013.
- [9] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, no. 1, pp. 21–57, 2014.
- [10] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing*, vol. 8, no. 1, pp. 687–697, 2008.
- [11] D. Karaboga and B. Akay, "A comparative study of artificial Bee colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.
- [12] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [13] X. Shi, Y. Li, H. Li, R. Guan, L. Wang, and Y. Liang, "An Integrated Algorithm Based On Artificial Bee Colony and Particle Swarm Optimization," in *Proceedings of the Sixth International Conference on Natural Computation (ICNC)*, pp. 2586–2590, IEEE, 2010.
- [14] R. K. Jatoth and A. Rajasekhar, "Speed control of PMSM by hybrid genetic artificial bee colony algorithm," in *Proceedings of the 2010 IEEE International Conference on Communication Control and Computing Technologies, ICCCT 2010*, pp. 241–246, India, October 2010.
- [15] A. Majeed, A. R. M. Piah, R. U. Gobithaasan, and Z. R. Yahya, "Craniofacial reconstruction using rational cubic ball curves," *PLoS ONE*, vol. 10, no. 4, Article ID e0122854, 2015.
- [16] A. Majeed, A. R. M. Piah, and Z. R. Yahya, "Surface reconstruction from parallel curves with application to parietal bone fracture reconstruction," *PLoS ONE*, vol. 11, no. 3, Article ID e0149921, 2016.
- [17] A. Majeed, A. R. Piah, M. Rafique, J. Yap Abdullah, and Z. A. Rajion, "NURBS curves with the application of multiple bones fracture reconstruction," *Applied Mathematics and Computation*, vol. 315, pp. 70–84, 2017.
- [18] M. Sarfraz and A. Masood, "Capturing outlines of planar images using Bézier cubics," *Computers and Graphics*, vol. 31, no. 5, pp. 719–729, 2007.
- [19] L. Shao and H. Zhou, "Curve fitting with Bezier cubics," *Graphical models and image processing*, vol. 58, no. 3, pp. 223–232, 1996.
- [20] Y. Kumar, S. K. Srivastava, A. K. Bajpai, and N. Kumar, "Development of CAD algorithms for bezier curves/surfaces independent of operating system," *WSEAS Transactions on Computers*, vol. 11, no. 6, pp. 159–169, 2012.
- [21] M. Sarfraz and N. Al-Dabbous, "Curve representation for outlines of planar images using multilevel coordinate search," *WSEAS Transactions on Computers*, vol. 12, no. 2, pp. 62–73, 2013.
- [22] K. Barkeshli, M. Mokhtari, and N. Mahdavi Amiri, "Image reconstruction of impenetrable cylinders using cubic B-splines and genetic algorithms," *IEEE Antennas and Propagation Society, AP-S International Symposium (Digest)*, vol. 2, pp. 686–689, 2001.
- [23] W. Böhm, G. Farin, and J. Kahmann, "A survey of curve and surface methods in CAGD," *Computer Aided Geometric Design*, vol. 1, no. 1, pp. 1–60, 1984.
- [24] G. Farin, "Class A Bézier curves," *Computer Aided Geometric Design*, vol. 23, no. 7, pp. 573–581, 2006.
- [25] K. Itoh and Y. Ohno, "A curve fitting algorithm for character fonts," *Electronic Publishing*, vol. 6, no. 3, pp. 195–205, 1993.
- [26] A. S. Betigeri and M. Dixit, "Modification In Human Face Image For Personal Identification," *International Journal of Applied Engineering Research and Development*, vol. 4, no. 2, pp. 13–22, 2014.
- [27] M. Rudek, Y. B. Gumiel, and O. Canciglieri, "Autonomous ct replacement method for the skull prosthesis modelling," *Facta Universitatis, series: Mechanical Engineering*, vol. 13, no. 3, pp. 283–294, 2015.
- [28] Z. Zhang, J. Lin, and Y. Shi, "Application of artificial bee colony algorithm to maximum likelihood DOA estimation," *Journal of Bionic Engineering*, vol. 10, no. 1, pp. 100–109, 2013.
- [29] A. Askarzadeh and A. Rezazadeh, "Artificial bee swarm optimization algorithm for parameters identification of solar cell models," *Applied Energy*, vol. 102, pp. 943–949, 2013.
- [30] J. Luo, Q. Liu, Y. Yang, X. Li, M. R. Chen, and W. Cao, "An artificial bee colony algorithm for multi-objective optimisation," *Applied Soft Computing*, pp. 235–251, 2017.

- [31] Y. Sönmez, "Estimation of fuel cost curve parameters for thermal power plants using the ABC algorithm," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 21, no. 1, pp. 1827–1841, 2013.
- [32] E. Zorarpacı and S. A. Özel, "A hybrid approach of differential evolution and artificial bee colony for feature selection," *Expert Systems with Applications*, vol. 62, pp. 91–103, 2016.
- [33] K. Z. Gao, P. N. Suganthan, Q. K. Pan, T. J. Chua, C. S. Chong, and T. X. Cai, "An improved artificial bee colony algorithm for flexible job-shop scheduling problem with fuzzy processing time," *Expert Systems with Applications*, vol. 65, pp. 52–67, 2016.
- [34] J.-Q. Li, Q.-K. Pan, and M. F. Tasgetiren, "A discrete artificial bee colony algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities," *Applied Mathematical Modelling*, vol. 38, no. 3, pp. 1111–1132, 2014.
- [35] D. Karaboga and B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems," in *Foundation of fuzzy logic and soft computing*, pp. 789–798, 2007.
- [36] R. Akbari, R. Hedayatzadeh, K. Ziarati, and B. Hassanizadeh, "A multi-objective artificial bee colony algorithm," *Swarm and Evolutionary Computation*, vol. 2, pp. 39–52, 2012.

## Research Article

# A Three-Term Conjugate Gradient Algorithm with Quadratic Convergence for Unconstrained Optimization Problems

Gaoyi Wu,<sup>1</sup> Yong Li,<sup>2</sup> and Gonglin Yuan <sup>3</sup>

<sup>1</sup>College of Business Administration, Guangxi University of Finance and Economics, Nanning, Guangxi 530003, China

<sup>2</sup>Department of Mathematics, Baise University, Baise, Guangxi 533000, China

<sup>3</sup>College of Mathematics and Information Science, Guangxi University, Nanning, Guangxi 530004, China

Correspondence should be addressed to Gonglin Yuan; glyuan@gxu.edu.cn

Received 4 December 2017; Accepted 30 May 2018; Published 27 June 2018

Academic Editor: Nibaldo Rodríguez

Copyright © 2018 Gaoyi Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper further studies the WYL conjugate gradient (CG) formula with  $\beta_k^{WYL} \geq 0$  and presents a three-term WYL CG algorithm, which has the sufficiently descent property without any conditions. The global convergence and the linear convergence are proved; moreover the n-step quadratic convergence with a restart strategy is established if the initial step length is appropriately chosen. Numerical experiments for large-scale problems including the normal unconstrained optimization problems and **the engineer problems (Benchmark Problems)** show that the new algorithm is competitive with the other similar CG algorithms.

## 1. Introduction

Consider the following minimization optimizations modelling:

$$\min_{x \in \mathbb{R}^n} f(x), \quad (1)$$

where  $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$  is a continuously differentiable function. The CG algorithms for (1) have the following iterative processes:

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, 2, \dots, \quad (2)$$

where  $x_k$  is the  $k$ th iterate,  $\alpha_k$  is the step length, and  $d_k$  is the search direction defined by

$$d_k = \begin{cases} -g_k + \beta_k d_{k-1}, & \text{if } k \geq 1 \\ -g_k, & \text{if } k = 0, \end{cases} \quad (3)$$

where  $g_k = \nabla f(x_k)$  is the gradient and the parameter  $\beta_k$  is a scalar determining different formulas (see [1–8], etc.). The PRP algorithm [6, 7] is one of the most effective CG algorithms and its convergence can be found (see [7, 9, 10],

etc.). Powell [9] suggested that  $\beta_k$  should not be less than zero; then many new CG formulas are proposed (see [11–17], etc.) to ensure the scalar  $\beta_k \geq 0$ . At present, there are many results obtained in CG algorithms (see [11, 18–26], etc.) and a modified weak Wolfe-Powell line search technique is presented to study open unconstrained optimization (see [27, 28]). If a restart strategy is used, the PRP algorithm is n-step quadratic convergence (see [29–31]). Li and Tian [32] proved that a three-term CG algorithm has quadratic convergence with a restart strategy under some inexact line searches and the suitable assumptions.

Recently, Wei et al. [21] proposed a new CG formula defined by  $\beta_k^{WYL} = g_k^T y_{k-1}^{WYL} / \|g_{k-1}\|^2$ , where  $y_{k-1}^{WYL} = g_k - (\|g_k\| / \|g_{k-1}\|) g_{k-1}$ ,  $g_k$  and  $g_{k+1}$  are the gradient of  $f$  at  $x_k$  and  $x_{k+1}$ , respectively, and  $\|\cdot\|$  denotes the Euclidean norm of vectors. It is easy to deduce that  $\beta_k^{WYL} = g_k^T (g_k - (\|g_k\| / \|g_{k-1}\|) g_{k-1}) / \|g_{k-1}\|^2 \geq g_k^T (\|g_k\| - (\|g_k\| / \|g_{k-1}\|) \|g_{k-1}\|) / \|g_{k-1}\|^2 = 0$ . The global convergences of the WYL algorithm with exact linear search, the Grippo-Lucidi Armijo line search, and the Wolfe-Powell line search have been established by [21, 33–35]. By restricting the parameter  $\zeta_2 < 1/4$  under the strong Wolfe-Powell linear search, the WYL algorithm can meet the sufficiently descent property.

However the quadratic convergence is still open based on this algorithm. In this paper, we mainly further research the WYL algorithm. On the base of paper [32] and the paper [21], we propose a new WYL three-term CG formula. We show that the new CG algorithm has global convergence for general functions and has the n-step quadratic convergence for uniformly convex functions with r-step restart and standard Armijo line search under appropriate conditions. The numerical results show that the new algorithm performs quite well. The main attributes of this algorithm are listed as follows.

(i) A new WYL three-term CG algorithm is introduced, which has sufficiently descent property automatically.

(ii) The global convergence, linear convergent rate, and the n-step quadratic convergence are established.

(iii) Numerical results show that this algorithm is competitive with the normal algorithm for the given problems.

This paper is arranged as follows. In Section 2, we mainly review the motivation and introduce the modified WYL algorithm. We show that the global convergence and r-step linear convergence of the new algorithm with the standard Armijo line search in Section 3. In Section 4, the n-step quadratic convergence of the given algorithm is proved. In Section 5, some numerical experiments are done.

## 2. Motivation and Algorithm

In this section, we will give motivations based on the WYL formulas. Consider the WYL search direction

$$d_k = \begin{cases} -g_k + \beta_k^{WYL} d_{k-1}, & \text{if } k \geq 1 \\ -g_k, & \text{if } k = 0, \end{cases} \quad (4)$$

and we all know that  $d_k$  is sufficiently descent by restricting the parameter  $\varsigma_2 < 1/4$  under the strong Wolfe-Powell linear search [33]. By the definition of  $d_k$  in (4), for  $k \geq 1$ , we get

$$d_k^T g_k = -\|g_k\|^2 + \frac{g_k^T y_{k-1}^{WYL}}{\|g_{k-1}\|^2} d_{k-1}^T g_k. \quad (5)$$

In order to ensure that the sufficiently descent property holds, then the first term of the above equality should be maintained. So the directive idea is to add another term to eliminate the second term of the above equality; at the same time, the conjugacy should be guaranteed. Therefore the new conjugate gradient formula called MWYL algorithm is defined by

$$d_k^{WYL} = \begin{cases} -g_k + \beta_k^{WYL} d_{k-1}^{WYL} - \theta_k^{WYL} y_{k-1}^{WYL}, & \text{if } k \geq 1 \\ -g_k, & \text{if } k = 0, \end{cases} \quad (6)$$

where  $\theta_k^{WYL} = \frac{g_k^T d_{k-1}^{WYL}}{\|g_{k-1}\|^2}$  and  $\beta_k^{WYL} = \frac{g_k^T y_{k-1}^{WYL}}{\|g_{k-1}\|^2}$ . It is easy to see that the above search direction is the normal WYL algorithm if the exact linear search is used. It is not difficult to see from (6) that  $d_k^{WYL}$  is a descent direction of  $f$  at  $x_k$ ; namely, we have

$$(d_k^{WYL})^T g_k = (-g_0)^T g_0 = -\|g_0\|^2, \quad (k = 0) \quad (7)$$

and

$$\begin{aligned} (d_k^{WYL})^T g_k &= (-g_k + \beta_k^{WYL} d_{k-1}^{WYL} - \theta_k^{WYL} y_{k-1}^{WYL})^T g_k \\ &= -\|g_k\|^2 + \frac{g_k^T y_{k-1}^{WYL}}{\|g_{k-1}\|^2} \cdot (d_{k-1}^{WYL})^T g_k \\ &\quad - \frac{g_k^T d_{k-1}^{WYL}}{\|g_{k-1}\|^2} \cdot (y_{k-1}^{WYL})^T g_k = -\|g_k\|^2, \end{aligned} \quad (8)$$

( $k \geq 1$ ).

Moreover we obtain

$$(d_k^{WYL})^T g_k = -\|g_k\|^2, \quad k \geq 0 \quad (9)$$

and

$$\|g_k\| \leq \|d_k^{WYL}\|. \quad (10)$$

Now we list some linear search techniques that will be used in the following sections.

(i) The exact line search is to find  $\alpha_k$  such that the function is minimized along the direction  $d_k$ , that is,  $\alpha_k$ , satisfying

$$f(x_k + \alpha_k d_k) = \min_{\alpha \geq 0} f(x_k + \alpha d_k) \quad (11)$$

(ii) The Armijo line search is to find a step length  $\alpha_k = \max\{\rho^i \mid i = 0, 1, 2, \dots\}$  which satisfies

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \varsigma_1 \alpha_k g_k^T d_k, \quad (12)$$

where  $\rho \in (0, 1)$  and  $\varsigma_1 \in (0, 1/2)$ .

(iii) The Wolfe line search conditions are

$$\begin{aligned} f(x_k + \alpha_k d_k) &\leq f(x_k) + \varsigma_1 \alpha_k g_k^T d_k \\ g(x_k + \alpha_k d_k)^T d_k &\geq \varsigma_2 g_k^T d_k, \end{aligned} \quad (13)$$

where  $0 < \varsigma_1 < 1/2$  and  $\varsigma_1 < \varsigma_2 < 1$ .

In the following we will give the MWYL algorithm.

*Algorithm 1* (a modified WYL three-terms CG algorithm, called MWYL).

Step 0: Choose an initial point  $x_0 \in \mathfrak{R}^n$ ,  $\varsigma_1 \in (0, 1/2)$ ,  $\rho \in [0, 1)$ ,  $\epsilon \in [0, 1)$ ; let  $d_0 = -g_0$ ,  $k := 0$ .

Step 1: If  $\|g_k\| \leq \epsilon$ , stop. Otherwise go to the next step.

Step 2: Compute step size  $\alpha_k$  by the Armijo line search or Wolfe line search.

Step 3: Let  $x_{k+1} = x_k + \alpha_k d_k^{WYL}$ . If  $\|g_{k+1}\| \leq \epsilon$ , then stop.

Step 4: Compute the search direction using (6).

Step 5: Let  $k := k + 1$ . Go to step 2.

### 3. Convergence of Algorithm 1

In this part, we will prove the global convergence and the r-linear convergence of the MWYL algorithm with the Armijo line search and Wolfe line search. The following assumptions are required.

*Assumption i.* The level set  $\Omega = \{x \mid f(x) \leq f(x_1)\}$  is bounded and, in some neighborhood  $N$  of  $\overline{co}\Omega$ ,  $f$  is continuously differentiable and bounded below, and its gradient is globally Lipschitz continuous; namely, there exists a constant  $L > 0$  such that

$$\|g(x) - g(y)\| \leq L\|x - y\|, \quad \forall x, y \in N, \quad (14)$$

where  $\overline{co}\Omega$  is the closed convex hull of  $\Omega$ .

Now we establish the global convergence of Algorithm 1.

**Theorem 2.** *Let Assumption i hold and the sequence  $\{x_k\}$  be generated by Algorithm 1. Then the relation*

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0 \quad (15)$$

holds.

*Proof.* We will prove this theorem by contradiction. Suppose that (15) does not hold, then, for all  $k$ , there exists a constant  $\varepsilon_1 > 0$  satisfying

$$\|g_k\| \geq \varepsilon_1. \quad (16)$$

Using (9) and (12), if  $f$  is bounded from below, it is not difficult to get

$$\sum_{k=0}^{\infty} \alpha_k^2 \|d_k^{WYL}\|^2 < \infty. \quad (17)$$

In particular, we have

$$\lim_{k \rightarrow \infty} \alpha_k \|d_k^{WYL}\| = 0. \quad (18)$$

If  $\lim_{k \rightarrow \infty} \alpha_k > 0$ , by (9) and (18), we obtain  $\lim_{k \rightarrow \infty} \inf \|g_k\| = 0$ . This contradicts (16); then (15) holds.

Otherwise if  $\lim_{k \rightarrow \infty} \alpha_k = 0$ . Then there exists an infinite index set  $N$  satisfying

$$\lim_{k \in N, k \rightarrow \infty} \alpha_k = 0. \quad (19)$$

By Step 2 of Algorithm 1, when  $k$  is sufficiently large,  $\alpha_k \rho^{-1}$  does not satisfy (12), which implies that

$$f(x_k + \alpha_k \rho^{-1} d_k^{WYL}) - f(x_k) > -\varsigma_1 \rho^{-2} \alpha_k^2 \|d_k^{WYL}\|^2. \quad (20)$$

By (16), similar to the proof of Lemma 2.1 in [36], it is easy to deduce that there exists a constant  $\varrho > 0$  such that

$$\|d_k^{WYL}\| \leq \varrho, \quad \forall k. \quad (21)$$

Using (21), (9), and the mean-value theorem, we have

$$\begin{aligned} & f(x_k + \alpha_k \rho^{-1} d_k^{WYL}) - f(x_k) \\ &= \rho^{-1} \alpha_k g(x_k + \xi_0 \rho^{-1} \alpha_k d_k^{WYL})^T d_k^{WYL} \\ &= \rho^{-1} \alpha_k g_k^T d_k^{WYL} \\ &+ \rho^{-1} \alpha_k (g(x_k + \xi_0 \rho^{-1} \alpha_k d_k^{WYL}) - g_k)^T d_k^{WYL} \\ &\leq \rho^{-1} \alpha_k g_k^T d_k^{WYL} + M \rho^{-2} \alpha_k^2 \|d_k^{WYL}\|^2, \end{aligned} \quad (22)$$

where  $\xi_0 \in (0, 1)$  and the last inequality follows (26). Combining with (20), for all  $k \in N$  sufficiently large, we obtain

$$\|g_k\|^2 \leq \rho^{-1} (M + \varsigma_1) \alpha_k \|d_k^{WYL}\|^2. \quad (23)$$

By (21) and  $\lim_{k \rightarrow \infty} \alpha_k = 0$ , then the above inequality implies that  $\lim_{k \in N, k \rightarrow \infty} \|g_k\| = 0$ . This is a contradiction too. The proof is complete.  $\square$

In the next, we will prove the linear convergence of the sequence  $\{x_k\}$  by the MWYL algorithm with the Armijo or Wolfe line search. The following assumption is further needed.

*Assumption ii.*  $f$  is twice continuously differentiable and the uniformly convex function. In other words, there are positive constants  $M \geq m > 0$  such that

$$m \|d\|^2 \leq d^T \nabla^2 f(x) d \leq M \|d\|^2, \quad \forall x, d \in \mathfrak{R}^n, \quad (24)$$

where  $\nabla^2 f(x)$  denotes the Hessian matrix of  $f$  at  $x$ .

It is not difficult to see that, under the Assumption ii,  $\nabla^2 f(x)$  is continuous and  $g$  is Lipschitz continuous and problem (1) has a unique solution  $x^*$  which satisfies

$$\frac{1}{2} m \|x - x^*\|^2 \leq f(x) - f(x^*) \leq \frac{1}{2} M \|x - x^*\|^2, \quad (25)$$

$\forall x \in \mathfrak{R}^n$

and

$$m \|x - x^*\| \leq \|g_k\| \leq M \|x_k - x^*\|. \quad (26)$$

**Lemma 3.** *Let Assumption ii hold and the sequence  $\{x_k\}$  be generated by the MWYL algorithm with the Armijo or Wolfe line search, one has*

$$\begin{aligned} & -\sum_{k=0}^{\infty} g_k^T s_k^{WYL} < \infty, \\ & \sum_{k=0}^{\infty} \|s_k^{WYL}\| < \infty, \\ & c_1 \alpha_k \|d_k^{WYL}\|^2 \leq -g_k^T d_k^{WYL}, \end{aligned} \quad (27)$$

where  $s_k^{WYL} = x_{k+1} - x_k$  and  $c_1 = (1/2)(1 - \sigma_1)^{-1} m$ . In addition, if the Wolfe line search is used, the following holds:

$$-g_k^T d_k^{WYL} \leq (1 - \varsigma_2)^{-1} M \alpha_k \|d_k^{WYL}\|^2. \quad (28)$$

*Proof.* We have from line search (12) that

$$\begin{aligned} f(x_0) - f(\hat{x}_{\hat{k}}) &= \sum_{k=0}^{\hat{k}-1} (f(x_k) - f(x_{k+1})) \\ &\geq -\sum_{k=0}^{\hat{k}-1} \sigma_1 g_k^T s_k^{WYL} > 0 \end{aligned} \quad (29)$$

hold for any  $\hat{k} > 1$ , because the objective function  $f$  is uniformly convex of Assumption ii,  $f$  is bounded below, so the inequality  $-\sum_{k=0}^{\hat{k}-1} g_k^T s_k^{WYL} < +\infty$  holds. Combining the Taylor theorem and Assumption ii, we obtain

$$\begin{aligned} f(x_{k+1}) &= f(x_k) + g_k^T s_k^{WYL} \\ &\quad + \frac{1}{2} (s_k^{WYL})^T \nabla^2 f(\xi_k) s_k^{WYL}, \end{aligned} \quad (30)$$

where  $\xi_k$  belong to the segment  $[x_k, x_{k+1}]$ . Therefore, we get

$$\begin{aligned} \sum_{k=0}^{\hat{k}-1} (f(x_{k+1}) - f(x_k)) - \sum_{k=0}^{\hat{k}-1} g_k^T s_k^{WYL} \\ = \sum_{k=0}^{\hat{k}-1} \frac{1}{2} (s_k^{WYL})^T \nabla^2 f(\xi_k) s_k^{WYL} \geq \sum_{k=0}^{\hat{k}-1} \frac{1}{2} m \|s_k^{WYL}\|^2. \end{aligned} \quad (31)$$

By the inequalities  $\sum_{k=0}^{\hat{k}-1} (f(x_{k+1}) - f(x_k)) < 0$  and  $-\sum_{k=0}^{\hat{k}-1} g_k^T s_k^{WYL} < +\infty$ , we get  $\sum_{k=0}^{\infty} \|s_k^{WYL}\|^2 < +\infty$ . Using (12), (30), and Assumption ii, we obtain

$$\begin{aligned} \varsigma_1 g_k^T s_k^{WYL} &\geq f(x_{k+1}) - f(x_k) \\ &= g_k^T s_k^{WYL} + \frac{1}{2} (s_k^{WYL})^T \nabla^2 f(\xi_k) s_k^{WYL} \\ &\geq g_k^T s_k^{WYL} + \frac{1}{2} m \|s_k^{WYL}\|^2, \end{aligned} \quad (32)$$

which includes  $-g_k^T s_k^{WYL} \geq (m/2(1 - \varsigma_1)) \|s_k^{WYL}\|^2$ .

It is not difficult to get that

$$-g_k^T d_k^{WYL} \geq \frac{m}{2(1 - \varsigma_1)} \cdot \alpha_k \|d_k^{WYL}\|^2 = c_1 \alpha_k \|d_k^{WYL}\|^2. \quad (33)$$

By the second inequality of (13), we get

$$\begin{aligned} g(x_k + \alpha_k d_k^{WYL}) \\ = \left[ g_k + \int_0^1 \nabla^2 f(x_k + \tau \alpha_k d_k^{WYL}) d\tau \cdot \alpha_k d_k^{WYL} \right]^T \\ \cdot d_k^{WYL} \geq \varsigma_2 g_k^T d_k^{WYL} \end{aligned} \quad (34)$$

and

$$\begin{aligned} \alpha_k (d_k^{WYL})^T \int_0^1 \nabla^2 f(x_k + \tau \alpha_k d_k^{WYL}) d\tau \cdot d_k^{WYL} \\ \geq (\varsigma_2 - 1) g_k^T d_k^{WYL}. \end{aligned} \quad (35)$$

By Assumption ii, we obtain  $-g_k^T d_k^{WYL} \leq (1 - \varsigma_2)^{-1} M \alpha_k \|d_k^{WYL}\|^2$ . This completes the proof.  $\square$

**Lemma 4.** Let the sequence  $\{x_k\}$  be generated by the MWYL algorithm with the Armijo or Wolfe line search and Assumption ii hold; then there is a constant  $c > 0$  such that

$$\alpha_k > c, \quad \forall k \geq 0. \quad (36)$$

*Proof.* Set

$$G_{k-1} = \int_0^1 \nabla^2 f(x_{k-1} + \tau s_{k-1}) d\tau, \quad (37)$$

where  $s_{k-1}^{WYL} = x_k - x_{k-1} = \alpha_{k-1} d_{k-1}^{WYL}$ . By the mean-value theorem, we have

$$y_k = g_k - g_{k-1} = G_{k-1} s_{k-1} = \alpha_{k-1} G_{k-1} d_{k-1}^{WYL} \quad (38)$$

and

$$\begin{aligned} |y_{k-1}^{WYL}| &= \left\| g_k - \frac{\|g_k\|}{\|g_{k-1}\|} \cdot g_{k-1} \right\| \\ &= \left\| g_k - g_{k-1} + g_{k-1} - \frac{g_k}{g_{k-1}} \cdot g_{k-1} \right\| \\ &\leq \|g_k - g_{k-1}\| + \left| \|g_{k-1}\| - \|g_k\| \right| \\ &\leq 2 \|g_k - g_{k-1}\| \leq 2\alpha_{k-1} \cdot \|G_{k-1}\| \cdot \|d_{k-1}^{WYL}\| \end{aligned} \quad (39)$$

Therefore, by (9), (39), Lemma 3, and the Assumption ii, we get

$$\begin{aligned} |\beta_k^{WYL}| &= \frac{\|g_k\| \cdot \|y_{k-1}^{WYL}\|}{\|g_{k-1}\|^2} \leq \frac{2\alpha_k \|g_k\| \cdot \|G_{k-1}\| \|d_{k-1}^{WYL}\|}{c_1 \alpha_k \|d_{k-1}^{WYL}\|^2} \\ &\leq \frac{2c_1^{-1} M \|g_k\|}{\|d_{k-1}^{WYL}\|} \end{aligned} \quad (40)$$

and

$$|\theta_k^{WYL}| = \frac{|g_k^T \cdot d_{k-1}^{WYL}|}{\|g_{k-1}\|^2} = \frac{|g_k^T \cdot d_{k-1}^{WYL}|}{-g_{k-1}^T \cdot d_{k-1}^{WYL}} \leq \frac{2c_1^{-1} M \|g_k\|}{\|d_{k-1}^{WYL}\|} \quad (41)$$

By the above conclusion, (6), and the Lipschitz continuity of  $g$ , we get

$$\begin{aligned} \|d_k^{WYL}\| &\leq \|g_k\| + |\beta_k^{WYL}| \cdot \|d_{k-1}^{WYL}\| + |\theta_k^{WYL}| \cdot \|y_{k-1}^{WYL}\| \\ &\leq \|g_k\| + 2Mc_1^{-1} \|g_k\| + 2Mc_1^{-1} \|g_k\| \\ &\leq (1 + 4Mc_1^{-1}) \cdot \|g_k\| \end{aligned} \quad (42)$$

If the Armijo line search is used, using the line search rule, if  $\alpha_k \neq 1$ , then  $\alpha'_k = \alpha_k \rho^{-1}$  will not satisfy line search condition (12). Namely,

$$f(x_k + \alpha'_k d_k^{WYL}) - f(x_k) > \varsigma_1 \alpha'_k g_k^T d_k^{WYL}. \quad (43)$$

Using the mean-value theorem and the above inequality, there exists  $\mu_k \in (0, 1)$  satisfying

$$\begin{aligned}
 & \varsigma_1 \alpha'_k g_k^T d_k^{WY\bar{L}} < f(x_k + \alpha'_k d_k^{WY\bar{L}}) - f(x_k) \\
 & = \alpha'_k g(x_k + \mu_k \alpha'_k d_k^{WY\bar{L}})^T d_k^{WY\bar{L}} \\
 & = \alpha'_k (g(x_k + \mu_k \alpha'_k d_k^{WY\bar{L}}) - g(x_k))^T d_k^{WY\bar{L}} \\
 & \quad + \alpha'_k g_k^T d_k^{WY\bar{L}} \\
 & = \mu_k (\alpha'_k)^2 (d_k^{WY\bar{L}})^T \int_0^1 \nabla^2 f(x_k + \tau \mu_k \alpha'_k d_k^*) d\tau \\
 & \quad \cdot d_k^{WY\bar{L}} + \alpha'_k g_k^T d_k^{WY\bar{L}} \\
 & \leq (\alpha'_k)^2 M \cdot \|d_k^{WY\bar{L}}\|^2 + \alpha'_k g_k^T d_k^{WY\bar{L}}.
 \end{aligned} \tag{44}$$

Thus, by the above conclusion and (42), we get

$$\begin{aligned}
 \alpha_k & = \rho \alpha'_k \geq -\frac{(1 - \varsigma_1) \rho}{M} \cdot \frac{g_k^T d_k^{WY\bar{L}}}{\|d_k^{WY\bar{L}}\|^2} \\
 & = \frac{(1 - \varsigma_1) \rho}{M} \cdot \frac{\|g_k\|^2}{\|d_k^{WY\bar{L}}\|^2} \\
 & \geq M^{-1} (1 - \varsigma_1) \rho (1 + 4M\varsigma_1^{-1})^{-2} \equiv c^*,
 \end{aligned} \tag{45}$$

and letting  $c = \min\{1, c^*\}$ , we have (36). If the Wolfe line search is used, from the second inequality of (13), we obtain

$$\begin{aligned}
 M\alpha_k \|d_k^{WY\bar{L}}\|^2 & \geq (g(x_k + \alpha_k d_k^{WY\bar{L}}) - g_k)^T d_k^{WY\bar{L}} \\
 & \geq -(1 - \varsigma_2) g_k^T d_k^{WY\bar{L}}.
 \end{aligned} \tag{46}$$

By similar way to that for the Armijo line search, we can find a lower positive bound of  $\alpha_k$ ; the proof is completed.  $\square$

Similar to [32], It is easy to get the r-linear convergence theorem of Algorithm 1. So we only state it as follows but omit the proof.

**Theorem 5.** *Let Assumption ii hold,  $x^*$  be the unique solution of (1), and the sequence  $\{x_k\}$  be generated by the MWYL algorithm with the Armijo or Wolfe line search. Then there are constants  $a > 0$  and  $r \in (0, 1)$  satisfying*

$$\|x_k - x^*\| \leq ar^k. \tag{47}$$

*Proof.* By (12) or the first relation of (13), we get

$$\begin{aligned}
 f(x_{k+1}) - f(x^*) & \leq f(x_k) - f(x^*) + \varsigma_1 \alpha_k d_k^T g_k \\
 & = f(x_k) - f(x^*) - \varsigma_1 \alpha_k \|g_k\|^2 \\
 & \leq f(x_k) - f(x^*) - c\varsigma_1 m^2 \alpha_k \|x_k - x^*\|^2
 \end{aligned}$$

$$\begin{aligned}
 & \leq f(x_k) - f(x^*) - \frac{2c\varsigma_1 m^2}{M} [f(x_k) - f(x^*)] \\
 & = \left[1 - \frac{2c\varsigma_1 m^2}{M}\right] [f(x_k) - f(x^*)],
 \end{aligned} \tag{48}$$

where the first equality follows (9), the second inequality follows (26), and the last inequality (25). Setting  $r = [1 - 2c\varsigma_1 m^2/M]^{1/2} \in (0, 1)$  generates

$$\begin{aligned}
 f(x_{k+1}) - f(x^*) & \leq r^2 [f(x_k) - f(x^*)] \leq \dots \\
 & \leq r^{2k} [f(x_0) - f(x^*)].
 \end{aligned} \tag{49}$$

By (25) again, we have

$$\begin{aligned}
 \|x_k - x^*\| & \leq \frac{2}{m} [f(x_k) - f(x^*)] \\
 & \leq \frac{2}{m} [f(x_0) - f(x^*)] r^{2k},
 \end{aligned} \tag{50}$$

and this relation shows that the proof is complete.  $\square$

#### 4. The Restart MWYL Algorithm's N-Step Quadratic Convergence

Setting  $\bar{\alpha}_k$  as exact line search step length, then

$$g(x_k + \bar{\alpha}_k d_k^{WY\bar{L}})^T d_k^{WY\bar{L}} = 0 \tag{51}$$

holds. Thus,

$$\begin{aligned}
 \|g_k\|^2 & = -g_k^T d_k^{WY\bar{L}} \\
 & = (g(x_k + \bar{\alpha}_k d_k^{WY\bar{L}}) - g(x_k))^T d_k^{WY\bar{L}} \\
 & = \bar{\alpha}_k (d_k^{WY\bar{L}})^T \bar{G}_k d_k^{WY\bar{L}},
 \end{aligned} \tag{52}$$

where  $\bar{G}_k = \int_0^1 \nabla^2 f(x_k + \tau \bar{\alpha}_k d_k^{WY\bar{L}}) d\tau$ . It is feasible to use the initial step length of the Armijo or Wolfe line search as an approximation of  $\bar{\alpha}_k$ , where  $\bar{\alpha}_k$  is defined by

$$\bar{\alpha}_k \equiv \frac{\|g_k\|^2}{(d_k^{WY\bar{L}})^T \bar{G}_k d_k^{WY\bar{L}}} \tag{53}$$

$$\approx \frac{\epsilon_k \|g_k\|^2}{(d_k^{WY\bar{L}})^T (g(x_k + \epsilon_k d_k^{WY\bar{L}}) - g(x_k))} \equiv \gamma_k,$$

where the integer sequence  $\{\epsilon_k\} \rightarrow 0$  as  $k \rightarrow \infty$ . If  $f$  is a quadratic function, then  $\gamma_k$  and  $\bar{\alpha}_k$  are consistent; namely,

$$\begin{aligned}
|\bar{\alpha}_k - \gamma_k| &= \left| \frac{\|g_k\|^2}{(d_k^{WYL})^T \bar{G}_k d_k^{WYL}} - \frac{\epsilon_k \|g_k\|^2}{(d_k^{WYL})^T (g(x_k + \epsilon_k d_k^{WYL}) - g(x_k))} \right| \\
&= \frac{\left| (d_k^{WYL})^T (g(x_k + \epsilon_k d_k^{WYL}) - g_k) \|g_k\|^2 - (d_k^{WYL})^T \bar{G}_k d_k^{WYL} \epsilon_k \|g_k\|^2 \right|}{\left( (d_k^{WYL})^T \bar{G}_k d_k^{WYL} \right) (d_k^{WYL})^T (g(x_k + \epsilon_k d_k^{WYL}) - g(x_k))} \leq \frac{o(\epsilon_k \|d_k^{WYL}\|^2) \cdot \|g_k\|^2}{m^2 \epsilon_k \|d_k^{WYL}\|^4} \rightarrow 0.
\end{aligned} \tag{54}$$

The above discussions can also be found in [32]; in fact, our ideas are motivated by this paper partly. The following Theorem 6 will show that, for sufficiently large  $k$ , the inexact line search step  $\gamma_k$  which is defined by (53) satisfies the Armijo and Wolfe conditions.

**Theorem 6.** *Let sequence  $\{x_k\}$  be generated by the MWYL algorithm and Assumption ii hold. Then, when  $k$  is sufficiently large,  $\gamma_k$  satisfies the Armijo and Wolfe conditions.*

*Proof.* Let  $A_k = \int_0^1 \nabla^2 f(x_k + \tau \epsilon_k d_k^{WYL}) d\tau$ , using Assumption ii and (10), we have

$$\gamma_k = \frac{\|g_k\|^2}{(d_k^{WYL})^T A_k d_k^{WYL}} \geq \frac{\|g_k\|^2}{M \|d_k^{WYL}\|^2} \tag{55}$$

and

$$\gamma_k = \frac{\|g_k\|^2}{(d_k^{WYL})^T A_k d_k^{WYL}} \leq \frac{\|g_k\|^2}{m \|d_k^{WYL}\|^2} \leq \frac{1}{m}. \tag{56}$$

Using  $\{d_k^{WYL}\} \rightarrow 0$ , Assumption ii, (47), and (55), we get

$$\begin{aligned}
&f(x_k + \gamma_k d_k^{WYL}) \\
&= f(x_k) + \gamma_k g_k^T d_k^{WYL} + \frac{1}{2} \gamma_k^2 (d_k^{WYL})^T A_k d_k^{WYL} \\
&\quad + \gamma_k^2 o(\|d_k^{WYL}\|^2) \\
&= f(x_k) + \frac{1}{2} \gamma_k g_k^T d_k^{WYL} + \gamma_k^2 o(\|d_k^{WYL}\|^2) \\
&= f(x_k) + \varsigma_1 \gamma_k g_k^T d_k^{WYL} - \left(\frac{1}{2} - \varsigma_1\right) \gamma_k \|g_k\|^2 \\
&\quad + \gamma_k^2 o(\|d_k^{WYL}\|^2) \\
&\leq f(x_k) + \varsigma_1 \gamma_k g_k^T d_k^{WYL} \\
&\quad - \left(\frac{1}{2} - \varsigma_1\right) \frac{(1 + 2Mc_1^{-1})^{-4}}{M} \|d_k^{WYL}\|^2 \\
&\quad + \gamma_k^2 o(\|d_k^{WYL}\|^2)
\end{aligned} \tag{57}$$

For  $k$  is sufficiently large, we have

$$f(x_k + \gamma_k d_k^{WYL}) \leq f(x_k) + \varsigma_1 \gamma_k g_k^T d_k^{WYL}. \tag{58}$$

When  $k$  is sufficiently large,  $\alpha_k = \gamma_k$  satisfies the Armijo condition. Setting  $\bar{A}_k = \int_0^1 \nabla^2 f(x_k + \tau \gamma_k d_k^{WYL}) d\tau$ , we get

$$\begin{aligned}
&g(x_k + \gamma_k d_k) d_k^{WYL} - \varsigma_2 g_k^T d_k^{WYL} \\
&= (g(x_k + \gamma_k d_k^{WYL}) - g_k)^T d_k^{WYL} \\
&\quad + (1 - \varsigma_2) g_k^T d_k^{WYL} \\
&= \gamma_k (d_k^{WYL})^T \bar{A}_k d_k^{WYL} - (1 - \varsigma_2) \|g_k\|^2 \\
&= \left( \frac{(d_k^{WYL})^T \bar{A}_k d_k^{WYL}}{(d_k^{WYL})^T A_k d_k^{WYL}} - 1 \right) \|g_k\|^2 + \varsigma_2 \|g_k\|^2 \\
&\leq f(x_k) + \varsigma_1 \gamma_k g_k^T d_k^{WYL} \\
&\quad - \left(\frac{1}{2} - \varsigma_1\right) \frac{(1 + 2Mc_1^{-1})^{-4}}{M} \|d_k^{WYL}\|^2 \\
&\quad + \gamma_k^2 o(\|d_k^{WYL}\|^2) \\
&= \left( \frac{(d_k^{WYL})^T (\bar{A}_k - A_k) d_k^{WYL}}{(d_k^{WYL})^T A_k d_k^{WYL}} \right) \|g_k\|^2 + \varsigma_2 \|g_k\|^2 \\
&= \varsigma_2 \|g_k\|^2 + o(\|g_k\|^2).
\end{aligned} \tag{59}$$

So, for sufficiently large  $k$ , we have

$$g(x_k + \gamma_k d_k^{WYL})^T d_k^{WYL} \geq \varsigma_2 g_k^T d_k^{WYL}. \tag{60}$$

This implies that  $\alpha_k = \gamma_k$  satisfy the Wolfe line search. The proof is complete.  $\square$

If we use the restart MWYL algorithm, the  $n$ -step quadratic convergence is desirable. In the next, we use the  $|\gamma_k|$  as the initial step-length and give the algorithm steps of the restart MWYL algorithm.

*Algorithm 7* (called RWYL).

Step 0: Given  $x_0 \in \mathfrak{R}^n$ ,  $r > 0$ ,  $\varsigma_1 \in (0, 1/2)$ ,  $\rho \in [0, 1)$ ,  $\epsilon \in [0, 1)$ , let  $k := 0$ .

Step 1: If  $\|g_k\| \leq \epsilon$ , stop.

Step 2: If the inequality  $f(x_k + |\gamma_k| d_k^{WYL}) \leq f(x_k) + \varsigma_1 |\gamma_k| g_k^T d_k^{WYL}$  holds, we set  $\alpha_k = |\gamma_k|$ . Otherwise,

we determine  $\alpha_k = \max\{|\gamma_k| \rho^j \mid j = 0, 1, 2, \dots\}$  satisfying

$$f(x_k + \alpha_k d_k^{WYL}) \leq f(x_k) + \varsigma_1 \alpha_k g_k^T d_k^{WYL}. \quad (61)$$

Step 3: Let  $x_{k+1} = x_k + \alpha_k d_k^{WYL}$ , and  $k := k + 1$ .

Step 4: If  $\|g_k\| \leq \epsilon$ , stop.

Step 5: If  $k = r$ , we let  $x_0 := x_k$ . Go to step 1.

Step 6: Compute  $d_k^{WYL}$  by (6). Go to step 2.

**Lemma 8.** *Let Assumption ii hold and  $\{x_k\}$  be generated by the RWYL algorithm. Then there exist positive numbers  $c_i^*$ ,  $i = 1, 2, 3, 4$ , such that*

$$\begin{aligned} \|g_{k+1}\| &\leq c_1^* \|d_k^{WYL}\|, \\ |\beta_{k+1}^{WYL}| &\leq c_2^*, \\ |\theta_{k+1}^{WYL}| &\leq c_3^*, \\ \|d_{k+1}^{WYL}\| &\leq c_4^* \|d_k^{WYL}\|. \end{aligned} \quad (62)$$

*Proof.* Considering the first inequality of (62), we get

$$\begin{aligned} \|g_{k+1}\| &= \|g_k + (g_{k+1} - g_k)\| \\ &\leq \|g_k\| + |\gamma_k| \cdot \|\widehat{A}_k d_k^{WYL}\| \\ &\leq \|d_k^{WYL}\| + \frac{M}{m} \cdot \|d_k^{WYL}\| = \left(1 + \frac{M}{m}\right) \|d_k^{WYL}\| \\ &\equiv c_1^* \|d_k^{WYL}\|, \end{aligned} \quad (63)$$

where  $\widehat{A}_k = \int_0^1 \nabla^2 f(x_k + \tau |\gamma_k| d_k^{WYL}) d\tau$ . By the definition of  $\beta_{k+1}^{WYL}$  we discuss the other three inequalities of (62), respectively. Starting from  $\beta_{k+1}^{WYL}$ , by the (39) and (62), we get

$$\begin{aligned} |\beta_{k+1}^{WYL}| &= \frac{\|g_{k+1}^T y_k^{WYL}\|}{\|g_k\|^2} \leq \frac{\|g_{k+1}^T\| \cdot \|y_k^{WYL}\|}{\|g_k\|^2} \\ &\leq \frac{2c_1^{-1} M \|g_{k+1}\|}{\|d_k^{WYL}\|} \leq 2c_1^{-1} M c_1^* \equiv c_2^*. \end{aligned} \quad (64)$$

By (40), (62), and the definition of  $\theta_{k+1}^{WYL}$ , we obtain

$$\begin{aligned} |\theta_{k+1}^{WYL}| &= \left| \frac{g_{k+1}^T d_k^{WYL}}{\|g_k\|^2} \right| \leq \|g_{k+1}\| \cdot \frac{\|d_k^{WYL}\|}{\|g_k\|^2} \\ &\leq \frac{c_1^* \|d_k^{WYL}\|^2}{\|g_k\|^2} \leq c_1^* (1 + 4M c_1^{-1})^2 \equiv c_3^*. \end{aligned} \quad (65)$$

By the above conclusion and the definition of  $d_{k+1}^{WYL}$ , we have

$$\begin{aligned} \|d_{k+1}^{WYL}\| &= \left\| -g_{k+1} + \beta_{k+1}^{WYL} d_k^{WYL} - \theta_{k+1}^{WYL} y_k^{WYL} \right\| \\ &\leq \|g_{k+1}\| + |\beta_{k+1}^{WYL}| \cdot \|d_k^{WYL}\| + |\theta_{k+1}^{WYL}| \\ &\quad \cdot \|y_k^{WYL}\| \\ &\leq c_1^* \|d_k^{WYL}\| + c_2^* \|d_k^{WYL}\| + 2c_3^* M \alpha_k \|d_k^{WYL}\| \\ &\leq (c_1^* + c_2^* + 2M c_3^*) \|d_k^{WYL}\| \equiv c_4^* \|d_k^{WYL}\|. \end{aligned} \quad (66)$$

The proof is complete.  $\square$

In the following, we will prove the n-order quadratic convergence of the RWYL algorithm. We always let Assumption ii hold and  $\{x_k\}$  be generated by the RWYL algorithm. Using  $x^*$  as the unique solution of problem (1), by Theorem 2, we have  $\{x_k\} \rightarrow x^*$ . The equation  $\alpha_k = \gamma_k$  always holds if only  $k$  is large enough by the Theorem 6. In order to establish this convergence of the RWYL algorithm, we further need the following assumption.

*Assumption iii.* In some neighborhood  $N$  of  $x^*$ ,  $\nabla^2 f$  is Lipschitz continuous.

Based on Assumption iii and the above lemma, we have the following remarks. Let  $\widehat{f}_{kr}(x)$  be the second-order approximate function of  $f$  in the neighborhood of the initial point  $x_{kr}$ , then we have

$$\begin{aligned} \widehat{f}_{kr}(x) &= f(x_{kr}) + g(x_{kr})^T (x - x_{kr}) \\ &\quad + \frac{1}{2} (x - x_{kr})^T \nabla^2 f(x_{kr}) (x - x_{kr}). \end{aligned} \quad (67)$$

Let  $\{x_{kr}^i\}$  and  $\{d_{kr}^{WYL(i)}\}$  be the iterations and directions generated by the RWYL algorithm to minimize the quadratic function  $\widehat{f}_{kr}(x)$  with initial point  $x_{kr}^0 = x_{kr}$ . Specifically, the sequence  $\{x_{kr}^i\}$  is generated by using the following process:

$$\begin{aligned} x_{kr}^0 &= x_{kr}, \\ x_{kr}^{i+1} &= x_{kr}^0 + \alpha_{kr}^i d_{kr}^{WYL(i)}, \end{aligned} \quad (68)$$

$i = 0, 1, \dots,$

and

$$\begin{aligned} d_{kr}^{WYL(i)} &= \begin{cases} -g_{kr}^i + \beta_{kr}^{WYL(i)} d_{kr}^{WYL(i-1)} - \theta_{kr}^i y_{kr}^{WYL(i-1)}, & \text{if } k \geq 1 \\ -g_{kr}^0, & \text{if } k = 0, \end{cases} \end{aligned} \quad (69)$$

where  $g_{kr}^i = g(x_{kr}^i)$  for  $i = 1, 2, \dots$ :

$$\begin{aligned} \beta_{kr}^{WYL(i)} &= \frac{(g_{kr}^i)^T y_{kr}^{WYL(i-1)}}{\|g_{kr}^{i-1}\|^2}, \\ y_{kr}^{WYL(i-1)} &= g_{kr}^i - g_{kr}^{i-1}, \end{aligned}$$

$$\theta_{kr}^{WYL(i)} = \frac{(g_{kr}^i)^T d_{kr}^{WYL(i-1)}}{\|g_{kr}^{i-1}\|^2} \quad (70)$$

From the proof process of Theorem 6, it is not difficult to see that when  $k$  is sufficiently large, step length  $\gamma_k$  can always be found. Because  $\widehat{f_{kr}}(x)$  is a quadratic function,  $\gamma_k$  is the same as the step length obtained by the exact line search. Consequently, we have  $\theta_{kr}^{WYL(i)} = 0$ ; moreover there is an index  $j(kr) \leq n$  such that  $x_{kr}^{j(kr)}$  is the exact minimizer of  $\widehat{f_{kr}}$ .

Similar to Lemmas (A.8)-(A-10) in the paper [30], it is not difficult to get the following relations:

$$\begin{aligned} & \|\beta_{kr+i}^{WYL} d_{kr+i-1}^{WYL} - \beta_{kr}^{WYL(i)} d_{kr}^{WYL(i-1)}\| \\ &= O(\|d_{kr+i-1}^{WYL} - d_{kr}^{WYL(i-1)}\|) + O(\|d_{kr}^{WYL}\|^2), \end{aligned} \quad (71)$$

$$\begin{aligned} & \|g_{kr+i} - g_{kr}^i\| \\ & \leq \|g_{kr+i-1} - g_{kr}^{i-1}\| + O(\|d_{kr}^{WYL}\|^2) \\ & \quad + M \|\alpha_{kr+i-1} d_{kr+i-1}^{WYL} - \alpha_{kr}^{WYL(i-1)} d_{kr}^{WYL(i-1)}\|, \end{aligned} \quad (72)$$

and

$$\begin{aligned} & \|\alpha_{kr+i} d_{kr+i}^{WYL} - \alpha_{kr}^i d_{kr}^{WYL(i)}\| \\ &= O(\|g_{kr+i} - g_{kr}^i\|) + O(\|d_{kr+i}^{WYL} - d_{kr}^{WYL(i)}\|) \\ & \quad + O(\|d_{kr}^{WYL}\|^2). \end{aligned} \quad (73)$$

The following lemma shows that the parameter  $\theta_k^{WYL}$  will converge to 0.

**Lemma 9.** For the parameter  $\theta_k^{WYL}$ , one has

$$|\theta_{kr+i+1}^{WYL}| = O(\|g_{kr+i}\|) = O(\|g_{kr}\|). \quad (74)$$

*Proof.* Let  $\widehat{A}_{kr+i}$  and  $A_{kr+i}$  be defined by Lemma 8; we get

$$\begin{aligned} & \left| \frac{(d_{kr+i}^{WYL})^T \widehat{A}_{kr+i} d_{kr+i}^{WYL}}{(d_{kr+i}^{WYL})^T A_{kr+i} d_{kr+i}^{WYL}} - 1 \right| \\ & \leq \frac{|(d_{kr+i}^{WYL})^T \widehat{A}_{kr+i} d_{kr+i}^{WYL} - (d_{kr+i}^{WYL})^T A_{kr+i} d_{kr+i}^{WYL}|}{m \|d_{kr+i}^{WYL}\|^2} \\ & \leq \frac{1}{m \|d_{kr+i}^{WYL}\|^2} \left\{ \left\| \int_0^1 \nabla^2 f(x_{kr+i} + \tau \alpha_{kr+i} d_{kr+i}^{WYL}) d\tau \right. \right. \\ & \quad \left. \left. - \int_0^1 \nabla^2 f(x_{kr+i} + \epsilon_{kr+i} \tau \alpha_{kr+i} d_{kr+i}^{WYL}) d\tau \right\| \right. \\ & \quad \left. \cdot \|d_{kr+i}^{WYL}\|^2 \right\} \leq \frac{L}{m} \int_0^1 (1 - \epsilon_{kr+i}) \tau d\tau \cdot \|\alpha_{kr+i}\| \end{aligned}$$

$$\begin{aligned} & \cdot d_{kr+i}^{WYL}\| \leq \frac{L}{2m} (1 - \epsilon_{kr+i}) \cdot \|d_{kr+i}^{WYL}\| \\ & \leq \frac{L(1 - \epsilon_{kr+i})(1 + 4Mc_1^{-1})}{2m} \|g_{kr+i}\| = O(\|g_{kr+i}\|) \end{aligned} \quad (75)$$

where  $L$  is Lipschitz constant for  $\nabla^2 f$  on set  $N$ . Then we get

$$\left| \frac{(d_{kr+i}^{WYL})^T \widehat{A}_{kr+i} d_{kr+i}^{WYL}}{(d_{kr+i}^{WYL})^T A_{kr+i} d_{kr+i}^{WYL}} - 1 \right| = O(\|g_{kr+i}\|). \quad (76)$$

For  $k$  sufficiently large, we get  $\alpha_k = \gamma_k$ . By the mean value theorem, we have

$$\begin{aligned} & g(x_{kr+i})^T d_{kr+i}^{WYL} = g(x_{kr+i} + \gamma_{kr+i} d_{kr+i}^{WYL})^T d_{kr+i}^{WYL} \\ &= g(x_{kr+i})^T d_{kr+i}^{WYL} \\ & \quad + (g(x_{kr+i} + \gamma_{kr+i} d_{kr+i}^{WYL}) - g(x_{kr+i}))^T d_{kr+i}^{WYL} \\ & \leq g_{kr+i}^T d_{kr+i}^{WYL} + \gamma_{kr+i} (d_{kr+i}^{WYL})^T \widehat{A}_{kr+i} d_{kr+i}^{WYL} \\ &= \left( \frac{(d_{kr+i}^{WYL})^T \widehat{A}_{kr+i} d_{kr+i}^{WYL}}{(d_{kr+i}^{WYL})^T A_{kr+i} d_{kr+i}^{WYL}} - 1 \right) \|g_{kr+i}\|^2 \\ &= O(\|g_{kr+i}\|^3). \end{aligned} \quad (77)$$

Therefore, by the definition of  $\theta_k^{WYL}$ , we have

$$\begin{aligned} |\theta_{kr+i+1}^{WYL}| &= \frac{|g(x_{kr+i+1})^T d_{kr+i}^{WYL}|}{\|g_{kr+i}\|^2} = O(\|g_{kr+i}\|) \\ &= O(\|g_{kr}\|). \end{aligned} \quad (78)$$

This completes the proof.  $\square$

**Theorem 10.** Let Assumptions ii and iii hold; then, for all  $i = 0, 1, \dots$ , one gets

$$\|\alpha_{kr+i} d_{kr+i}^{WYL} - \alpha_{kr}^i d_{kr}^{WYL(i)}\| = O(\|d_{kr}^{WYL}\|^2) \quad (79)$$

and

$$\|\alpha_{kr+i} d_{kr+i}^{WYL} - \alpha_{kr}^i d_{kr}^{WYL(i)}\| = O(\|x_{kr} - x^*\|^2). \quad (80)$$

*Proof.* First, we will prove the following relationship. For all  $i = 0, 1, \dots, j(kr)$ , we have

$$\|g_{kr+i} - g_{kr}^i\| = O(\|d_{kr}^{WYL}\|^2), \quad (81)$$

$$\|d_{kr+i}^{WYL} - d_{kr}^{WYL(i)}\| = O(\|d_{kr}^{WYL}\|^2), \quad (82)$$

and

$$\|\alpha_{kr+i} d_{kr+i}^{WYL} - \alpha_{kr}^i d_{kr}^{WYL(i)}\| = O(\|d_{kr}^{WYL}\|^2). \quad (83)$$

For the RWYL algorithm, by  $x_{kr}^0 = x_{kr}$  and  $\alpha_{kr} = \gamma_{kr}$ . The equalities (81)-(83) obviously hold for  $i = 0$ . Suppose that (81)-(83) hold for  $i \geq 0$ , we prove that the equalities (81)-(83) hold for  $i + 1$ . By inequality (72), we get

$$\begin{aligned} \|g_{kr+i+1} - g_{kr}^{i+1}\| &\leq \|g_{kr+i} - g_{kr}^i\| + O\left(\|d_{kr}^{WYL}\|^2\right) \\ &\quad + M \left\| \alpha_{kr+i} d_{kr}^{WYL} - \alpha_{kr}^i d_{kr}^{WYL(i)} \right\| \\ &= O\left(\|d_{kr}^{WYL}\|^2\right) + O\left(\|d_{kr}^{WYL}\|^2\right) \\ &\quad + MO\left(\|d_{kr}^{WYL}\|^2\right) = O\left(\|d_{kr}^{WYL}\|^2\right) \end{aligned} \quad (84)$$

Using the equality  $d_{kr} = -g_{kr}$  and the mean value theorem, we get

$$\begin{aligned} \|d_{kr+i+1}^{WYL} - d_{kr}^{WYL(i+1)}\| &= \left\| -g_{kr+i+1} + \beta_{kr+i+1}^{WYL} d_{kr+i}^{WYL} \right. \\ &\quad \left. - \theta_{kr+i+1}^{WYL} y_{kr+i}^{WYL} + g_{kr}^{i+1} - \beta_{kr}^{WYL(i+1)} d_{kr}^{WYL(i)} \right\| \\ &\leq \|g_{kr+i+1} - g_{kr}^{i+1}\| + |\theta_{kr+i+1}^{WYL}| \cdot \|y_{kr+i}^{WYL}\| \\ &\quad + \left\| \beta_{kr+i+1}^{WYL} d_{kr+i}^{WYL} - \beta_{kr}^{WYL(i+1)} d_{kr}^{WYL(i)} \right\| \\ &\leq O\left(\|d_{kr}^{WYL}\|^2\right) + O\left(\|d_{kr}^{WYL}\|\right) \cdot O\left(\|g_{kr}\|\right) \\ &\quad + O\left(\|d_{kr}^{WYL}\|^2\right) = O\left(\|d_{kr}^{WYL}\|^2\right), \end{aligned} \quad (85)$$

and

$$\begin{aligned} \left\| \alpha_{kr+i+1} d_{kr+i+1}^{WYL} - \alpha_{kr}^{i+1} d_{kr}^{WYL(i+1)} \right\| \\ &= O\left(\|g_{kr+i+1} - g_{kr}^{i+1}\|\right) + O\left(\|d_{kr+i+1}^{WYL} - d_{kr}^{WYL(i+1)}\|\right) \\ &\quad + O\left(\|d_{kr}^{WYL}\|^2\right) \\ &\leq O\left(\|d_{kr}^{WYL}\|^2\right) + O\left(\|d_{kr}^{WYL}\|^2\right) + O\left(\|d_{kr}^{WYL}\|^2\right) \\ &= O\left(\|d_{kr}^{WYL}\|^2\right), \end{aligned} \quad (86)$$

where the above inequalities follow (71), (72), (73), (81), and (82). Thus, equalities (81)-(83) hold for all  $i = 1, 2, \dots, j(kr)$ . Moreover equality (79) holds too. Now we prove that (80) holds. Considering

$$\begin{aligned} \|d_{kr}^{WYL}\| &= \|g_{kr}\| = \|g(x_{kr}) - g(x^*)\| \\ &\leq M \cdot \|x_{kr} - x^*\|, \end{aligned} \quad (87)$$

we get

$$\|d_{kr}^{WYL}\|^2 \leq M^2 \|x_{kr} - x^*\|^2. \quad (88)$$

Therefore equality (80) holds. This completes the proof.  $\square$

Based on the above lemmas, similar to Theorem 4.2 of [32], we can get the n-step quadratic convergence of the RWYL algorithm. Here we only state it as follows but omit the proof too.

**Theorem 11.** *Let Assumptions ii and iii hold; then there exists a constant  $c' > 0$  satisfying*

$$\limsup_{k \rightarrow \infty} \frac{\|x_{kr+n} - x^*\|}{\|x_{kr} - x^*\|^2} \leq c' < \infty. \quad (89)$$

Namely, the RWYL algorithm is quadratically convergent.

## 5. Numerical Results

This section reports some numerical experiments with Algorithm 7 (RWYL). In order to show the effectiveness of the given algorithm, we will test the algorithm in [15, 19] (Hager-Zhang), the algorithm in [6, 7] (PRP), and the algorithm in [21] (WYL). In these four algorithms, the Wolfe-Powell line search technique is used as well as the parameters  $\varsigma_1 = 0.1$  and  $\varsigma_2 = 0.9$ . The restart constant  $r = 10$ . The program will be stopped if  $\|g(x_k)\|_\infty \leq \max\{10^{-6}, 10^{-12}\|g(x_0)\|_\infty\}$  or  $\|g(x_k)\|_\infty \leq 10^{-6}(1 + |f(x_k)|)$  holds.

**5.1. Normal Unstrained Optimization Problems.** The unconstrained optimization problems with the given initial points can be found at

<http://camo.ici.ro/neculai/THRECECG/funname.txt>,

which were collected by Neculai Andrei. The programs are written by Fortran and the codes are downloaded from

<http://users.clas.ufl.edu/hager/papers/Software/>,

which are written by Hager and Zhang.

All codes run on PC Core 2 Duo CPU at 3.2 GHz, 2.00GB of RAM, and Windows 7 operation system. The dimension of the test problems is 10000, 50000, and 10,000 variables. The dimension (dim) of the variable, the CPU time in seconds (CPU) and the number of iterations (NI), function evaluations and gradient evaluations (NFG), the final function values, and the norm value of the gradient when the program is stopped for these four algorithms are computed. The profiles of Dolan and Moré [37] are used to analyze the performance data of these four algorithms. The fraction  $P$  of problems is plotted where any given algorithm is within a factor  $t$  of the best time. In a performance profile plot, the top curve shows that the algorithm solves the most problems in a time, which was within a factor  $t$  of the best time.

Figures 1–3 show the performance of the RWYL, Hager-Zhang, PRP, and WYL algorithms with the dimension 10000 about NI, NFG, and CPU time, respectively. It is not difficult to see that these four algorithms can successfully solve the given problems. The RWYL is the best profile among these four algorithms and the normal WYL algorithm has the worst performance.

In Figures 4–6, we use CPU time, NI, and NFG to compare the performance of the conjugate gradient codes RWYL, Hager-Zhang, PRP, and WYL algorithms on the dimension 50000. These four figures indicate that, relative to the CPU time, NI, and NFG, RWYL is fastest, then PRP,

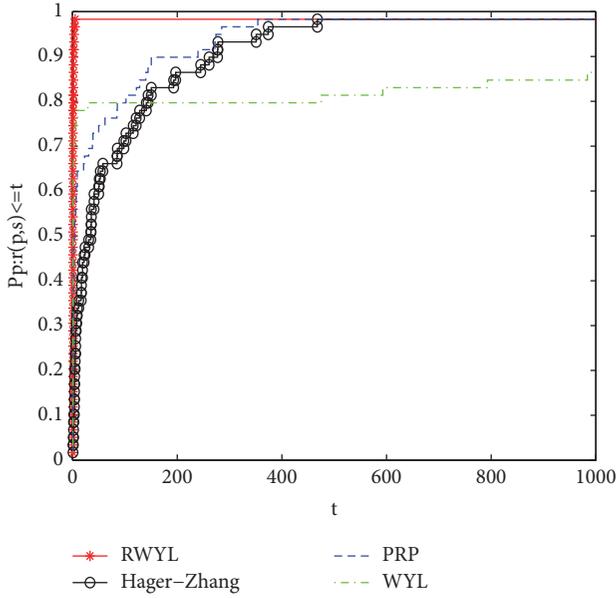


FIGURE 1: Performance profiles of these algorithms on NI (dim=10000).

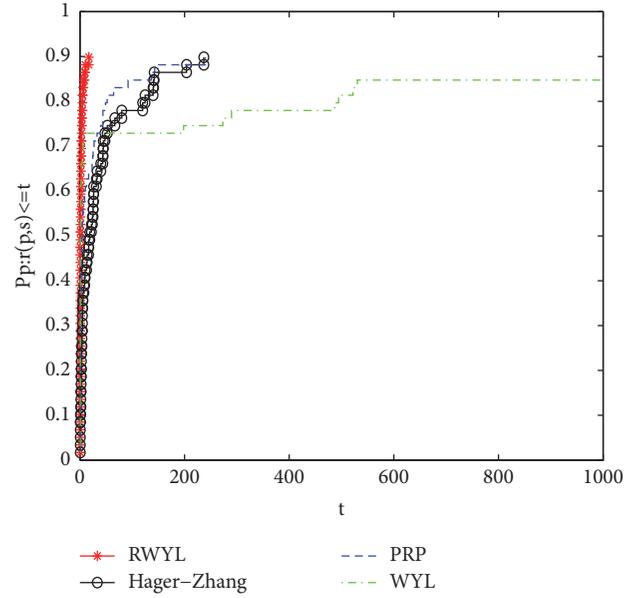


FIGURE 3: Performance profiles of these algorithms on CPU (dim=10000).

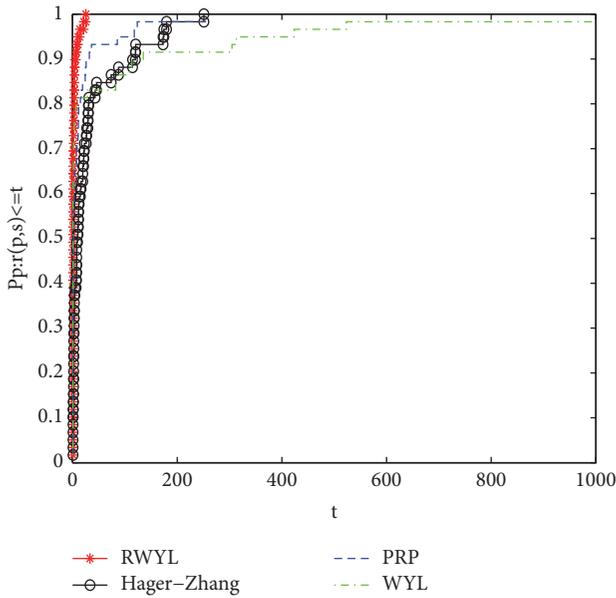


FIGURE 2: Performance profiles of these algorithms on NFG (dim=10000).

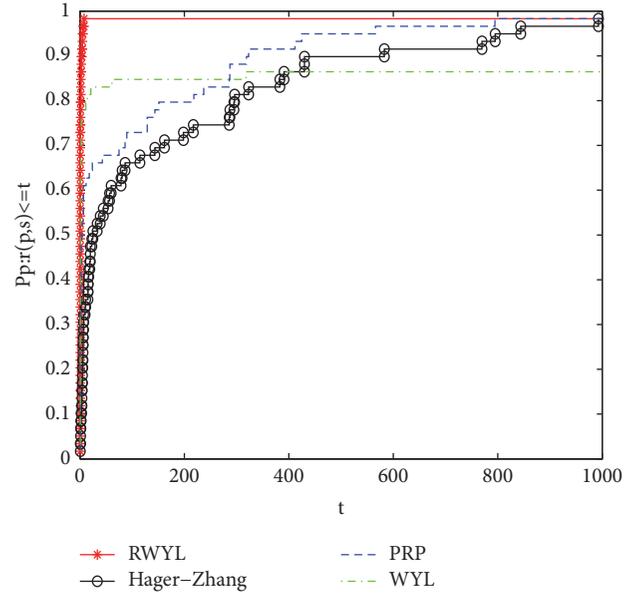


FIGURE 4: Performance profiles of these algorithms on NI (dim=50000).

then Hager-Zhang algorithm, and then WYL. These codes only differ in their choice of the search direction; then we can conclude that the RWYL generates the best search directions for these test problems, on average.

In Figures 7–9, we use CPU time, NI, and NFG to compare the performance of the conjugate gradient codes RWYL, Hager-Zhang algorithm, PRP, and WYL on the dimension 100000. These four figures indicate that, relative to the CPU time, NI, and NFG, RWYL is fastest, then Hager-Zhang algorithm, then PRP, and then WYL.

According to these nine figures, it is easy to see that the RWYL has the best performance for the dimensions 10000, 50000, and 100000. The Hager-Zhang algorithm becomes competitive with the dimension become large, which shows that the Hager-Zhang algorithm is very effective for large-scale problems. The PRP algorithm has the stable numerical performance for any dimensions. The normal WYL can also successfully solve the optimization problems and its efficiency is limited. To directly show the CPU time, NI, and NFG of these four algorithms, each algorithm number is listed in Table 1.

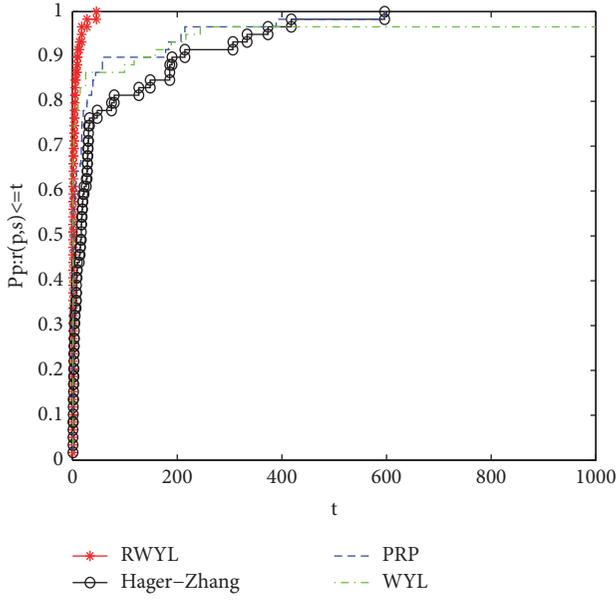


FIGURE 5: Performance profiles of these algorithms on NFG (dim=50000).

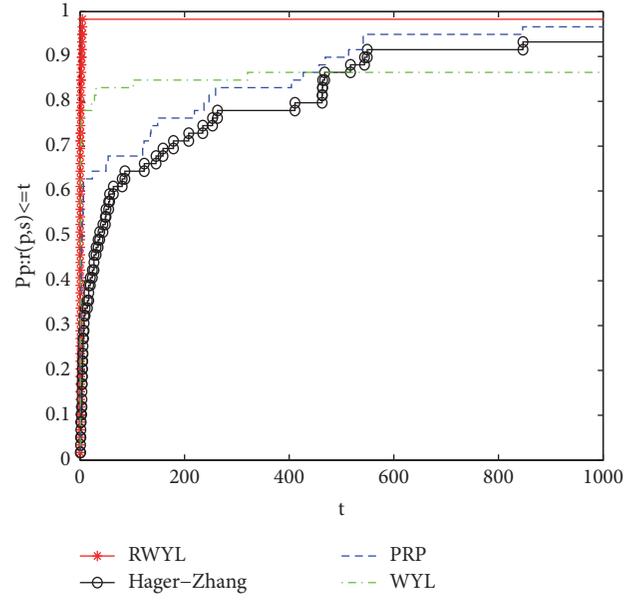


FIGURE 7: Performance profiles of these algorithms on NI (dim=100000).

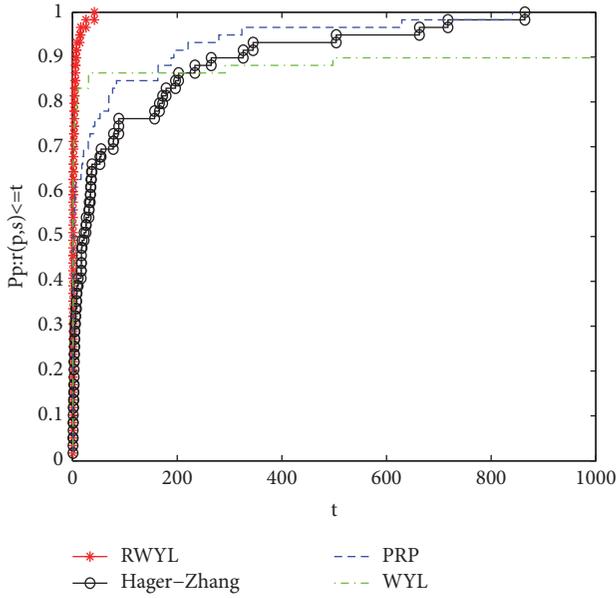


FIGURE 6: Performance profiles of these algorithms on CPU (dim=50000).

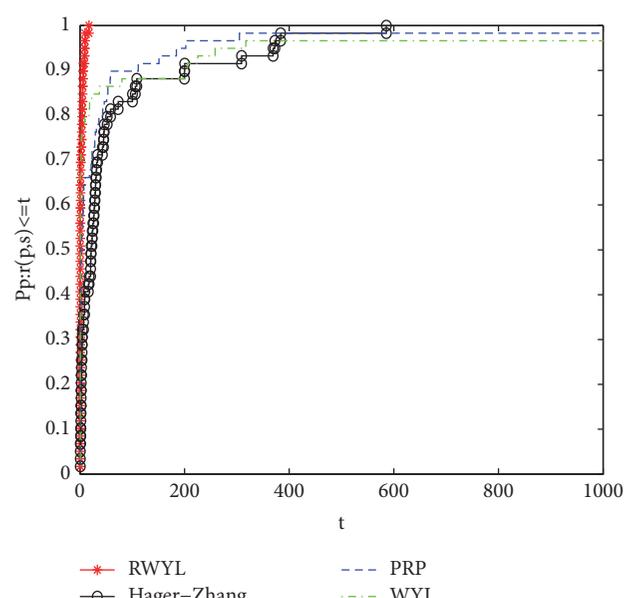


FIGURE 8: Performance profiles of these algorithms on NFG (dim=100000).

5.2. Benchmark Problems (Engineer Problems). The following Benchmark Problems can be found at

<http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume24/ortizboyer05a-html/node6.html>.

(1) Sphere function:

$$f_{sph}(x) = \sum_{i=1}^n x_i^2, \quad x_i \in [-5.12, 5.12] \quad (90)$$

$$x^* = (0, 0, \dots, 0), \quad f_{sph}(x^*) = 0.$$

(2) Schwefel function:

$$f_{sch}(x) = 418.9828n + \sum_{i=1}^n x_i \sin \sqrt{|x_i|},$$

$$x_i \in [-512.03, 511.97] \quad (91)$$

$$x^* = (-420.9678, -420.9678, \dots, -420.9678),$$

$$f_{sch}(x^*) = 0.$$

TABLE 1: Results of total.

Algorithm	dim=10000			dim=50000			dim=100000		
	NI	NFG	CPU	NI	NFG	CPU	NI	NFG	CPU
RWYL	144	4035	8.78	222	4546	76.88	151	3653	141.79
Hager-Zhang	8655	33062	116.14	16329	58406	950.75	20661	72516	2251.47
PRP	6769	25186	144.00	12563	43475	699.07	116090	354062	10985.57
WYL	59433	181937	415.10	89221	271059	3117.06	167044	503781	11868.37

TABLE 2: Results of Benchmark Problems for RWYL.

No.	1	2	3	4	5
$x_0$	$(-0.001, \dots)$				
	NI/NFG/CPU	NI/NFG/CPU	NI/NFG/CPU	NI/NFG/CPU	NI/NFG/CPU
n=300	2/6/4.680e-2	1/3/1.560e-2	150/450/1.926e+1	31/93/1.092e+0	3/9/9.360e-2
n=1000	2/6/1.560e-2	1/3/0.000e+0	452/1356/2.033e+3	51/153/1.070e+1	801/6394/4.570e+0
$x_0$	$(0.001, \dots)$				
	NI/NFG/CPU	NI/NFG/CPU	NI/NFG/CPU	NI/NFG/CPU	NI/NFG/CPU
n=300	2/6/1.560e-2	1/3/1.560e-02	150/450/1.931e+1	31/93/7.020e-1	3/9/6.240e-2
n=1000	2/6/1.560e-2	1/3/0.000e+0	452/1356/2.033e+3	51/153/1.065e+1	801/6394/4.243e+0

(3) Schwefel's function:

$$f_{schds}(x) = \sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)^2, \quad x_i \in [-65.536, 65.536] \quad (92)$$

$$x^* = (0, 0, \dots, 0), \quad f_{schds}(x^*) = 0.$$

(4) Griewank function:

$$f_{gri}(x) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos \frac{x_i}{i}, \quad (93)$$

$$x_i \in [-600, 600]$$

$$x^* = (0, 0, \dots, 0), \quad f_{gri}(x^*) = 0.$$

(5) Rastrigin function:

$$f_{ras}(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)), \quad (94)$$

$$x_i \in [-5.12, 5.12]$$

$$x^* = (0, 0, \dots, 0), \quad f_{ras}(x^*) = 0.$$

Benchmark Problems are from the engineer fields and there are many scholars focusing on the studies of these problems. The given algorithm of this paper can also successfully solve them. We do the experiments about the RWYL and the normal WYL for comparing and omit the other two methods Hager-Zhang and PRP. The parameters and the stopping rule are the same as the above subsection. The codes are written by Matlab 2017 and run on PC Core 2 Duo CPU at 2.26

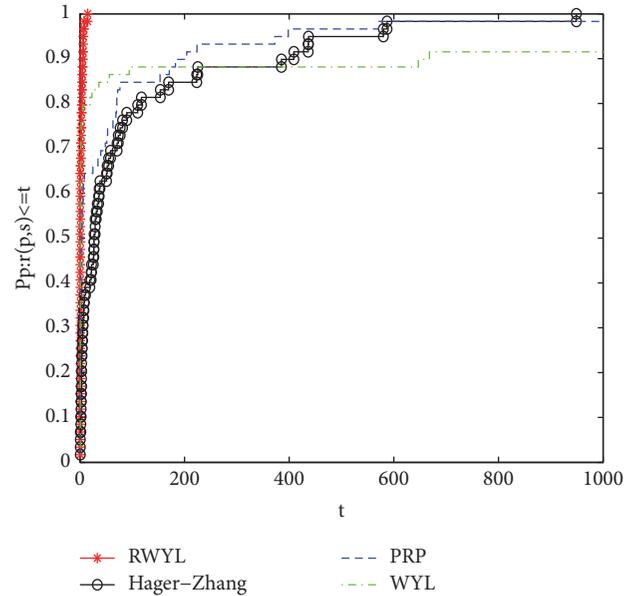


FIGURE 9: Performance profiles of these algorithms on CPU (dim=100000).

GHz, 6.00GB of RAM, and Windows 7 operation system. The dimension is 300 and 1000 variables. The detailed numerical results are listed in Tables 2 and 3.

To directly see the results of Tables 2 and 3, we compute the total NI and NFG and set them in Table 4.

The results of Table 4 show that the restart algorithm is more competitive with the normal algorithm for the Benchmark Problems.

TABLE 3: Results of Benchmark Problems for WYL.

No.	1	2	3	4	5
$x_0$	(-0.001, ...)	(-0.001, ...)	(-0.001, ...)	(-0.001, ...)	(-0.001, ...)
	NI/NFG/CPU	NI/NFG/CPU	NI/NFG/CPU	NI/NFG/CPU	NI/NFG/CPU
n=300	519/2075/1.684e+0	1/3/1.560e-2	801/4002/5.396e+1	141/389/1.716e+0	801/4003/1.981e+0
n=1000	549/2195/1.809e+0	1/3/0.000e+0	801/4803/1.824e+3	152/419/1.622e+1	801/4003/2.199e+0
$x_0$	(0.001, ...)	(0.001, ...)	(0.001, ...)	(0.001, ...)	(0.001, ...)
	NI/NFG/CPU	NI/NFG/CPU	NI/NFG/CPU	NI/NFG/CPU	NI/NFG/CPU
n=300	519/2075/1.606e+0	1/3/0.000e+0	801/4002/5.343e+1	141/389/1.638e+0	801/4003/1.996e+0
n=1000	549/2195/1.450e+0	1/3/0.000e+0	801/4803/1.822e+3	152/419/1.597e+1	801/4003/2.230e+0

TABLE 4: Results of Tables 2 and 3.

	RWYL	WYL
NI	2988	9134
NFG	16946	43790

5.3. Parameters Estimation of Nonlinear Muskingum Models. The basic Muskingum model, the continuity, and storage equations are defined by

$$\frac{dS_t}{dt} = I_t - Q_t, \quad S_t = k [xI_t + (1 - x)Q_t], \quad (95)$$

where, at time  $t$ ,  $S_t$  is channel storage,  $I_t$  is rate of inflow and  $Q_t$  = denotes outflow,  $k$  is storage-time constant, and

min SSQ1

$$= \sum_{i=1}^{n-1} \left( \left( 1 - \frac{\Delta t}{2} \right) k [xI_{i+1} + (1 - x)Q_{i+1}]^m - \left( 1 - \frac{\Delta t\theta}{2} \right) k [xI_i + (1 - x)Q_i]^m - \frac{\Delta t}{2} (I_i - Q_i) - \frac{\Delta t}{2} (1 - \Delta t\theta) (I_{i+1} - Q_{i+1}) \right)^2, \quad (97)$$

min SSQ2

$$= \sum_{i=1}^{n-1} \left( \left( 1 - \frac{\Delta t}{2} \right) k [xI_{i+1}^m + (1 - x)Q_{i+1}^m] - \left( 1 - \frac{\Delta t\theta}{2} \right) k [xI_i^m + (1 - x)Q_i^m] - \frac{\Delta t}{2} (I_i - Q_i) - \frac{\Delta t}{2} (1 - \Delta t\theta) (I_{i+1} - Q_{i+1}) \right)^2. \quad (98)$$

This subsection will use our RWYL to estimate the parameters of the above two Muskingum models, named Model 1 (97) and Model 2 (98).

All in all, we can conclude that that the restart algorithm is competitive with the norm algorithm without restart technique and other similar algorithms.

## 6. Conclusions

Nonlinear conjugate gradient algorithm is one of the most effective algorithms in optimization algorithms, especially for large-scale optimization problems. Many scholars have obtained many interesting results in this field. This paper focuses on a modified WYL CG algorithm with restart technique for large-scale optimization. In our opinion, there are at

$x$  is weighting factor for the river reach. The generalized trapezoidal formula [38] is

$$\min f(k, x, m)$$

$$= \sum_{i=1}^{n-1} \min \left( 1 - \frac{\Delta t}{2} \theta \right) k [xI_{i+1} + (1 - x)Q_{i+1}]^m - \left( 1 - \frac{\Delta t}{2} \theta \right) k [xI_i + (1 - x)Q_i]^m - \frac{\Delta t}{2} [I_i - Q_i] + \frac{\Delta t}{2} (1 - \Delta t\theta) [I_{i+1} - Q_{i+1}]. \quad (96)$$

To conveniently estimate the parameters  $k$ ,  $x$ , and  $m$  in the nonlinear Muskingum model, the objective function can be rewritten as

least seven issues that warrant further research and improvement: (i) The first issue that should be considered is the choice of the restart parameter  $r$  in the RWYL algorithm; the value ( $r = 10$ ) used here is not the only choice. (ii) The second important issue is the termination condition; better termination conditions may exist for the CG algorithms, which may improve the numerical performance and convergence. (iii) Under the restart strategy, other similar CG algorithms with quadratic convergence are worth studying. (iv) It would be interesting to test the performance of the given algorithm when applied to other optimization problems that arise in the image processing field. (v) We all know that the nonmonotone line search techniques are very effective. In the future, we will study the possibility of combining CG algorithms with nonmonotonic techniques for large-scale optimization

problems and will attempt to obtain good results. (vi) In the experiment, there are 59 optimization problems with the dimension 10000, 50000, and 100000 variables that are tested. We also do the test for the Benchmark Problems which has wild applications in engineer fields. In the future, more problems and numerical experiments should be done to turn out the performance of the CG algorithms. (vii) The last issue is the use of the CG algorithm for nonsmooth optimization and nonlinear equations, which we consider to be important for future research. All these topics will be the focus of our future work.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant no. 11661009), the Guangxi Science Fund for Distinguished Young Scholars (no. 2015GXNS-FGA139001), the Guangxi Natural Science Key Fund (no. 2017GXNSFDA198046), and the Guangxi Science Fund of Young and Middle-Aged Teachers for the Basic Ability Promotion (no. 2017KY0019).

## References

- [1] Y. H. Dai and Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," *SIAM Journal on Optimization*, vol. 10, no. 1, pp. 177–182, 1999.
- [2] R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, New York, NY, USA, 2nd edition, 1987.
- [3] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The Computer Journal*, vol. 7, pp. 149–154, 1964.
- [4] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436 (1953), 1952.
- [5] Y. Liu and C. Storey, "Efficient generalized conjugate gradient algorithms, Part 1," *Journal of Optimization Theory and Applications*, vol. 69, no. 1, pp. 129–137, 1991.
- [6] B. Polak, "The conjugate gradient method in extreme problems," *Computational Mathematics and Mathematical Physics*, vol. 9, pp. 94–112, 1969.
- [7] E. Polak and G. Ribière, "Note sur la convergence de méthodes de directions conjuguées," *Revue Française d'Informatique et de Recherche Opérationnelle*, vol. 3, no. 16, pp. 35–43, 1969.
- [8] G. Yuan and Z. Sheng, *Press of Science*, Press of Science, Beijing, China, 2017.
- [9] M. J. D. Powell, "Nonconvex minimization calculations and the conjugate gradient method," in *Numerical Analysis*, vol. 1066 of *Lecture Notes in Mathematics*, pp. 122–141, Springer, Berlin, Germany, 1984.
- [10] Y. Yuan, "Analysis on the conjugate gradient method," *Optimization Methods and Software*, vol. 2, no. 1, pp. 19–29, 2010.
- [11] Y. H. Dai and Y. Yuan, *Nonlinear conjugate gradient Methods*, vol. 14, Shanghai Scientific and Technical Publishers, 1998.
- [12] X. Dong, H. Liu, and Y. He, "A self-adjusting conjugate gradient method with sufficient descent condition and conjugacy condition," *Journal of Optimization Theory and Applications*, vol. 165, no. 1, pp. 225–241, 2015.
- [13] X. L. Dong, H. W. Liu, Y. B. He, and X. . Yang, "A modified Hestenes-Stiefel conjugate gradient method with sufficient descent condition and conjugacy condition," *Journal of Computational and Applied Mathematics*, vol. 281, pp. 239–249, 2015.
- [14] J. C. Gilbert and J. Nocedal, "Global convergence properties of conjugate gradient methods for optimization," *SIAM Journal on Optimization*, vol. 2, no. 1, pp. 21–42, 1992.
- [15] W. W. Hager and H. Zhang, "Algorithm 851: CG-DESCENT, a conjugate gradient method with guaranteed descent," *ACM Transactions on Mathematical Software*, vol. 32, no. 1, pp. 113–137, 2006.
- [16] M. J. Powell, "Convergence properties of algorithms for nonlinear optimization," *SIAM Review*, vol. 28, no. 4, pp. 487–500, 1986.
- [17] G. Yuan, "Modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems," *Optimization Letters*, vol. 3, no. 1, pp. 11–21, 2009.
- [18] Y.-h. Dai, *Analysis of conjugate gradient methods [Ph.D. Thesis]*, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, 1997.
- [19] W. W. Hager and H. Zhang, "A new conjugate gradient method with guaranteed descent and an efficient line search," *SIAM Journal on Optimization*, vol. 16, no. 1, pp. 170–192, 2005.
- [20] X. Li, S. Wang, Z. Jin, and H. Pham, "A conjugate gradient algorithm under Yuan-Wei-Lu line search technique for large-scale minimization optimization models," *Mathematical Problems in Engineering*, vol. 2018, Article ID 4729318, 11 pages, 2018.
- [21] Z. Wei, S. Yao, and L. Liu, "The convergence properties of some new conjugate gradient methods," *Applied Mathematics and Computation*, vol. 183, no. 2, pp. 1341–1350, 2006.
- [22] G. Yuan and X. Lu, "A modified PRP conjugate gradient method," *Annals of Operations Research*, vol. 166, pp. 73–90, 2009.
- [23] G. Yuan, X. Lu, and Z. Wei, "A conjugate gradient method with descent direction for unconstrained optimization," *Journal of Computational and Applied Mathematics*, vol. 233, no. 2, pp. 519–530, 2009.
- [24] G. Yuan, Z. Meng, and Y. Li, "A modified Hestenes and Stiefel conjugate gradient algorithm for large-scale nonsmooth minimizations and nonlinear equations," *Journal of Optimization Theory and Applications*, vol. 168, no. 1, pp. 129–152, 2016.
- [25] G. Yuan, Z. Wei, and G. Li, "A modified Polak-Ribière-Polyak conjugate gradient algorithm for nonsmooth convex programs," *Journal of Computational and Applied Mathematics*, vol. 255, pp. 86–96, 2014.
- [26] G. Yuan and M. Zhang, "A three-terms Polak-Ribière-Polyak conjugate gradient algorithm for large-scale nonlinear equations," *Journal of Computational and Applied Mathematics*, vol. 286, pp. 186–195, 2015.
- [27] G. Yuan, Z. Sheng, B. Wang, W. Hu, and C. Li, "The global convergence of a modified BFGS method for nonconvex functions," *Journal of Computational and Applied Mathematics*, vol. 327, pp. 274–294, 2018.
- [28] G. Yuan, Z. Wei, and X. Lu, "Global convergence of BFGS and PRP methods under a modified weak Wolfe-Powell line search," *Applied Mathematical Modelling: Simulation and Computation for Engineering and Environmental Systems*, vol. 47, pp. 811–825, 2017.

- [29] W. Burmeister, "Die Konvergenzordnung des Fletcher-Powell-Algorithmus," *Zeitschrift für Angewandte Mathematik und Mechanik. Ingenieurwissenschaftliche Forschungsarbeiten*, vol. 53, pp. 693–699, 1973.
- [30] A. I. Cohen, "Rate of convergence of several conjugate gradient algorithms," *SIAM Journal on Numerical Analysis*, vol. 9, pp. 248–259, 1972.
- [31] K. Ritter, "On the rate of superlinear convergence of a class of variable metric methods," *Numerische Mathematik*, vol. 35, no. 3, pp. 293–313, 1980.
- [32] D.-H. Li and B.-S. Tian, " $n$ -step quadratic convergence of the MPRP method with a restart strategy," *Journal of Computational and Applied Mathematics*, vol. 235, no. 17, pp. 4978–4990, 2011.
- [33] H. Huang, Z. Wei, and Y. Shengwei, "The proof of the sufficient descent condition of the Wei-Yao-Liu conjugate gradient method under the strong Wolfe-Powell line search," *Applied Mathematics and Computation*, vol. 189, no. 2, pp. 1241–1245, 2007.
- [34] Y. Shengwei, Z. Wei, and H. Huang, "A note about WYLs conjugate gradient method and its applications," *Applied Mathematics and Computation*, vol. 191, no. 2, pp. 381–388, 2007.
- [35] L. Zhang, "An improved Wei-Yao-Liu nonlinear conjugate gradient method for optimization computation," *Applied Mathematics and Computation*, vol. 215, no. 6, pp. 2269–2274, 2009.
- [36] L. Zhang, W. Zhou, and D. H. Li, "A descent modified Polak-Ribiere-Polyak conjugate gradient method and its global convergence," *IMA Journal of Numerical Analysis (IMAJNA)*, vol. 26, no. 4, pp. 629–640, 2006.
- [37] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, pp. 201–213, 2002.
- [38] M. M. Chawla, M. A. Al-Zanaidi, and D. J. Evans, "Generalized trapezoidal formulas for parabolic equations," *International Journal of Computer Mathematics*, vol. 70, no. 3, pp. 429–443, 1999.

## Research Article

# Local Negative Base Transform and Image Scrambling

Gangqiang Xiong <sup>1</sup>, Shengqian Zheng,<sup>1</sup> Jiang Wang,<sup>1</sup> Zhanchuan Cai,<sup>2</sup> and Dongxu Qi<sup>3</sup>

<sup>1</sup>Guangdong Medical University, Dongguan, China

<sup>2</sup>Macau University of Science and Technology, Taipa, Macau

<sup>3</sup>North China University of Technology, Beijing, China

Correspondence should be addressed to Gangqiang Xiong; [douglasxiong@163.com](mailto:douglasxiong@163.com)

Received 5 October 2017; Accepted 29 April 2018; Published 25 June 2018

Academic Editor: Nivaldo Rodríguez

Copyright © 2018 Gangqiang Xiong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Scrambling transform is an important tool for image encryption and hiding. A new class of scrambling algorithms is obtained by exploiting negative integer as the base of number representation to express the natural numbers. Unlike Arnold transform, the proposed scrambling transform is one-dimensional and nonlinear, and an image can be shuffled by using the proposed transform to rearrange the rows and columns of the image separately or to permute the pixels of the image after scanned into a sequence of pixels; it can be also applied to shuffle certain part region of an image. Firstly, the transformation algorithm for converting nonnegative integers in base  $B$  to the corresponding integers in base  $-B$  is given in this paper, which is the computational core of scrambling transform and the basis of studying scrambling transform. Then, the three kinds of transforms are introduced, that is, negative base transform (abbreviated as NBT), modular negative base transform (MNBT), and local negative base transform (LNBT) with three parameters, where NBT is an injection and MNBT a surjection and LNBT a bijection. The minimum transform periods of LNBT are calculated for some different values of the three parameters, and the algorithm for calculating the inverse transform of LNBT is given. The image scrambled by LNBT can be recovered by the transform period or the inverse transform. Numerical experiments show that LNBT is an efficient scrambling transform and a strong operation of confusing gray values of pixels in the application of image encryption. Therefore, the proposed transform is a novel tool for information hiding and encryption of two-dimensional image and one-dimensional audio.

## 1. Introduction

Because of the properties of visibility, intuitive, abundance in information expression and so on, digital images are widely applied in more and more fields such as remote sensing and geographical information, medical image, audiovisual art, radar navigation, copyright, and trademark; therefore, secure transmission and confidential storage of digital images have been gotten more and more attention by government, industry, and academia. Security protection of digital image includes mainly data encryption and information hiding. Image scrambling is a technique of image encryption or auxiliary encryption [1–9], and also an important method of preprocessing and postprocessing in image hiding, sharing, and digital watermarking [7, 10–17]. Particularly in the application of robust digital watermark, scrambling technique is

a very effective method against cropping attack by shuffling watermark image so that the pixels are dispersed in the position space of carrier image. The popular image scrambling methods include Arnold transform [1–3, 10–12, 18], Fibonacci transform [7, 10, 13, 14], Peano-Hilbert space filling curve [19], knight's tour [8, 20], Lucas transform [16], magic square transform [9, 17], Gray code [21, 22], cellular automata [15, 23–25], Baker map [4–6, 26], subaffine transform [27], and sampling technology [28].

Most of image scrambling transforms are two-dimensional, such as Arnold transform, baker map, Peano-Hilbert curve, John Conway game, knight's tour, magic square transform, and Fibonacci-Q transform [10]. For two-dimensional scrambling transforms, the popular scrambling transform is Arnold transform, but it requires the sizes of images are the form of  $N \times N$  [10]; Fibonacci-Q

transform is the special case of Arnold transforms [10]. Unlike Arnold transform, knight's tour [8] and Peano-Hilbert curve [19], John Conway game [15], and magic square transform [9] are not easy to calculate and are rarely used in the application of image scrambling. In general, if using John Conway game to shuffle images we first need to generate a random matrix of 0 and 1, and thus there are difficulties to restore scrambled images. In fact, Baker map is usually exploited to generate chaotic sequence on the interval  $[0, 1) \times [0, 1)$  [4], and if using baker map to shuffle images we must modify the formula of baker map.

One-dimensional scrambling transform can also be used in two-dimensional image encryption and information hiding. Battisti used Fibonacci transform to construct Fibonacci-Haar matrix and then studied digital watermarking and encryption algorithms in Fibonacci-Haar domain [7], while Li and Tsai exploited Fibonacci transform to shuffle watermark information [13, 14]. Gunjal combined Fibonacci transform with Lucas transform to construct Fibonacci-Lucas transform which is applied to shuffle watermark image [16]. Zou investigated a class of generalized Gray code and applied it to scramble color image. Two-dimensional scrambling transform or higher-dimensional scrambling transform [10] generally require that the shape of the region of scrambled image must be rectangular (i.e., the size of  $m \times n$ ) or even square such as Arnold transform. One-dimensional scrambling transform can not only permute a sequence of data, but also permute the pixels in an image region of any shape. In fact, as long as the pixels in the image region are scanned into a sequence in accordance with certain rule, we can exploit one-dimensional scrambling transform to permute these pixels.

In this paper, we use negative number as the base of numeral system to represent the natural numbers and then define negative base transform (abbreviated as NBT) with respect to radix  $-B$ ; construct modular negative base transform (abbreviated as MNBT) with respect to radix  $-B$ ; and modulo  $B^\mu$  followed by constructing local negative base transform (abbreviated as LNBT) whose radix is  $-B$  and modulo  $B^\mu$  and shift  $p$ , where  $B$  is an integer greater than 1 and  $\mu, p$  are natural numbers. In particular, LNBT is a bijective map from the subset  $\{p, p+1, \dots, p+B^\mu-1\}$  of natural numbers to the subset itself. Since LNBT is one-dimensional and nonlinear, we can use it to shuffle one-dimensional audio signal and also to shuffle two-dimensional image. The rest of the paper is organized as follows. Section 2 studies the conversion rules and conversion algorithms from base  $B$  integers to the corresponding base  $-B$  integers, which is the key of calculating NBT, MNBT, and LNBT. Section 3 gives the definitions of NBT, MNBT, and LNBT and then studies their properties and inverse transforms and last calculates the minimum transformation period of LNBT for some parameter values. Section 4 verifies the scrambling effect of LNBT by simulation experiments, which includes three kinds of scrambling algorithms: scrambling separately along x- and y-direction, scrambling a sequence of pixels after image scanned to a sequence line-by-line or scanned in zigzag order, and then tests the effect against cropping attack in digital watermarking application and the effect of image encryption

by shuffling the binary bits of image pixels. Section 5 analyzes the scrambling effect and compares it with other scrambling algorithms; finally, the conclusion is given.

## 2. Using Negative Base to Express Nonnegative Integers

*2.1. Principle of Number Representation with Negative Base.* If selecting a negative integer  $-B$  ( $B > 1$ ) as the base of number representation, we only need  $B$  figures to express all nonnegative integers, that is, we can use the symbols  $\{0, 1, 2, \dots\}$  to express all nonnegative integers [29]. In this paper, for convenience, the notation  $(a_n a_{n-1} \dots a_0)_\beta$  is used to denote an  $(n+1)$ -digit integer in base  $\beta$ , where  $\beta$  is an integer of absolute value greater than 1, and  $a_k = 0, 1, \dots, |\beta| - 1$ ,  $k = 0, 1, \dots, n$ . For the convenience of description, we also call  $a_k$  the  $k$ th digit of  $(a_n a_{n-1} \dots a_0)_\beta$ . The rest of this paper, we always suppose  $B$  is an integer greater than 1. Next, we shall discuss the calculation for conversion from nonnegative integers in base  $B$  to integers in base  $-B$ .

*2.2. Conversion Calculation from Base  $B$  Integer to Base  $-B$  Integer.* Assuming that  $B$  is an integer greater than 1, a base  $-B$  integer can be converted directly to the corresponding decimal integer expressed by the sum of the product which the  $k$ th power of  $-B$  is multiplied by the  $k$ th digit value, as do a base  $B$  integer to decimal integer, namely,

$$(a_n a_{n-1} \dots a_0)_{-B} = a_n \times (-B)^n + a_{n-1} \times (-B)^{n-1} + \dots + a_0 \times (-B)^0. \quad (1)$$

The method of seeking remainder can achieve the conversion from decimal integers to base  $-B$  integers [29]. Next, we shall investigate the algorithm for converting base  $B$  integers to base  $-B$  integers. Although we can apply the method in [29] to convert successfully base  $B$  integers to base  $-B$  integers after the base  $B$  integers being converted to decimal integers, we still need to investigate the method of straightforward conversion of base  $B$  integers to base  $-B$  integers. There are two main reasons. First, Theorem 7 needs to use the conclusion of Theorem 1, that is, we need to get the relation of the digit capacities between any base  $B$  integer and the corresponding base  $-B$  integer. Second, sometimes we need to convert base  $B$  integers directly to base  $-B$  integers. The proposed algorithm can achieve simply the conversion from base  $B$  integers to base  $-B$  integers; especially the conversion method from binary integers to negabinary integers is simpler. We will consider the issue of the digit capacity of the base  $-B$  integer converted from a known base  $B$  integer, that is, there is Theorem 1 as follows.

**Theorem 1.** *Suppose  $(a_n a_{n-1} \dots a_0)_B$  is a positive integer in base  $B$  to be converted, where  $a_n \neq 0$ , while  $(x_N x_{N-1} \dots x_0)_{-B}$  is the converted integer in base  $-B$ , where  $x_N \neq 0$ , then  $n \leq N \leq n+2$ .*

*Proof.* First,  $N \geq n$  holds obviously, followed by  $N$  must be even, since  $(x_N x_{N-1} \cdots x_0)_{-B} = \sum_{k=0}^N x_k (-B)^k < 0$  if  $N$  is odd, which contradicts  $(a_n a_{n-1} \cdots a_0)_B > 0$ .

Next, we prove  $N$  is not greater than  $n + 2$ . Denoted by  $\{(x_N x_{N-1} \cdots x_0)_{-B}\}_{\min}$  the minimum of  $N + 1$  digits integers in base  $-B$  for all satisfying  $x_N \neq 0$ . It is obvious that the value of the positive integer as expressed by  $(x_N x_{N-1} \cdots x_0)_{-B}$  is smallest only if  $x_N = 1$ ,  $x_{N-1} = x_{N-3} = \cdots = x_1 = B - 1$ ,  $x_{N-2} = x_{N-4} = \cdots = x_0 = 0$ , and there is

$$\begin{aligned} & \{(x_N x_{N-1} \cdots x_0)_{-B}\}_{\min} \\ &= (-B)^N + \sum_{k=0}^{N/2-1} x_{2k+1} (-B)^{2k+1} = \frac{B^N + B}{B + 1}. \end{aligned} \quad (2)$$

Assume  $N > n + 2$ , i.e.,  $N \geq n + 3$ , since  $(a_n a_{n-1} \cdots a_0)_B < B^{n+1}$ , we have, from (2)

$$\begin{aligned} & \{(x_N x_{N-1} \cdots x_0)_{-B}\}_{\min} - (a_n a_{n-1} \cdots a_0)_B \\ &> \{(x_N x_{N-1} \cdots x_0)_{-B}\}_{\min} - B^{n+1} = \frac{B^N + B}{B + 1} - B^{n+1} \\ &\geq \frac{B^{n+3} - B^{n+2} - B^{n+1} + B}{B + 1} \geq \frac{B^{n+2}(B - 2) + B}{B + 1} > 0. \end{aligned} \quad (3)$$

The above inequality shows that  $(a_n a_{n-1} \cdots a_0)_B$  cannot be converted to such a base  $-B$  integer of digit capacity greater than  $n + 2$ .  $\square$

Theorem 1 shows that  $(a_n a_{n-1} \cdots a_0)_B$  can be converted to a base  $-B$  integer with  $(n + 2)$ -digit at most, that is, when  $a_n$  be converted, there may be carries from the  $n$ th digit place to the  $(n + 1)$ th digit place or the  $(n + 2)$ th digit place, but not to the  $(n + 3)$ th digit place. To be more exact, there is the following Corollary 2.

**Corollary 2.** Suppose  $a_n \neq 0$ . If  $n$  is even, then  $(a_n a_{n-1} \cdots a_0)_B$  can be converted to the forms of  $(x_{n+2} x_{n+1} \cdots x_0)_{-B}$  or  $(x_n x_{n-1} \cdots x_0)_{-B}$ ; if  $n$  is odd, then  $(a_n a_{n-1} \cdots a_0)_B$  can be only converted to the form of  $(x_{n+1} x_n \cdots x_0)_{-B}$ .

*Proof.* It is obvious that  $(a_n a_{n-1} \cdots a_0)_B$  cannot be converted to the form of  $(x_m x_{m-1} \cdots x_0)_{-B}$  such that  $m < n$ . Then according to Theorem 1,  $(a_n a_{n-1} \cdots a_0)_B$  can be only converted to three forms as follows:

$$(x_{n+2} x_{n+1} \cdots x_0)_{-B}, \quad (4)$$

$$(x_{n+1} x_n \cdots x_0)_{-B}, \quad (5)$$

$$(x_n x_{n-1} \cdots x_0)_{-B}. \quad (6)$$

If  $n$  is even, it is clear that the form of (6) is likely to appear, for example, if  $a_n \neq 0$ ,  $a_k = 0$  ( $k = 0, 1, \dots, n - 1$ ), there is  $x_n = a_n$ ,  $x_k = 0$  ( $k = 0, 1, \dots, n - 1$ ) after converted to the corresponding integer in base  $-B$ . On the other hand, if taking  $a_n = B - 1$ ,  $a_{n-1} \neq 0$ ,  $a_k = 0$  ( $k = 0, 1, \dots, n - 2$ ), there is  $x_{n+2} = 1$ ,  $x_{n+1} = B - 1$ ,  $x_n = 0$ ,  $x_{n-1} = B - a_{n-1}$ ,  $x_k = 0$  ( $k = 0, 1, \dots, n - 2$ ); this indicates that  $(a_n a_{n-1} \cdots a_0)_B$  can be possibly converted to the form of (4). Since it can be

deduced from  $x_{n+1} \neq 0$  that  $(x_{n+1} x_n \cdots x_0)_{-B} < 0$ , the form of (5) cannot occur.

If  $n$  is odd, it is clear that  $(a_n a_{n-1} \cdots a_0)_B$  cannot converted to the forms of (4) and (6); otherwise, there is  $(a_n a_{n-1} \cdots a_0)_B < 0$ . So  $(a_n a_{n-1} \cdots a_0)_B$  is only converted to the form of (5).  $\square$

When an integer  $(a_n a_{n-1} \cdots a_0)_B$  in base  $B$  is converted to the corresponding integer  $(x_m x_{m-1} \cdots x_0)_{-B}$  in base  $-B$ , we suppose the conversion process is from right to left digit-by-digit. Then Corollary 2 shows that, when  $a_n$  being converted, there may be two carries need to be carried to the  $(n + 1)$ th digit place and the  $(n + 2)$ th digit place if  $n$  is even; otherwise, there is only a carry to the  $(n + 1)$ th digit place. From Corollary 2, we can get the following conversion rules, namely, Corollary 3.

**Corollary 3** (conversion rules of base  $B$  integer to base  $-B$  integer). Suppose  $(a_n a_{n-1} \cdots a_0)_B$  is a base  $B$  integer that needs to be converted, where  $a_n \neq 0$ , and  $(x_{n+2} x_{n+1} \cdots x_0)_{-B}$  is the converted base  $-B$  integer, where  $x_{n+2}$  and  $x_{n+1}$  are likely to be zeros. Let  $\{b_{n+2}, b_{n+1}, \dots, b_0\}$  be a sequence of carries in the process of conversion, and initial values  $b_k = 0$  ( $k = 0, 1, \dots, n + 2$ ). Then, in accordance with the following rules, base  $B$  integer can be correctly converted to the corresponding base  $-B$  integer.

- (i) When  $k$  is even, if  $b_k + a_k \geq B$ , then  $x_k \leftarrow b_k + a_k - B$ ,  $b_{k+2} \leftarrow 1$ ,  $b_{k+1} \leftarrow B - 1$ ; otherwise,  $x_k \leftarrow b_k + a_k$ .
- (ii) When  $k$  is odd, if  $b_k - a_k \geq 0$ , then  $x_k \leftarrow b_k - a_k$ ; otherwise,  $x_k \leftarrow b_k - a_k + B$ ,  $b_{k+1} \leftarrow b_{k+1} + 1$ .

The left arrow “ $\leftarrow$ ” in the above denotes that the value of the expression on the right side is assigned to the variable on the left side. In the following, we shall prove that Corollary 3, in fact, only proves that the converted integer in base  $-B$  is equal to the original base  $B$  integer.

*Proof.* If  $k$  is even and  $b_k + a_k \geq B$ , there is

$$\begin{aligned} b_k \times (-B)^k + a_k \times B^k &= (b_k + a_k - B) \times (-B)^k + (B - 1) \\ &\quad \times (-B)^{k+1} + 1 \times (-B)^{k+2}. \end{aligned} \quad (7)$$

Corollary 2 indicates  $0 \leq b_k \leq B - 1$ , and from  $0 < a_k \leq B - 1$ , we have  $0 \leq b_k + a_k - B < B - 1$ , i.e.,  $x_k \leftarrow b_k + a_k - B$ . Corollary 2 tells us again that, when  $a_{k-1}$  is converted, there is only a carry from the  $(k - 1)$ th digit place to the  $k$ th digit place, and when  $a_{k-2}$  is converted, there are two carries at most from the  $(k - 2)$ th digit place to the  $(k - 1)$ th digit place and the  $k$ th digit place. That is to say, when  $a_k$  being converted, the initial values  $b_{k+1} = b_{k+2} = 0$ . Therefore, it follows from (7) that  $b_{k+2} \leftarrow 1$ ,  $b_{k+1} \leftarrow B - 1$ . If  $b_k + a_k < B$ , obviously, there is

$$b_k \times (-B)^k + a_k \times B^k = (b_k + a_k) \times (-B)^k, \quad (8)$$

i.e.,  $x_k \leftarrow b_k + a_k$ .

```

Require:  $\{a_n, a_{n-1}, \dots, a_0\}$  % the sequence of digits of a base  $B$  integer
            $B$  % the base of the numeral system
Ensure:  $\{x_{n+2}, x_{n+1}, \dots, x_0\}$  % the sequence of digits of the base  $-B$  integer
Remark:  $\{x_{n+2}, x_{n+1}, \dots, x_0\}$  % satisfying  $(x_{n+2}x_{n+1} \dots x_0)_{-B} = (a_n a_{n-1} \dots a_0)_B$ 
 $k = 0$ 
While  $k \leq n + 2$  do
     $x_k = 0$ 
     $k = k + 1$ 
end while
 $k = 0$ 
while  $k \leq n$  do
    if  $k$  is even then
        if  $x_k + a_k \geq B$  then
             $x_k = x_k + a_k - B$ 
             $x_{k+1} = B - 1$ 
             $x_{k+2} = 1$ 
        else  $\{x_k + a_k < B\}$ 
             $x_k = x_k + a_k$ 
        end if
    else  $\{k$  is odd $\}$ 
        if  $x_k - a_k \geq 0$  then
             $x_k = x_k - a_k$ 
        else  $\{x_k - a_k < 0\}$ 
             $x_k = x_k - a_k + B$ 
             $x_{k+1} = x_{k+1} + 1$ 
        end if
    end if
     $k = k + 1$ 
end while

```

ALGORITHM 1: Converting base  $B$  integer to base  $-B$  integer  $(\{a_n, a_{n-1}, \dots, a_0\})$ .

If  $k$  is odd and  $b_k - a_k \geq 0$ , then  $b_k \times (-B)^k + a_k \times B^k = (b_k - a_k) \times (-B)^k$ , and the inequality  $b_k - a_k < b_k \leq B - 1$  indicates that  $x_k \leftarrow b_k - a_k$  is true. If  $b_k - a_k < 0$ , we have

$$b_k \times (-B)^k + a_k \times B^k = (b_k - a_k + B) \times (-B)^k + 1 \times (-B)^{k+1}. \quad (9)$$

The inequality  $0 < B - a_k \leq b_k - a_k + B \leq B - 1$  and (9) imply that  $x_k \leftarrow b_k - a_k + B$  and there is a carry 1 from the current digit place to the  $(k + 1)$ th digit place. In addition, it can be known from Corollary 2 that, when  $a_{k-1}$  being converted, maybe there exist two carries from the  $(k - 1)$ th digit place to the  $k$ th digit place and the  $(k + 1)$ th digit place, namely,  $b_{k+1}$  may not be equal to 0 when  $a_k$  being converted. Therefore,  $b_{k+1}$  need to accumulate the carry 1, i.e.,  $b_{k+1} \leftarrow b_{k+1} + 1$  is true.  $\square$

So we can obtain Algorithm 1 from Corollary 3.

For binary, the conversion process is more concise, and the specific conversion rules are described in the following Corollary 4.

**Corollary 4** (conversion rules of binary positive integers to negabinary integers). *Suppose  $(a_n a_{n-1} \dots a_0)_2$  is a binary positive integer to be converted, where  $a_n \neq 0$ ;  $(x_{n+2} x_{n+1} \dots x_0)_{-2}$  is*

*the converted negabinary integer, and the initial values  $x_k = 0$  for  $k = 0, 1, \dots, n + 2$ . Then the following conversion rules hold.*

- (i) *When  $k$  is odd and  $a_k = 1$ , if  $x_k = 0$ , then  $x_k \leftarrow 1$ ,  $x_{k+1} \leftarrow 1$ ; otherwise  $x_k \leftarrow 0$ .*
- (ii) *When  $k$  is even and  $a_k = 1$ , if  $x_k = 0$ , then  $x_k \leftarrow 1$ ; otherwise  $x_k \leftarrow 0$ ,  $x_{k+1} \leftarrow 1$ ,  $x_{k+2} \leftarrow 1$ .*
- (iii) *If  $a_k = 0$ , then the value of  $x_k$  keeps changeless.*

### 3. Scrambling Transform Based on Negative Base

Image scrambling, which can change a certain image into another meaningless image, is a special transform from a subset of natural numbers to the subset itself. Rearranging the position of the pixels in the position space of image by permutation on a finite set of the natural numbers can achieve the purpose of scrambling image, such as Arnold transform, baker map, and knight's tour. Next, we define a new class of scrambling transforms based on the principle of number representation in negative bases.

**3.1. Negative Base Transform.** Unlike the common image scrambling transform, the scrambling transform studied in this paper is one-dimensional, which is such a transform of

TABLE 1: NBT of the integers between 0 and 15 with parameter  $-2$ .

$x$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
NBT	0	1	6	7	4	5	26	27	24	25	30	31	28	29	18	19

a certain subset of the natural numbers to the subset itself. First, we define negative base transform in the following.

**Definition 5** (negative base transform). For any  $x \in \mathbb{N}$ , if transform  $T_{-B}$  satisfies

$$T_{-B}(x) = \sum_{k=0}^n x_k B^k, \quad (10)$$

then  $T_{-B} : \mathbb{N} \rightarrow \mathbb{N}$  is referred to as negative base transform with respect to base  $-B$ , abbreviated as NBT, and  $B$  is called transformation parameter of NBT, where  $x_k$  is the  $k$ th digit place of the converted integer  $(x_n x_{n-1} \cdots x_0)_{-B}$  in base  $-B$  corresponding to  $x$ .

From Definition 5, we know that, for computing  $T_{-B}(x)$ , the natural number  $x$  must be converted to the form of  $(x_n x_{n-1} \cdots x_0)_{-B}$ , and then calculate  $T_{-B}(x) = \sum_{k=0}^n x_k B^k$ . For example, since  $7 = (11011)_{-2}$ , then  $T_{-2}(7) = 2^4 + 2^3 + 2^1 + 2^0 = 27$ . Table 1 gives the values of NBT with parameter  $-2$  of the natural numbers from 0 to 15. According to Table 1, we can know that, after the natural numbers between 0 and 15 is transformed by NBT, the order of the transform  $T_{-2}(x)$  is already different from the original order of independent variable  $x$ , and some of  $T_{-2}(x)$  are no longer in the set  $\{0, 1, 2, \dots, 15\}$ , e.g.,  $T_{-2}(7) = 27$ .

**3.2. Modular Negative Base Transform.** Assume  $\mathbb{N}[\alpha, \beta] = \{x; x \in \mathbb{N} \cap [\alpha, \beta]\}$  denote a subset of the natural numbers from  $\alpha$  to  $\beta$ , where  $\alpha, \beta \in \mathbb{N}$ ,  $\alpha \leq \beta$ , and  $\mathbb{N}$  denotes the set of natural numbers. In general, Image scrambling transform is a bijective map from a finite set  $\mathbb{N}[\alpha_1, \beta_1] \times \mathbb{N}[\alpha_2, \beta_2]$  to  $\mathbb{N}[\alpha_1, \beta_1] \times \mathbb{N}[\alpha_2, \beta_2]$ . Unfortunately, NBT is an injective map from the set of natural numbers  $\mathbb{N}$  to itself, but not surjective, and cannot be used directly in application of image scrambling. Hence we shall make some modifications for NBT so that the modified transform is a bijective map from  $\mathbb{N}[\alpha, \beta]$  to  $\mathbb{N}[\alpha, \beta]$ . According to the transform NBT, we shall introduce another new transform from  $\mathbb{N}$  to  $\mathbb{N}[0, B^\mu - 1]$  as follows.

**Definition 6** (modular negative base transform). For  $x \in \mathbb{N}$ , if transform  $M_{-B, \mu}$  satisfies

$$M_{-B, \mu}(x) = T_{-B}(x) \pmod{B^\mu}, \quad (11)$$

then transform  $M_{-B, \mu}$  is called modular negative base transform with respect to base  $-B$  and modulo  $B^\mu$ , abbreviated as MNBT, where  $\mu$  is an integer greater than 0.

Obviously, MNBT is a surjection from  $\mathbb{N}$  to  $\mathbb{N}[0, B^\mu - 1]$ , but not an injection. In order to make MNBT a bijection, it

is necessary to restrict the definition domain of MNBT such that Theorem 7 is true in the following.

**Theorem 7.** *If the definition domain of MNBT is  $\mathbb{N}[p, p + B^\mu - 1]$ , then MNBT is a bijective map from  $\mathbb{N}[p, p + B^\mu - 1]$  to  $\mathbb{N}[0, B^\mu - 1]$ .*

*Proof.* Suppose  $M_{-B, \mu}$  is an MNBT as defined in (11). In fact, we only need to prove that  $M_{-B, \mu}$  is injective if  $p = 0$ , that is, prove that

$$\begin{aligned} &\text{for any } x_1, x_2 \in \mathbb{N}[0, B^\mu - 1] \\ &\text{satisfying } x_1 \neq x_2, \end{aligned} \quad (12)$$

$$\text{there is } M_{-B, \mu}(x_1) \neq M_{-B, \mu}(x_2).$$

Since any integer in  $\mathbb{N}[0, B^\mu - 1]$  can only be expressed as a  $\mu$ -digit integer in base  $B$  at most, it follows from Theorem 1 that  $x_1$  and  $x_2$  only are represented as a  $(\mu + 2)$ -digit integer in base  $-B$  at most. Without loss of generality, we expressed  $x_1$  and  $x_2$  as  $(\mu + 2)$ -digit integers in base  $-B$ ; if the number of digits is not enough, we fill 0 in front of them, that is,

$$\begin{aligned} x_1 &= (x_{1, \mu+1} x_{1, \mu} \cdots x_{1, 0})_{-B}, \\ x_2 &= (x_{2, \mu+1} x_{2, \mu} \cdots x_{2, 0})_{-B}, \end{aligned} \quad (13)$$

where for  $j = 1, 2$  and  $k = 0, 1, \dots, \mu + 1$ ,  $x_{j, k}$  is possibly equal to 0. Let

$$\begin{aligned} y_1 &= T_{-B}(x_1) = \sum_{k=0}^{\mu+1} x_{1, k} B^k, \\ y_2 &= T_{-B}(x_2) = \sum_{k=0}^{\mu+1} x_{2, k} B^k. \end{aligned} \quad (14)$$

Assume  $M_{-B, \mu}$  is not an injection; there is  $x_1 \neq x_2$  such that  $M_{-B, \mu}(x_1) = M_{-B, \mu}(x_2)$ , namely,  $y_1 \equiv y_2 \pmod{B^\mu}$ . Therefore, we can infer that

$$x_{1, k} = x_{2, k}, \quad k = 0, 1, \dots, \mu - 1. \quad (15)$$

Since  $x_1 \neq x_2$  implies  $y_1 \neq y_2$ , then there must be

$$x_{1, \mu} \neq x_{2, \mu} \quad (16)$$

$$\text{or } x_{1, \mu+1} \neq x_{2, \mu+1}.$$

If we can prove that (16) is not true, the fact that  $M_{-B, \mu}$  is an injective map is proven, i.e., we complete the proof of the theorem.

**Require:**  $x$  % the nonnegative integer  
 $B, \mu, p$  % the parameters of LNBT  
**Ensure:**  $y = f_{-B, \mu, p}(x)$  % the value of LNBT  
**Remark:**  $y$  %  $y$  satisfying  $y = f_{-B, \mu, p}(x)$   
 $\{x_n, x_{n-1}, \dots, x_0\} = \text{Converting base } B \text{ integer to base } -B \text{ integer } (\{a_n, a_{n-1}, \dots, a_0\})$   
%  $x$  satisfying  $x = \sum_{k=0}^n x_k (-B)^k$   
 $y = (\sum_{k=0}^n x_k B^k \bmod B^\mu) + p$

ALGORITHM 2: Calculating LNBT ( $x$ ).

TABLE 2: The results of LNBT.

$x$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$f_{-2,4,1}(x)$	2	7	8	5	6	11	12	9	10	15	16	13	14	3	4	1

In accordance with (15), there is

$$\begin{aligned}
 |x_1 - x_2| &= \left| \sum_{k=0}^{\mu+1} x_{1,k} (-B)^k - \sum_{k=0}^{\mu+1} x_{2,k} (-B)^k \right| \\
 &= \left| x_{1,\mu+1} (-B)^{\mu+1} + x_{1,\mu} (-B)^\mu - x_{2,\mu+1} (-B)^{\mu+1} \right. \\
 &\quad \left. - x_{2,\mu} (-B)^\mu \right| = B^\mu \left| B(x_{2,\mu+1} - x_{1,\mu+1}) \right. \\
 &\quad \left. + (x_{1,\mu} - x_{2,\mu}) \right|. \tag{17}
 \end{aligned}$$

If  $x_{1,\mu+1} \neq x_{2,\mu+1}$ , then, from (17), there is

$$\begin{aligned}
 |x_1 - x_2| &= B^\mu \left| B(x_{2,\mu+1} - x_{1,\mu+1}) + (x_{1,\mu} - x_{2,\mu}) \right| \\
 &\geq B^\mu \left( B|x_{2,\mu+1} - x_{1,\mu+1}| - |x_{1,\mu} - x_{2,\mu}| \right) \\
 &\geq B^\mu \left( B - |x_{1,\mu} - x_{2,\mu}| \right) \geq B^\mu. \tag{18}
 \end{aligned}$$

This contradicts obviously  $x_1, x_2 \in \mathbb{N}[0, B^\mu - 1]$ ; thus this shows  $x_{1,\mu+1} = x_{2,\mu+1}$ .

If  $x_{1,\mu} \neq x_{2,\mu}$ , then, from (17) and  $x_{1,\mu+1} = x_{2,\mu+1}$ , there is

$$\begin{aligned}
 |x_1 - x_2| &= B^\mu \left| B(x_{2,\mu+1} - x_{1,\mu+1}) + (x_{1,\mu} - x_{2,\mu}) \right| \\
 &= B^\mu |x_{1,\mu} - x_{2,\mu}| \geq B^\mu. \tag{19}
 \end{aligned}$$

This also contradicts  $x_1, x_2 \in \mathbb{N}[0, B^\mu - 1]$  and shows that  $x_{1,\mu} \neq x_{2,\mu}$  is not true. Therefore we have already completed the proof of Theorem 7.  $\square$

It is noteworthy that MNBT is a bijective map from  $\mathbb{N}[p, p + B^\mu - 1]$  to  $\mathbb{N}[0, B^\mu - 1]$ , but not from  $\mathbb{N}[p, p + B^\mu - 1]$  to the set itself. If taking  $p = 0$ , then MNBT is a bijective map from  $\mathbb{N}[0, B^\mu - 1]$  to  $\mathbb{N}[0, B^\mu - 1]$ , but in this case, MNBT will map some points to themselves, i.e., the transformation periods of these points are 1, for example,  $M_{-B, \mu}(0) = 0$ . Thus MNBT cannot be used directly in the application of image scrambling and need to be modified further. In fact, combining MNBT with translational transform with respect to shift  $p$ , we can obtain a permutation from  $\mathbb{N}[p, p + B^\mu - 1]$  to  $\mathbb{N}[p, p + B^\mu - 1]$ , which is of course a bijective map.

**3.3. Local Negative Base Transform.** Local negative base transform can be constructed by combining MNBT with a translational transform with shift  $p$ .

**Definition 8** (local negative base transform). If the map  $f_{-B, \mu, p} : \mathbb{N}[p, p + B^\mu - 1] \rightarrow \mathbb{N}[p, p + B^\mu - 1]$  satisfies

$$f_{-B, \mu, p}(x) = M_{-B, \mu}(x) + p, \quad x \in \mathbb{N}[p, p + B^\mu - 1], \tag{20}$$

then  $f_{-B, \mu, p}$  is called local negative base transform with respect to base  $-B$ , modulo  $B^\mu$ , and shift  $p$ , abbreviated as LNBT.

In Definition 8,  $f_{-B, \mu, p}$  is also referred to as LNBT with parameter  $(B, \mu, p)$ . Obviously, if  $p = 0$ , LNBT is simplified as MNBT. The following corollary is easily established in accordance with Theorem 7.

**Corollary 9.** LNBT is a bijective map from  $\mathbb{N}[p, p + B^\mu - 1]$  to  $\mathbb{N}[p, p + B^\mu - 1]$ , i.e., LNBT is a permutation on  $\mathbb{N}[p, p + B^\mu - 1]$ .

Corollary 9 shows that LNBT is a scrambling transform from  $\mathbb{N}[p, p + B^\mu - 1]$  to itself. Next, we describe the calculation method of LNBT, shown as the Algorithm 2.

For example, if we take  $(B, \mu, p) = (2, 4, 1)$ , we can calculate easily the values of LNBT according to Algorithm 2, and the calculated results as shown in Table 2, where  $f_{-2,4,1}$  in Table 2 is defined as (20).

It is obvious that scrambling transform LNBT can map the subset  $S$  of  $\mathbb{N}[p, p + B^\mu - 1]$  to  $\mathbb{N}[p, p + B^\mu - 1]$ ; moreover, LNBT can map (scatter) uniformly the entries of the subset  $S$  to the set  $\mathbb{N}[p, p + B^\mu - 1]$  after many times of LNBT. Figure 1 is the result after 19 times of permutation by LNBT, and it can be seen that the black pixels have been scattered uniformly onto the entire position space of the images; especially, the effect is better in the case of implementing transformation after the image is scanned into one-dimensional sequence (the form of row vector). Generally, as long as  $B$  is even and  $\mu$  is large enough, the scrambling effect is very good. When the pixels are permuted along x-direction, the positional relationship between adjacent pixels at y-direction cannot be changed. Likewise, when permuting the pixels along y-direction, the positional relationship between adjacent pixels at x-direction

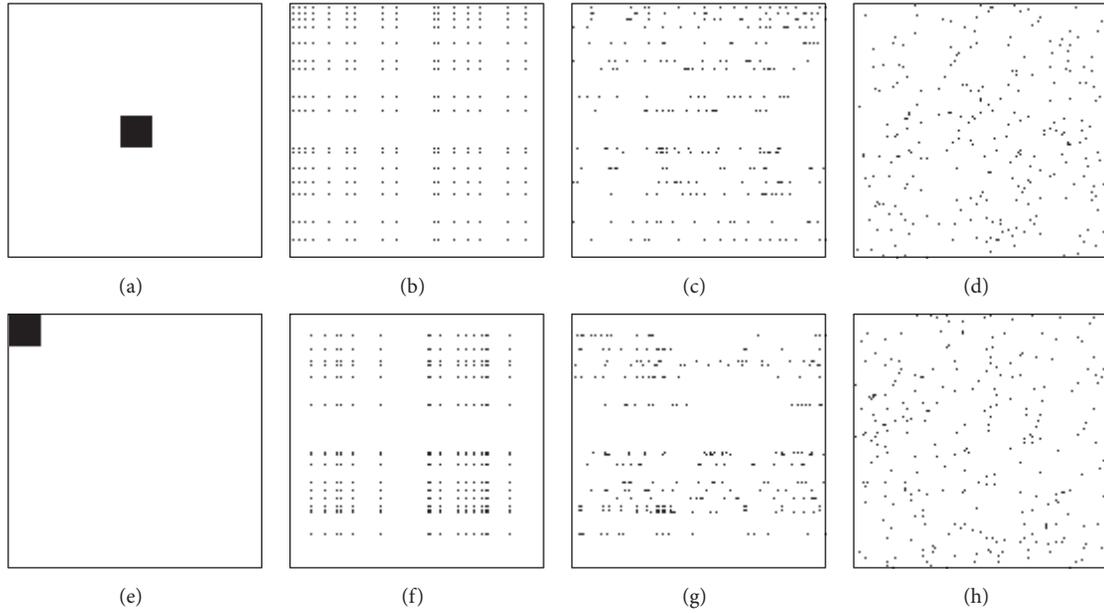


FIGURE 1: Images of size  $128 \times 128$  and black area of size  $16 \times 16$ , 19 times of LNBT: (a) black pixels in the center of an image, (e) black pixels in the upper-left of another image; (b) and (f) are the scrambled images of (a) and (e) with parameter  $(2, 7, 1)$  along  $x$ - and  $y$ -direction, respectively, (c) and (g) are the scrambled images of (a) and (e) with parameter  $(2, 14, 1)$  after scanned into a sequence line-by-line, and (d) and (h) are the scrambled images of (a) and (e) with parameter  $(2, 14, 1)$  after scanned into a sequence in zigzag order.

cannot be changed. Therefore, the pixels shuffled along  $x$ - and  $y$ -directions separately will result in the black pixels aligned, as shown in Figures 1(b) and 1(f).

**3.4. Minimum Transformation Period and Inverse Transformation.** In the application of digital watermarking and image encryption, it is often necessary to restore the scrambled image. There are two kinds of methods for restoring scrambled image, that is, by inverse scrambling transform and by minimum transformation period.

**3.4.1. Minimum Transformation Period.** Like Arnold transform, MNBT and LNBT exist transformation periods, as shown in Theorem 10. In Theorem 10, we use the notations  $M_{-B,\mu}^{\lambda_1}(x)$  and  $f_{-B,\mu,p}^{\lambda_2}(x)$ , which are defined by

$$\begin{aligned} M_{-B,\mu}^k(x) &= M_{-B,\mu}(M_{-B,\mu}^{k-1}(x)), \\ M_{-B,\mu}^1(x) &= M_{-B,\mu}(x), \\ f_{-B,\mu,p}^k(x) &= f_{-B,\mu,p}(f_{-B,\mu,p}^{k-1}(x)), \\ f_{-B,\mu,p}^1(x) &= f_{-B,\mu,p}(x), \end{aligned} \quad (21)$$

where  $M_{-B,\mu}$  and  $f_{-B,\mu,p}$  as defined by (11) and (20).

**Theorem 10** (period existence theorem for LNBT and MNBT). *There exists a positive integer  $\lambda_1$  such that  $M_{-B,\mu}^{\lambda_1}(x) = x$  for any  $x \in \mathbb{N}[0, B^\mu - 1]$  and a positive integer  $\lambda_2$  such that  $f_{-B,\mu,p}^{\lambda_2}(x) = x$  for any  $x \in \mathbb{N}[p, p + B^\mu - 1]$ .*

*Proof.* We only prove that  $f_{-B,\mu,p}$  exists the transformation period and can also use the same method to prove  $M_{-B,\mu}$ . Since the number of entries in the set  $\mathbb{N}[p, p + B^\mu - 1]$  is finite, we only prove that, for any  $x \in \mathbb{N}[p, p + B^\mu - 1]$ , there is  $\lambda_x$  such that  $f_{-B,\mu,p}^{\lambda_x}(x) = x$ ; in fact, the transform period of  $f_{-B,\mu,p}$  is the common multiple of all  $\lambda_x$ .

Suppose for any positive integer  $\lambda$  there is  $x_0$  such that  $f_{-B,\mu,p}^\lambda(x_0) \neq x_0$ . Let  $z_0 = f_{-B,\mu,p}^\lambda(x_0)$ , for any positive integer  $\lambda$ , we have, from the fact that  $f_{-B,\mu,p}$  is bijective,

$$\begin{aligned} z_0 &= f_{-B,\mu,p}(x_0) \neq f_{-B,\mu,p}(f_{-B,\mu,p}^\lambda(x_0)) \\ &= f_{-B,\mu,p}^{\lambda+1}(x_0) = f_{-B,\mu,p}^\lambda(f_{-B,\mu,p}(x_0)) \\ &= f_{-B,\mu,p}^\lambda(z_0). \end{aligned} \quad (22)$$

On the other hand, let  $y_k = f_{-B,\mu,p}^k(x_0)$ ,  $k = 1, 2, \dots, B^\mu$ , from  $x_0, y_k \in \mathbb{N}[p, p + B^\mu - 1]$  and pigeonhole principle; we know that there is  $i \neq j$  such that  $y_i = y_j$ . Without loss of generality, assume  $j > i$ , then from the fact that  $f_{-B,\mu,p}$  is a bijective map and  $f_{-B,\mu,p}^i(x_0) = f_{-B,\mu,p}^j(x_0)$ , we can conclude  $f_{-B,\mu,p}(x_0) = f_{-B,\mu,p}^{j-i}(x_0)$ , namely,  $z_0 = f_{-B,\mu,p}^{j-i}(z_0)$ . This contradicts (22). Therefore, there must exist a positive integer  $\lambda_x$  such that  $f_{-B,\mu,p}^{\lambda_x}(x) = x$ . The proof of this theorem has already completed.  $\square$

If using the periodicity of scrambling transform to restore scrambled image, we must find the minimum transformation period of LNBT, but Theorem 10 does not give a calculation

TABLE 3: Transformation periods on  $\mathbb{N}[p, p + B^\mu - 1]$ .

$\mu$	$p$	$B=2$	$B=3$	$B=4$	$B=5$	$B=6$	$B=7$	$B=8$	$B=9$	$B=10$
2	1	4	6	16	10	36	14	64	18	100
2	2	2	6	8	10	6	14	32	18	10
2	3	4	2	16	10	12	14	64	6	100
2	4	1	6	2	10	6	14	16	18	10
3	1	8	18	64	50	216	98	512	162	1000
3	2	2	18	32	50	18	98	256	162	50
3	3	8	6	64	50	72	98	512	54	1000
3	4	2	18	4	50	18	98	128	162	50
4	1	16	18	256	50	1296	98	4096	162	10000
4	2	4	18	128	50	108	98	2048	162	500
4	3	16	6	256	50	432	98	4096	54	10000
4	4	4	18	16	50	36	98	1024	162	500
5	1	32	54	1024	250	7776	686	32768	1458	100000
5	2	8	54	512	250	648	686	16384	1458	5000
5	3	32	18	1024	250	2592	686	32768	486	100000
5	4	8	54	64	250	216	686	8192	1458	5000

method of the minimum transform period. In general, it is not easy to determine the minimum transform period of MNBT and LNBT. Table 3 is the minimum transform periods of MNBT and LNBT with

$$(B, \mu, p) \in \{(i, j, k) : i = 2, 3, \dots, 10; j = 2, 3, 4, 5; k = 1, 2, 3, 4\} \quad (23)$$

for  $x \in \mathbb{N}[p, p + B^\mu - 1]$ . For example, for  $(B, \mu, p) = (3, 2, 1)$ , the minimum transformation periods of all entries on  $\mathbb{N}[1, 9]$  are 3 and 6, respectively. Then the minimum transformation period of LNBT with parameter  $(B, \mu, p) = (3, 2, 1)$  is 6 as shown Table 3.

Looking at Table 3, we can get the conclusion as shown in Proposition 11.

**Proposition 11.** *The set  $\mathbb{N}[p, p + B^\mu - 1]$  is the definition domain of LNBT, where  $p$  is nonnegative. For different  $B$  and  $p$ , there is the following conclusion.*

(i) *The minimum transformation periods of LNBT are not greater than  $B^\mu$ . If  $B=2,4,8$  and  $p$  is odd, or  $B$  is even and  $p = 1$ , the minimum transformation periods of LNBT is equal to  $B^\mu$ .*

(ii) *If  $p$  is even, the difference between  $B^\mu$  and the minimum transformation period is bigger.*

(iii) *If  $B$  is odd, the difference between  $B^\mu$  and the minimum transformation period is also bigger.*

Conclusion (i) in Proposition 11 is very important, and in the application of image scrambling, we often take  $B$  as an even number and  $p$  as an odd number; especially, take  $p = 1$  and  $B = 2$ . In general, the effect is better if the difference between  $B^\mu$  and the minimum transformation period is smaller. In fact,  $B^\mu$  is often a transformation period of LNBT with parameter  $(B, \mu, p)$ , but is not necessarily the minimum transformation period. For example, LNBT with parameter  $(-2, 3, 1)$  is a cycle on the set  $\mathbb{N}[1, 8]$ , namely,  $(1\ 2\ 7\ 4\ 5\ 6\ 3\ 8)$ , and this shows that the minimum

transformation period of LNBT with parameter  $(-2, 3, 1)$  is equal to 8. Figure 2 is the experimental results by using minimum transformation period to restore scrambled image, where the parameter values are  $(2, 9, 1)$  in (b) and (c).

**3.4.2. Inverse Transforms of MNBT and LNBT.** It is not easy to compute the minimum transformation period of MNBT and LNBT for different parameters  $(B, \mu, p)$ ; on the other hand, although the minimum transformation period is sometimes known, but the minimum transformation period is possibly large, so that the computational complexity of restoring scrambled image becomes large. For example, if we use LNBT with parameters  $(2, 18, 1)$  to shuffle the pixels of an image of size  $512 \times 512$  after scanned into a sequence, then the minimum transformation period is 262144. Therefore, the calculation is sometimes difficult by using the periodicity of LNBT to restore scrambled image, while the calculation of restoring scrambled image is more concise and faster by inverse transform of LNBT.

Let  $y = T_{-B}(x)$ ; (10) implies  $y = \sum_{k=0}^n x_k B^k$ , i.e.,  $y = (x_n x_{n-1} \cdots x_0)_B$  and  $x = (x_n x_{n-1} \cdots x_0)_{-B}$ . Thus, from (10), the inverse map  $T_{-B}^{-1}$  of NBT is obtained directly, namely,

$$x = T_{-B}^{-1}(y) = \sum_{k=0}^n x_k (-B)^k, \quad (24)$$

and  $x_k$  satisfies the condition  $y = (x_n x_{n-1} \cdots x_0)_B$ . Equation (24) tells us the process of calculation, which convert  $y$  to an integer in base  $B$  and then calculate the expression of right side in (24).

Suppose the definition domain of MNBT is  $\mathbb{N}[0, B^\mu - 1]$ . Next, we consider the calculation for the inverse map of  $y = M_{-B, \mu}(x)$ . According to Theorem 1 and (11), we can conclude that there exists a nonnegative integer  $k$  such that  $T_{-B}(x) =$

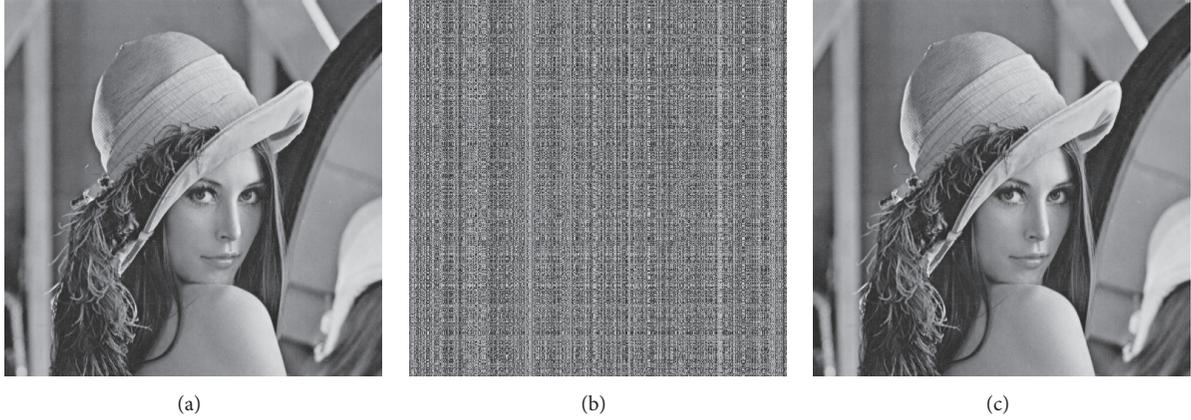


FIGURE 2: Restoring scrambled images by the minimum transformation period: (a) is an original images of size  $512 \times 512$ ; (b) is the scrambled images of (a) using 22 times of LNBT to shuffle the pixels along x- and y-direction separately; (c) is the restored image using 490 times of LNBT to permute the pixels along x- and y-direction separately from (b).

$kB^\mu + y$ . From (24), we can get the inverse map  $M_{-B,\mu}^{-1}$  of MNBT, namely,

$$x = M_{-B,\mu}^{-1}(y) = \sum_{k=0}^n x_k (-B)^k, \quad (25)$$

where  $x_k$  satisfies  $kB^\mu + y = (x_n x_{n-1} \cdots x_0)_B$ . In order to compute  $x$  in (25), we must first determine the value of  $k$ . In fact, the selected  $k$  must make the calculated  $x$  from (25) satisfy  $x \in \mathbb{N}[0, B^\mu - 1]$ , and from the fact that MNBT is permutation on  $\mathbb{N}[0, B^\mu - 1]$ , we know that  $k$  is uniquely determined.

From (25) and (20), we can get the inverse map  $f_{-B,\mu,p}^{-1}$  of LNBT, namely,

$$x = f_{-B,\mu,p}^{-1}(y) = \sum_{k=0}^n x_k (-B)^k + p, \quad (26)$$

$$kB^\mu + y - p = (x_n x_{n-1} \cdots x_0)_B, \quad (27)$$

$$k \text{ satisfying } x \in \mathbb{N}[p, B^\mu + p - 1] \text{ in Equation (26)}. \quad (28)$$

Therefore, the inverse transform  $x = f_{-B,\mu,p}^{-1}(y)$  can be calculated from (26), (27), and (28), and the specific details are depicted as Algorithm 3.

Figure 3 is the experimental results of image scrambled by LNBT and image restored by inverse LNBT, where the parameter is  $(2, 9, 1)$  and the size of image is  $512 \times 512$ . Image (b) is the scrambled image by 19 times of LNBT along x- and y-direction separately and image (c) is the restored image by 19 times of inverse transform of LNBT.

## 4. Experiments of Image Scrambling

*4.1. Scrambling along X- and Y-Direction Separately.* Unlike Arnold transform, LNBT is a bijective map from the definition domain  $\mathbb{N}[p, p + B^\mu - 1]$  to definition domain itself, and if

using LNBT to shuffle two-dimension image, we can permute the pixels of image along x- and y-direction. Selection of parameter  $(B, \mu, p)$  is relation to the size of image to be scrambled; suppose the size of image is  $M \times N$ , we should select  $B_y^\mu = M$  at y-direction and  $B_x^\mu = N$  at x-direction and take  $p$  as an odd number. For example, Figure 4(a) is an image of size  $216 \times 256$ ; then we can select the parameter  $(2, 8, 1)$  at x-direction and  $(6, 3, 1)$  at y-direction. Figure 4(e) is another image of size  $256 \times 256$ ; then we can select the same parameter  $(2, 8, 1)$  both x- and y-direction. From Figure 4, we find that the scrambled pixels are aligned along x- and y-direction, respectively.

### 4.2. Scrambling after Scanned into a Sequence

*(i) Scan Line-by-Line.* In order to eliminate the alignment phenomenon of pixels as described above after image was scrambled along x- and y-direction separately, the easiest method is that we use LNBT to permute the sequence of pixels after two-dimensional image is scanned into a one-dimensional sequence line-by-line. Figure 5(a) is an image of size  $216 \times 216$ , and after being scanned into a one-dimensional vector, the length of the corresponding one-dimension vector is  $6^6$ ; thus we select the parameter  $(6, 6, 1)$  to shuffle the pixels. Figure 5(e) is another image of size  $256 \times 256$ , and we can select the parameter  $(2, 16, 1)$  to shuffle the pixels after being scanned into a one-dimension vector. From Figures 4 and 5, we find that the smaller the parameter B is, the better the scrambling effect is.

*(2) Scan in Zigzag Order.* After scanning a two-dimension image into a sequence of pixels line-by-line, if taking  $B = 2$  we can obtain a good scrambling result by many times of LNBT, as shown in Figure 5(h), but if selecting  $B = 6$ , the scrambling effect is not very good. In Figure 6, we first use the zigzag scan technique to change a two-dimension image into the corresponding one-dimension vector and then use LNBT to permute this sequence of pixels; last, we use the same zigzag order to arrange the scrambled sequence and

```

Require:  $y$            % non-negative integer
            $B, \mu, p$       % the parameters of inverse LNBT
Ensure:  $x$            % the value of inverse LNBT
Remark:  $x$            %  $x$  satisfying  $x = f_{-B, \mu, p}^{-1}(y)$ 
 $x = -1$ 
 $k = 0$ 
while  $x < p$  or  $x > p + B^\mu - 1$  do
   $z = kB^\mu + y - p$ 
   $(x_n x_{n-1} \cdots x_0)_B = z$       % converting  $z$  into the sequence of base  $B$ 
   $x = \sum_{j=0}^n x_j (-B)^j$ 
   $k = k + 1$ 
7: end while

```

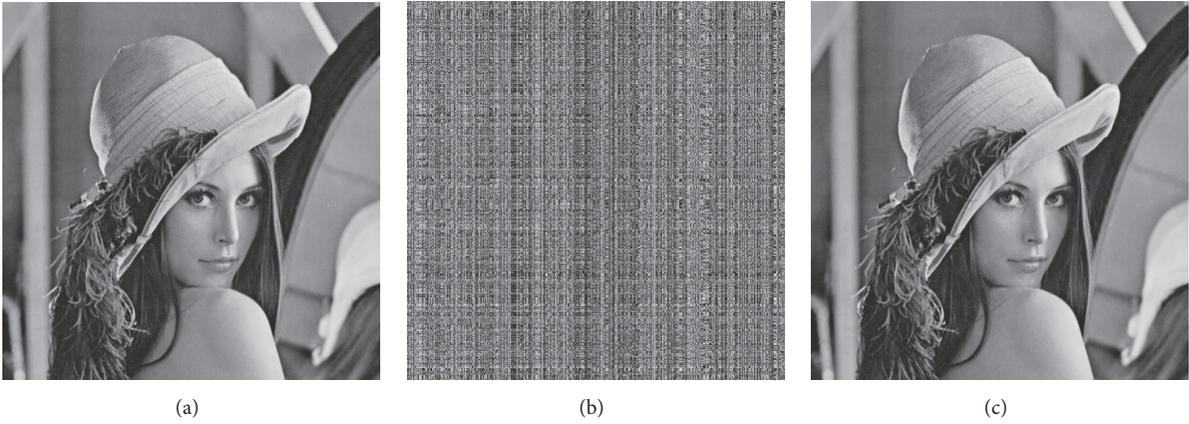
ALGORITHM 3: Calculating inverse transform of LNBT ( $y$ ).

FIGURE 3: Restoring scrambled image by 19 times of inverse map of LNBT: (a) original image of size 512x512; (b) scrambled image; (c) restored image.

obtain the corresponding scrambled image. Compared with Figure 5, we find the effect of image scrambling becomes better in Figure 6.

**4.3. Irregular Region Scrambling of Image.** We consider two cases that the size of image cannot express as the form of  $B_1^{\mu_1} \times B_2^{\mu_2}$  and the image region to be shuffled is irregular. Since LNBT is a kind of one-dimensional scrambling transform from  $\mathbb{N}[p, p + B^\mu - 1]$  to  $\mathbb{N}[p, p + B^\mu - 1]$ , if using LNBT to permute two-dimensional image, we must change the image into the corresponding one-dimensional sequence of pixels, and then we permute the sequence of pixels; of course, we can also permute an image along the horizontal and vertical direction separately. Therefore, we can shuffle image region of any shape and images of any size. In the application of part information hiding and encryption of image (e.g., sensitive area scrambling) [30–32], unlike Arnold transform, LNBT can shuffle image region of any shape.

(i) *Case 1: Irregular Size.* If the size of image cannot be expressed as the form of  $B_1^{\mu_1} \times B_2^{\mu_2}$ , we cannot use directly LNBT with parameter  $(B_1, \mu_1, p_1)$  to permute the x-coordinate and with parameter  $(B_2, \mu_2, p_2)$  to permute the y-coordinate. Therefore, such size of image is called irregular

size in this paper. Suppose the size of an image is  $M \times N$ , and for any positive integers  $B_1$  and  $\mu_1$  satisfying  $M \neq B_1^{\mu_1}$  or any positive integers  $B_2$  and  $\mu_2$  satisfying  $N \neq B_2^{\mu_2}$ . Without loss of generality, we suppose  $M$  cannot be expressed as the form of  $B^\mu$ , then there exist  $B_x, \mu_x, X$ , and  $R_x$  such that  $M = X \times B_x^{\mu_x} + R_x$  where  $1 \leq R_x < B_x^{\mu_x}$  and thus the set  $\mathbb{N}[1, M]$  can be divided into  $X + 1$  subsets. For the first  $X$  subsets, we select LNBT of parameters  $(B_x, \mu_x, k)$  to shuffle  $\mathbb{N}[\alpha B_x^{\mu_x} + 1, (\alpha + 1)B_x^{\mu_x}]$  where  $k = \alpha B_x^{\mu_x} + 1, \alpha = 0, 1, \dots, X - 1$ . For the last subset  $\mathbb{N}[XB_x^{\mu_x} + 1, M]$ , first, extend the left end of  $\mathbb{N}[XB_x^{\mu_x} + 1, M]$  to  $\mathbb{N}[M - XB_x^{\mu_x} + 1, M]$ ; then shuffle the extended subset  $\mathbb{N}[M - XB_x^{\mu_x} + 1, M]$  after the first  $X$  subsets were shuffled. Figure 7 is the scrambling result of the image of size 720x648. First, permute x-coordinate between 1 and 512 as shown in Figure 7(b), and next, permute x-coordinate between 209 and 720 as shown in Figure 7(c). The results of permuting y-coordinates are shown in Figures 7(d) and 7(e).

(2) *Case 2: Irregular Shape.* Sometimes we require shuffling a certain region of image [30–32]; in this case, the region to be shuffled may be irregular with respect to size or shape, and therefore it is difficult for two-dimensional scrambling transform to shuffle such irregular region. Unlike Arnold transform, baker map, John Conway game, knight's tour,

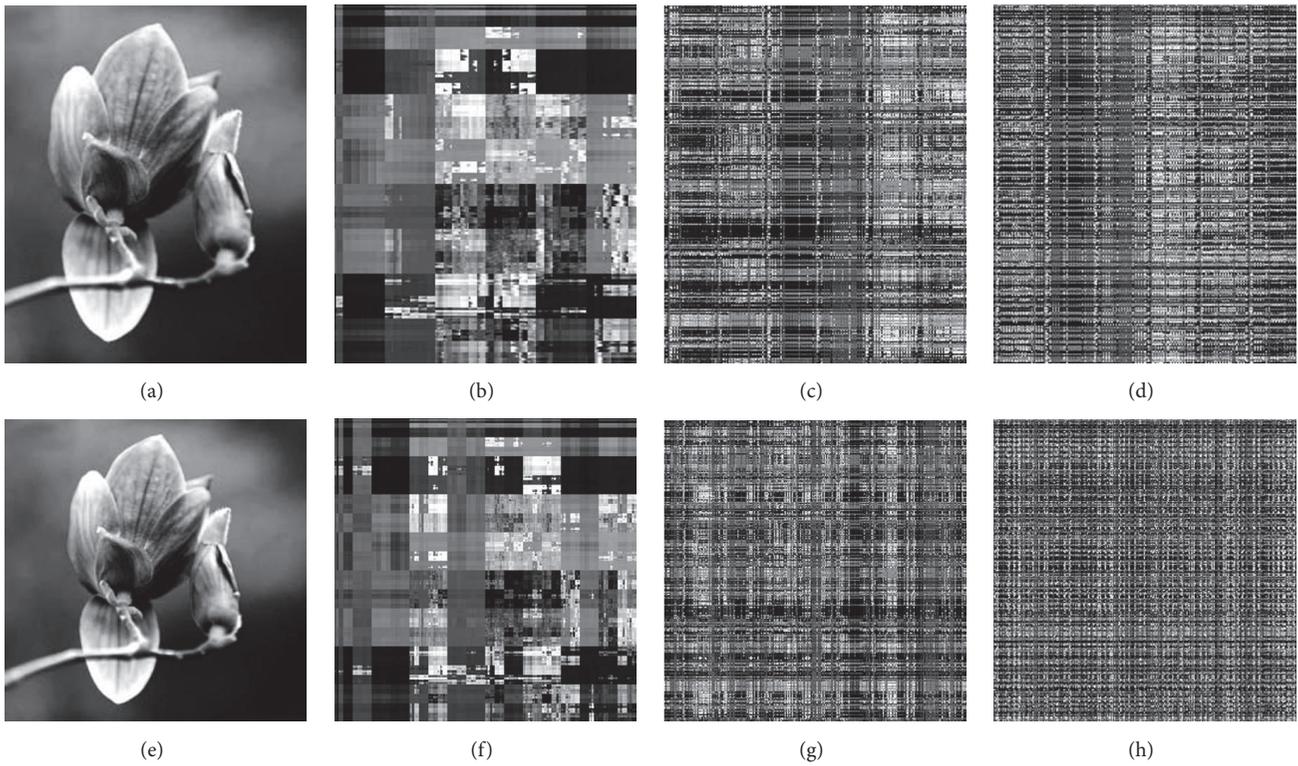


FIGURE 4: Image scrambling along x- and y-direction separately: (a) original image of size  $216 \times 256$ ; (e) original image of size  $256 \times 256$ ; (b), (c), and (d) scrambled images from (a) by once, 9 times and 25 times of LNBT, respectively; (f), (g), and (h) from (e) by once, 9 times, and 25 times of LNBT, respectively.

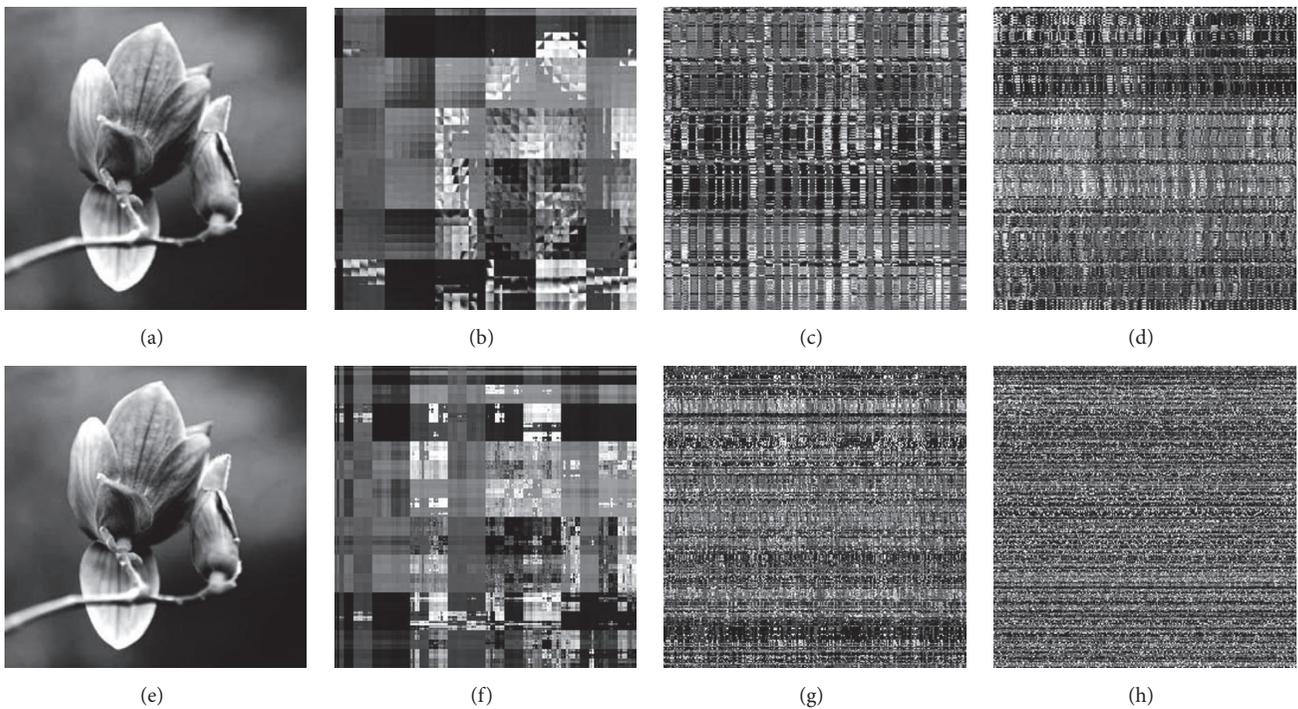


FIGURE 5: Image scrambling after scanned into a sequence line-by-line: (a) original image of  $216 \times 216$ ; (e) original image of  $256 \times 256$ ; (b), (c), and (d) scrambled images from (a) by once, 10 times, and 25 times of LNBT with parameter  $(6,6,1)$ , respectively; (f), (g), and (h) scrambled images of (e) by once, 10 times, and 25 times of LNBT with parameter  $(2,16,1)$ , respectively.

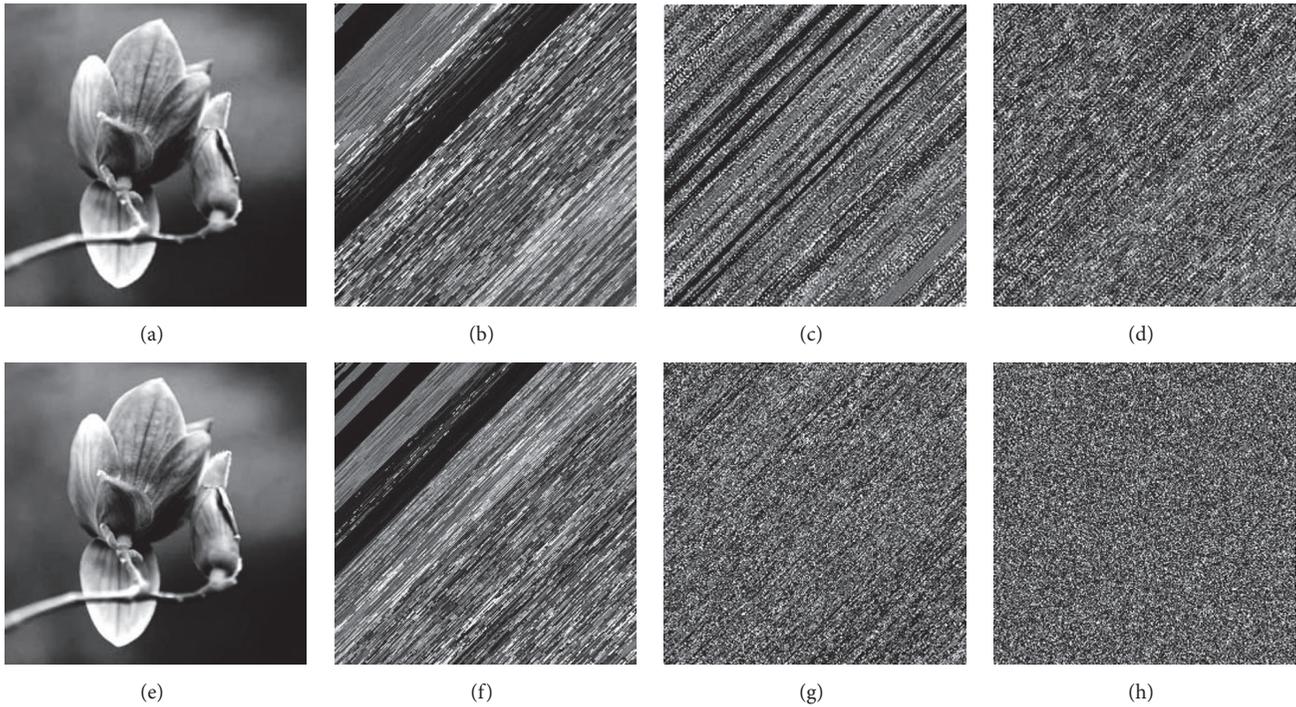


FIGURE 6: Image scrambling after scanned into a row in zigzag order: (a) original image of size  $216 \times 216$ ; (e) original image of size  $256 \times 256$ , (b), (c), and (d) scrambled images of (a) by 1-time, 10-time and 39-time of LNBT with parameter (6,6,1), respectively; (f), (g), and (h) scrambled images of (e) by 1 time, 10 times, and 39 times of LNBT with parameter (2,16,1), respectively.

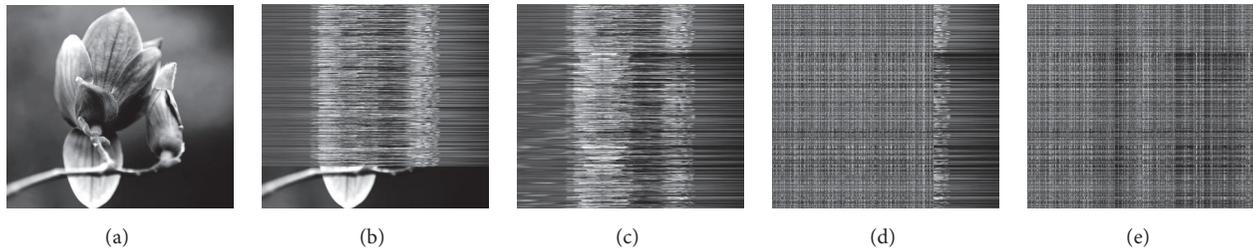


FIGURE 7: Scrambling image of irregular size by 20 times of LNBT with  $(2, 9, p)$ : (a) original image of size  $720 \times 648$ ; (b) permuting x-coordinate with  $p=1$ ; (c) permuting x-coordinate with  $p=209$ ; (d) permuting y-coordinate  $p=1$ ; (e) permuting y-coordinate with  $p=137$ .

magic square transform, and Peano-Hilbert curve, LNBT can be exploited to shuffle region with irregular shape of image, and specific operations are divided into two steps. First, the pixels of the region of image to be scrambled are scanned as one-dimensional sequence; second, use the above scrambling method for irregular size to shuffle the one-dimensional sequence. In Figure 8, we scan the pixels of image region into one-dimensional sequence and then permute these pixels of the sequence. Figure 8 is the results of three shapes by 19 times of LNBT.

**4.4. Scrambling on Bit Plane along  $x$ - and  $y$ -Direction.** Permuting the pixel position of an image can only destroy the correlation of adjacent pixels but cannot change the statistical properties of the image, such as histogram and information entropy. This is because scrambling transform based on pixels cannot change the gray value of pixels. If the binary bits of

the image pixels are scrambled, it can cause the pixel values to change and thereby can change the statistical properties of the image. If we use binary number of 8 bits to express the value of a pixel, then we can use 0 and 1 to represent a certain image. Therefore the size of a gray image of  $M \times N$  is  $8 \times M \times N$  and each pixel can be represented by binary block of size  $2 \times 4$  or  $4 \times 2$ ,  $1 \times 8$ ,  $8 \times 1$ . For example, if we use a binary block of  $2 \times 4$  to express a pixel of a gray image of size  $M \times N$ , we can get a binary matrix of size  $(2 \times M) \times (4 \times N)$ , and then use LNBT to permute the position of 0 and 1 in the binary matrix corresponding to the gray image. In Figure 9, we first convert original image to a binary matrix where the size of binary block of a pixel is  $2 \times 4$ ; next, apply LNBT to shuffle the binary matrix along  $x$ - and  $y$ -direction separately; finally, we convert the scrambled binary matrix to a scrambled image, i.e., a binary block of size  $2 \times 4$  is converted to a pixel. From Figure 9, we see that the scrambling transform based on bit

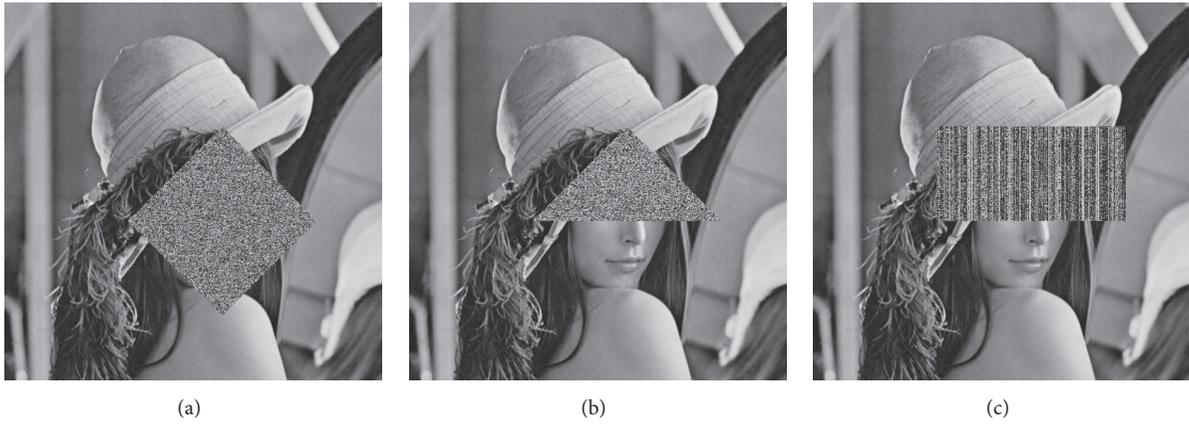


FIGURE 8: Scrambling the face region of Lena image, (a) diamond-shaped region, (b) triangular region, and (c) rectangular region.

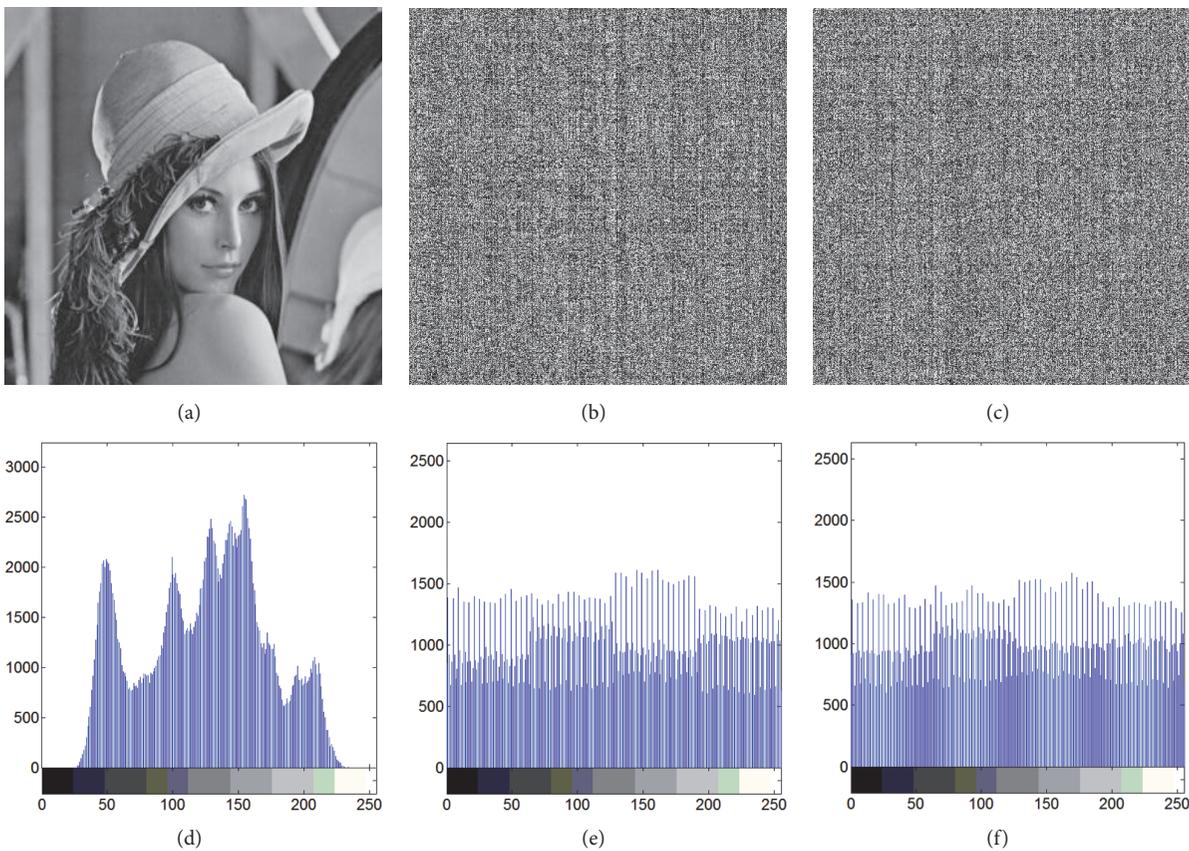


FIGURE 9: Image scrambling on bit plane by LNBT along x- and y-direction: (a) original image of 512×512; (b) scrambled image from (a) by 11 times of LNBT at x- and y-direction; (c) scrambled image from (a) by 15 times of LNBT at y-direction and 19 times at x-direction; (d) histogram of (a); (e) histogram of (b); (f) histogram of (c).

plane not only makes the scrambled image meaningless but also changes the statistical properties of the original image. Thus this shows that LNBT is also a tool of image encryption.

4.5. Digital Watermarking against Cropping Attack. In the robust watermarking of digital images, image scrambling is a powerful method for resisting cropping attack. LSB (least

significant bit) algorithm is a simple digital watermarking technology, but as a large amount of data hiding method, LSB has a significant position. Figure 10 is the experimental result of combining LNBT with LSB algorithm. The implementation process is as follows. First, use 19 times of LNBT to shuffle the carrier image, then embed the bits of the watermark image into the least significant bits of the scrambled carrier image, and finally, use the inverse transform of LNBT to obtain the

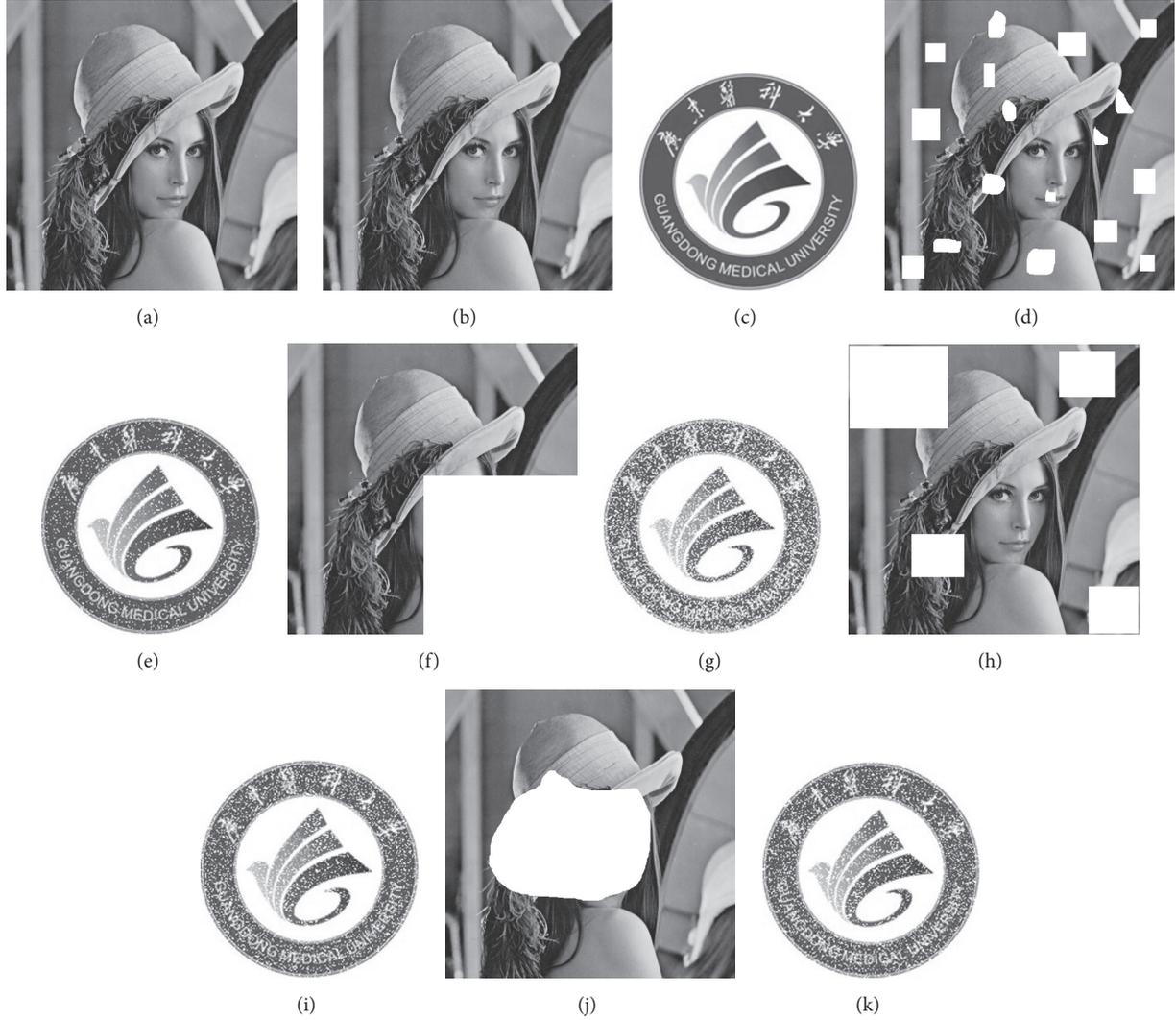


FIGURE 10: Watermarking experiment based on LNBT and LSB: (a) original image; (b) watermarked image; (c) watermark image, (d), (f), (h), and (j) attacked images by cropping; (e), (g), (i), and (k) watermarks extracted from the attacked watermarked images.

watermarked image (b). Images (e), (g), (i), and (k) are the extracted watermarks from the watermarked image after attacked by cropping, and Figure 10 shows that LNBT is effective against cropping attack.

## 5. Analysis and Conclusions

**5.1. Correlations between Adjacent Pixels.** The goal of image scrambling is to remove the correlation between the pixels of a plaintext image, and therefore the correlation value between adjacent pixels of image is an important index for evaluating the performance of the image scrambling algorithm. Assuming that  $x_n$  and  $y_n$  are two sequences consisted of horizontal or vertical or diagonally adjacent pixels, their correlation value can be calculated by [6]

$$Corr = \frac{|N \sum_{i=1}^N (x_i \times y_i) - \sum_{i=1}^N x_i \times \sum_{i=1}^N y_i|}{\sqrt{(N \sum_{i=1}^N x_i^2 - (\sum_{i=1}^N x_i)^2) \times (N \sum_{i=1}^N y_i^2 - (\sum_{i=1}^N y_i)^2)}} \quad (29)$$

where  $N$  is the length of sequences. Obviously, if the correlation value is close to 1, it indicates that there is a higher correlation between adjacent pixels. If the correlation value is close to 0, then the two adjacent pixel sequences have a lower correlation. Therefore, the smaller the correlation value is, the better the performance of the scrambling algorithm is.

To measure the performance of LNBT scrambling algorithm by using the correlation values of neighboring pixel sequences. First, we use other common scrambling algorithms and the proposed algorithm in this paper to perform a 32-round scrambling transformation on the Lena image.

TABLE 4: GDD and correlation values.

Method name	rounds	GDD	Correlation values		
			Horizontal	Vertical	Diagonal
Original image	\	\	0.9273	0.9590	0.9062
2D Arnold [10]	6	0.8995	0.1239	0.06369	0.0003
Discrete Baker map [26]	11	0.8972	0.0696	0.0384	0.0483
Sub-affine [27]	13	0.8985	0.0807	0.0921	0.0071
CA [15]	13	0.8908	0.0114	0.0270	0.0080
Fibonacci-Q [10]	32	0.7280	0.2968	0.9570	0.3012
Hilbert curve [19]	19	0.8910	0.0083	0.0013	0.0146
Magic cube [9]	27	0.8959	0.0670	0.0514	0.1107
2×2 sampling [28]	1*	0.8957	0.1225	0.0352	0.0770
LNBT-1	32*	0.8864	0.0749	0.1456	0.0431
LNBT-2	32	0.8884	0.0323	0.0170	0.0016
LNBT-3	24	0.8909	0.00005	0.0227	0.0057

\*Remark: visually, the scrambling performance of the third round of 2x2 sampling transform is best, and the 29th round of LNBT-1 has the best scrambling effect. Figures 12(i) and 12(j) are the scrambled images of the third round of 2x2 sampling transform and of the 29th round of LNBT-1, respectively.

Then pick out the largest value of GDD (see Section 5.2) in the 32-round transforms as the last results of each scrambling algorithm for comparison. Finally, we randomly select 5000 pairs of horizontally adjacent pixels, vertically adjacent pixels, and diagonally adjacent pixels and then calculate the correlation values of the neighboring pixel sequences of the three directions in the original image and the correlation values of the three directions in the scrambled images, respectively. Columns 4, 5, and 6 of Table 4 are correlation values of adjacent pixel sequences of the three directions in the original image and in the scrambled images, where LNBT-1 denotes the LNBT scrambling algorithm of permuting separately along x- and y-direction; LNBT-2 denotes the LNBT scrambling algorithm after scanned line-by-line; LNBT-3 denotes the LNBT scrambling algorithm after scanned in zigzag order. We can know from Table 4 that the LNBT scrambling transform can successfully remove the correlation between adjacent pixels. Figures 11(a), 11(b), and 11(c) display the gray-scale distributions of 5000 pairs of horizontally adjacent pixels, of vertically adjacent pixels, and of diagonally adjacent pixels, respectively, which are selected randomly from original image. (d), (e), and (f) are the gray-scale distributions of an adjacent pixel sequence pairs for horizontal, vertical and diagonal directions in the scrambled image using LNBT-1; (g), (h), and (i) show the gray distributions of an adjacent pixel sequence pairs for the three directions in the scrambled image using LNBT-2; (j), (k), and (l) are the gray-scale distributions using LNBT-3. The results of Table 4 and Figure 11 show that LNBT can successfully remove the correlation between adjacent pixels of plaintext image

**5.2. Scrambling Degree.** Unlike correlation values, scrambling degree can be used to evaluate the correlation of adjacent pixels of a whole scrambled image. If a scrambling algorithm can obtain a large scrambling degree, the algorithm

can achieve high confusion and diffusion properties. Image scrambling degree is determined by the gray difference between adjacent pixels [25]. Assuming that  $p(i, j)$  is the gray value of the pixel at position  $(i, j)$ , the gray difference of pixel  $p(i, j)$  can be calculated using [23–25]

$$GD(i, j) = \frac{1}{4} \sum_{s,t \in \{(i-1,j), (i+1,j), (i,j-1), (i,j+1)\}} (p(i, j) - p(s, t))^2, \quad (30)$$

where  $i = 2, 3, \dots, M-1$ ,  $j = 2, 3, \dots, N-1$ , and the image size is  $M \times N$ . We use  $E(GD)$  to represent the average gray-scale difference of whole image; obviously,  $E(GD)$  can be calculated using

$$E(GD) = \frac{1}{(M-2)(N-2)} \sum_{i=2}^{M-1} \sum_{j=2}^{N-1} GD(i, j). \quad (31)$$

Assume that  $E(GD)$  represents the average gray-scale difference of plaintext image and  $E'(GD)$  represents the average gray-scale of the scrambled image. Then, according to paper [25], image scrambling degree can be calculated as follows:

$$GDD = \frac{E'(GD) - E(GD)}{E'(GD) + E(GD)}. \quad (32)$$

From (32), it can be seen that the value of  $GDD$  is between -1 and 1, and the larger the value of  $GDD$  is, the better the performance of the scrambling algorithm is. The data in the third column of Table 4 is the maximum value in 32-round transform for each scrambling algorithm. The remark below Table 4 shows that GDD does not accurately evaluate the performance of a scrambling algorithm, and sometimes there is a large visual difference. Figure 12 is the scrambled images using scrambling algorithms to shuffle Lena image. The scrambling effect of Figures 12(e), 12(g), 12(k), and

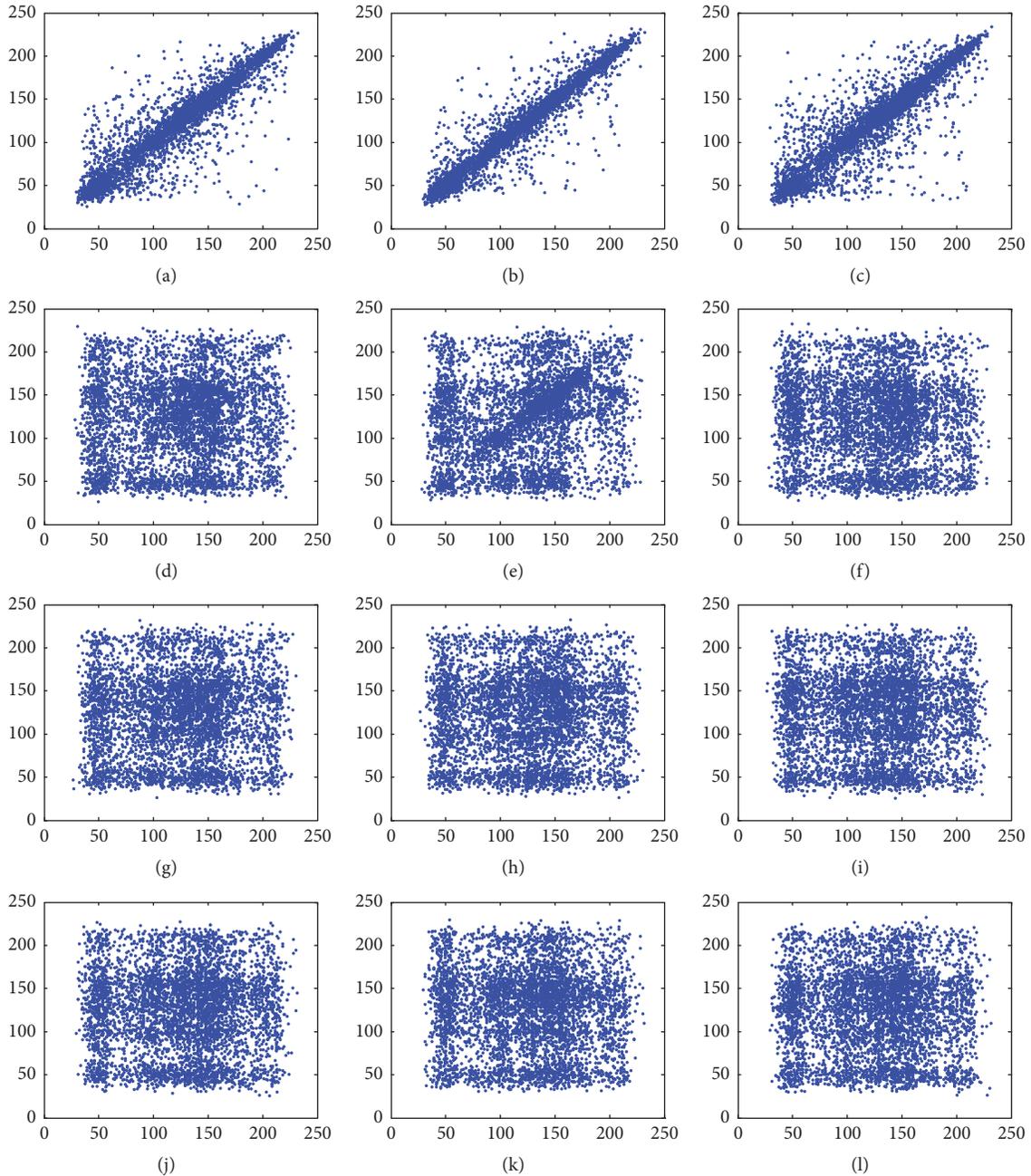


FIGURE 11: Gray-scale distributions of adjacent pixel sequence pairs: (a) horizontal of original image, (b) vertical of original image, (c) diagonal of original image, (d) horizontal of scrambled image using LNBT-1, (e) vertical of scrambled image using LNBT-1, (f) diagonal of scrambled image using LNBT-1, (g) horizontal of scrambled image using LNBT-2, (h) vertical of scrambled image using LNBT-2, (i) diagonal of scrambled image using LNBT-2, (j) horizontal of scrambled image using LNBT-3, (k) vertical of scrambled image using LNBT-3, and (l) diagonal of scrambled image using LNBT-3.

12(l) is obviously better. Therefore, it can be seen from Table 4 and Figure 12 that LNBT has good image scrambling performance.

**5.3. Conclusions.** This paper gives a new kind of scrambling transform, which is one-dimensional and is called LNBT with three parameters. LNBT is bijective, and thus LNBT is invertible and is a permutation on a finite subset of the natural numbers. Since LNBT has good scrambling performance, it

can be used to shuffle two-dimensional image; in fact, it can be directly used to scrambled one-dimensional audio signal and also used as a tool of encryption.

### Conflicts of Interest

These authors declare that there are no conflicts of interest regarding the publication of this paper.

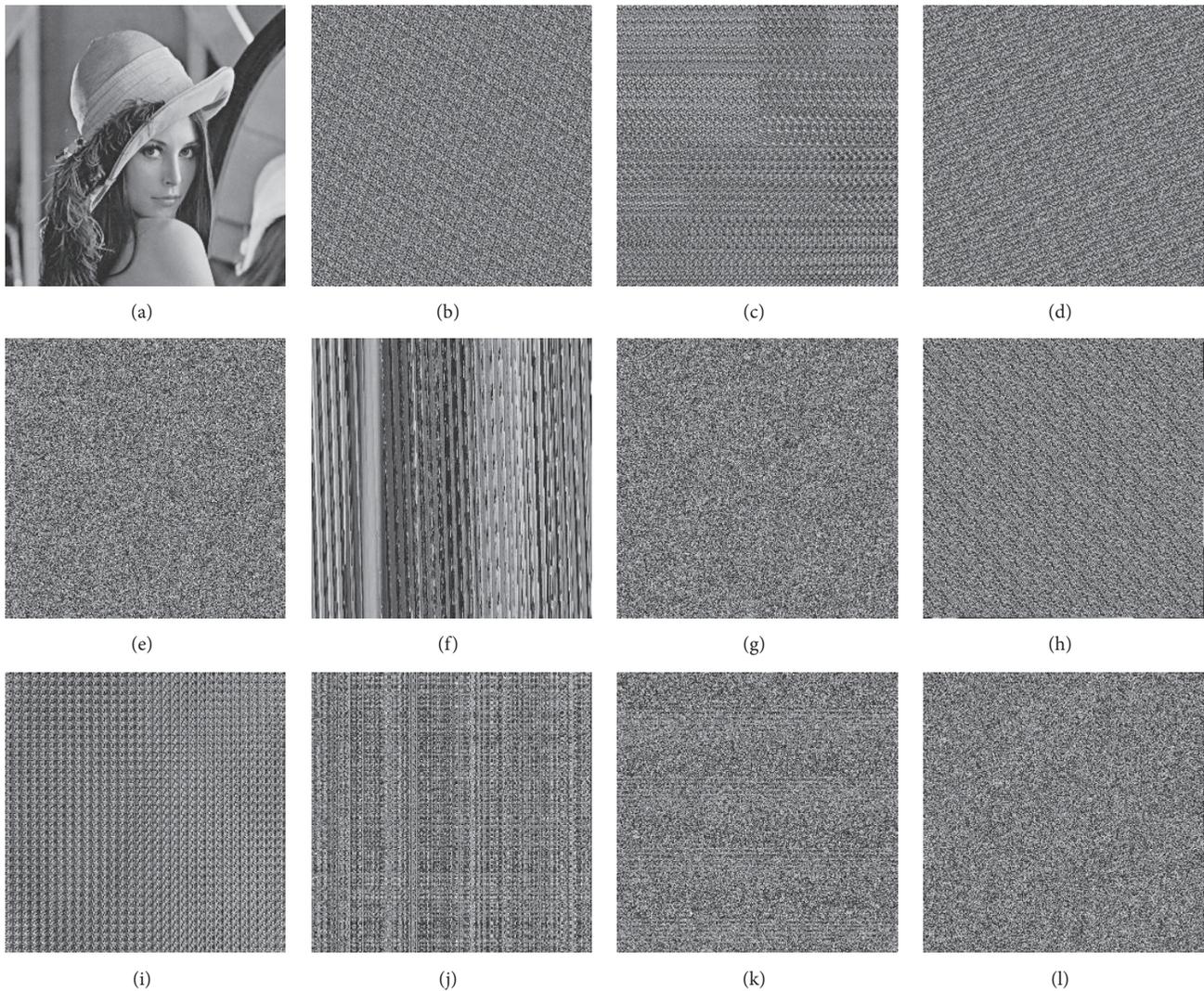


FIGURE 12: Scrambled images: (a) original image, (b) 6-round 2D Arnold transform, (c) 11-round discrete Baker map, (d) 13-round subaffine transform, (e) 13-round cellular automata transform, (f) 32-round Fibonacci transform, (g) 19-round Hilbert curve transform, (h) 27-round magic cube transform, (i) 3-round  $2 \times 2$  sampling, (j) 29-round LNBT-1, (k) 32 LNBT-2, and (l) 24-round LNBT-3.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant 61170320, the Natural Science Foundation of Guangdong Province of China under Grant S2011040002981, and the Science and Technology Development Fund of Macau under Grant 048/2016/A2 and Grant 0012/2018/A1.

## References

- [1] N. A. Abbas, "Image encryption based on independent component analysis and arnold's cat map," *Egyptian Informatics Journal*, vol. 17, no. 1, pp. 139–146, 2016.
- [2] G. Ye and K.-W. Wong, "An efficient chaotic image encryption algorithm based on a generalized Arnold map," *Nonlinear Dynamics*, vol. 69, no. 4, pp. 2079–2087, 2012.
- [3] R. Ye, "A novel chaos-based image encryption scheme with an efficient permutation-diffusion mechanism," *Optics Communications*, vol. 284, no. 22, pp. 5290–5298, 2011.
- [4] M. A. Gondal, Abdul Raheem, and I. Hussain, "A Scheme for Obtaining Secure S-Boxes Based on Chaotic Baker's Map," *3D Research*, vol. 5, no. 3, 2014.
- [5] I. Hussain, T. Shah, M. A. Gondal, and H. Mahmood, "Efficient method for designing chaotic S-boxes based on generalized Baker's map and TDERC chaotic sequence," *Nonlinear Dynamics*, vol. 74, no. 1-2, pp. 271–275, 2013.
- [6] L. Liu and S. Miao, "An image encryption algorithm based on Baker map with varying parameter," *Multimedia Tools & Applications*, pp. 1–17, 2016.
- [7] M. Carli, F. Battisti, M. Cancellaro, G. Boato, and A. Neri, "Joint watermarking and encryption of color images in the fibonacci-haar domain," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, 2009.

- [8] X. Ji, S. Bai, and G. Zhu, "Image encryption and compression based on the generalized knights tour, discrete cosine transform and chaotic maps," *Multimedia Tools & Applications*, pp. 1–15, 2016.
- [9] K. Muhammad, M. Sajjad, I. Mehmood, S. Rho, and S. W. Baik, "A novel magic LSB substitution method (M-LSB-SM) using multi-level encryption and achromatic component of an image," *Multimedia Tools and Applications*, vol. 75, no. 22, pp. 14867–14893, 2016.
- [10] D. Qi, J. Zou, and X. Han, "A new class of scrambling transformation and its application in the image information covering," *Science in China, Series E: Technological Sciences*, vol. 43, no. 3, pp. 304–312, 2000.
- [11] S. Roy and A. K. Pal, "A robust blind hybrid image watermarking scheme in RDWT-DCT domain using Arnold scrambling," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 3577–3616, 2017.
- [12] L. Sun, J. Xu, and S. Liu, "A robust image watermarking scheme using Arnold transform and BP neural network," *Neural Computing & Applications*, pp. 1–16, 2017.
- [13] J. Li, Q. Lin, C. Yu, X. Ren, and P. Li, "A QDCT- and SVD-based color image watermarking scheme using an optimized encrypted binary computer-generated hologram," *Soft Computing*, pp. 1–19, 2016.
- [14] M.-J. Tsai, "Wavelet tree based digital image watermarking by adopting the chaotic system for security enhancement," *Multimedia Tools and Applications*, vol. 52, no. 2-3, pp. 347–367, 2011.
- [15] T. D. Nguyen, S. Arch-int, and N. Arch-int, "A novel secure block data-hiding algorithm using cellular automata to enhance the performance of JPEG steganography," *Multimedia Tools and Applications*, vol. 74, no. 15, pp. 5661–5682, 2015.
- [16] B. L. Gunjal and S. N. Mali, "MEO based secured, robust, high capacity and perceptual quality image watermarking in DWT-SVD domain," *SpringerPlus*, vol. 4, no. 1, pp. 1–16, 2015.
- [17] J. J. Ranjani, "Data hiding using pseudo magic squares for embedding high payload in digital images," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 3715–3729, 2017.
- [18] L. Yang and K. Chen, "On the orders of transformation matrices (mod  $n$ ) and two types of generalized Arnold transformation matrices," *Science China Information Sciences*, vol. 47, no. 5, pp. 655–668, 2004.
- [19] N. Jiang, L. Wang, and W.-Y. Wu, "Quantum Hilbert Image Scrambling," *International Journal of Theoretical Physics*, vol. 53, no. 7, pp. 2463–2484, 2014.
- [20] A. V. Diaconu, A. Costea, and M.-A. Costea, "Color image scrambling technique based on transposition of pixels between RGB channels using Knight's moving rules and digital chaotic map," *Mathematical Problems in Engineering*, vol. 2014, Article ID 932875, 15 pages, 2014.
- [21] J. C. Zou, G. F. Li, and D. X. Qi, "Generalized Gray code and its application in the scrambling technology of digital images," *Applied Mathematics A Journal of Chinese Universities*, vol. 3, no. 3, pp. 363–370, 2002.
- [22] R.-G. Zhou, Y.-J. Sun, and P. Fan, "Quantum image Gray-code and bit-plane scrambling," *Quantum Information Processing*, vol. 14, no. 5, pp. 1717–1734, 2015.
- [23] A. L. Abu Dalhoum, A. Madain, and H. Hiary, "Digital image scrambling based on elementary cellular automata," *Multimedia Tools and Applications*, vol. 75, no. 24, pp. 17019–17034, 2016.
- [24] A. L. A. Dalhoum, B. A. Mahafzah, A. A. Awwad, I. Aldhamari, A. Ortega, and M. Alfonsoeca, "Digital image scrambling using 2D cellular automata," *IEEE MultiMedia*, vol. 19, no. 4, pp. 28–36, 2012.
- [25] R. Ye and H. Li, "A novel image scrambling and watermarking scheme based on cellular automata," in *Proceedings of the International Symposium on Electronic Commerce and Security, ISECS 2008*, pp. 938–941, chn, August 2008.
- [26] F. Zhao X, "Digital image scrambling based on the baker's transformation," *Journal of Northwest Normal University*, 2003.
- [27] S. Bai and Chongqing., "Property of sub-affine transformation and its application," *Journal of Computer Aided Design & Computer Graphics*, 2003.
- [28] W. Ding, W. Yan, and D. Qi, "Digital image scrambling," *Progress in Natural Science*, vol. 11, no. 6, pp. 454–460, 2001.
- [29] [https://en.wikipedia.org/wiki/Negative\\_base](https://en.wikipedia.org/wiki/Negative_base).
- [30] H. Cheng and X. Li, "Partial encryption of compressed images and videos," *IEEE Transactions on Signal Processing*, vol. 48, no. 8, pp. 2439–2451, 2000.
- [31] J.-L. Liu, "Efficient selective encryption for JPEG 2000 images using private initial table," *Pattern Recognition*, vol. 39, no. 8, pp. 1509–1517, 2006.
- [32] A. Moumen, M. Bouye, and H. Sissaoui, "New secure partial encryption method for medical images using graph coloring problem," *Nonlinear Dynamics*, vol. 82, no. 3, pp. 1475–1482, 2015.

## Research Article

# A Projection Neural Network for Circular Cone Programming

Yaling Zhang <sup>1,2</sup> and Hongwei Liu<sup>1</sup>

<sup>1</sup>School of Mathematics and Statistics, Xidian University, Xi'an 710071, China

<sup>2</sup>School of Computer Science, Xi'an Science and Technology University, Xi'an 710054, China

Correspondence should be addressed to Yaling Zhang; zyldella@126.com

Received 5 January 2018; Revised 24 April 2018; Accepted 3 May 2018; Published 10 June 2018

Academic Editor: Nibaldo Rodríguez

Copyright © 2018 Yaling Zhang and Hongwei Liu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A projection neural network method for circular cone programming is proposed. In the KKT condition for the circular cone programming, the complementary slack equation is transformed into an equivalent projection equation. The energy function is constructed by the distance function and the dynamic differential equation is given by the descent direction of the energy function. Since the projection on the circular cone is simple and costs less computation time, the proposed neural network requires less state variables and leads to low complexity. We prove that the proposed neural network is stable in the sense of Lyapunov and globally convergent. The simulation experiments show our method is efficient and effective.

## 1. Introduction

The circular cone is a pointed, closed, convex cone having hyperspherical sections orthogonal to its axis of revolution about which the cone is invariant to rotation [1–3]. Let its half-aperture angle be  $\theta_i \in (0, \pi/2)$ ,  $i = 1, 2, \dots, N$ , and then the  $n_i$ -dimensional circular cone denoted by  $L_{\theta_i}$  can be expressed as follows:

$$L_{\theta_i} = \left\{ x_i = \begin{bmatrix} x_{i1} \\ x_{i0} \end{bmatrix} \in R^{n_i-1} \times R : \|x_{i1}\| \leq \tan \theta_i x_{i0} \right\}, \quad (1)$$

where  $\|\cdot\|$  is the standard Euclidean norm.

When  $\theta_i = \pi/4$ , the circular cone reduces to the well-known second-order cone given by

$$K^{n_i} = \left\{ x_i = \begin{bmatrix} x_{i1} \\ x_{i0} \end{bmatrix} \in R^{n_i-1} \times R : \|x_{i1}\| \leq x_{i0} \right\}. \quad (2)$$

In this paper, we consider a linear circular cone programming (LCCP), which is described as follows:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b, \quad x \in L_{\theta}, \end{aligned} \quad (3)$$

where  $A \in R^{m \times n}$ ,  $c \in R^n$ ,  $b \in R^m$  and  $x = [x_1, \dots, x_N] \in R^{n_1} \times \dots \times R^{n_N}$ , and  $\sum_{i=1}^N n_i = n$ . Here,  $x = [x_1, \dots, x_N] \in R^{n_1} \times \dots \times R^{n_N}$  is viewed as a column vector in  $R^{n_1 + \dots + n_N}$ . In addition,

$$L_{\theta} = L_{\theta_1} \times L_{\theta_2} \times \dots \times L_{\theta_N}, \quad (4)$$

and  $x_i \in L_{\theta_i}$ . When  $\theta_i = \pi/4$ ,  $i = 1, 2, \dots, N$ , we have

$$K = K^{n_1} \times K^{n_2} \times \dots \times K^{n_N}. \quad (5)$$

The circular cone programming arises in some real-life engineering problems, such as the optimal grasping manipulation problems for multifingered robots [4, 5]. In addition, circular cone programming is applied in the perturbation analysis of second-order cone programming problems [6].

By the Lagrangian method, the dual programming to LCCP can be given as

$$\begin{aligned} \min \quad & b^T y \\ \text{s.t.} \quad & A^T y + z = c, \\ & z \in L_{\theta}^*, \end{aligned} \quad (6)$$

where  $y \in R^n$  is the Lagrange multipliers, and  $L_\theta^*$  is the dual cone of cone  $L_\theta$ , which is defined as

$$L_\theta^* = L_{\theta_1}^* \times L_{\theta_2}^* \times \cdots \times L_{\theta_N}^*, \quad (7)$$

where

$$L_{\theta_i}^* := \{z_i \mid z_i^T x_i \geq 0, x_i \in L_{\theta_i}\}. \quad (8)$$

Under mild constraint qualifications (e.g., Slater condition), strong duality holds for problem (3). Then, by the strong duality theorem for general conic programming problems [7], the KKT condition for (3) is given as

$$\begin{aligned} Ax &= b, \\ A^T y + z &= c, \\ x^T z &= 0, \quad x \in L_\theta, \quad z \in L_\theta^*. \end{aligned} \quad (9)$$

Circular cone programming is a nonlinear convex programming, and the second-order cone programming is a special case [8, 9]. There are many efficient methods to deal with the second-order cone programming [8, 10, 11]. However, different from the second-order cone, the circular cone  $L_\theta$  is nonsymmetric. Some properties holding in the second-order cone can be extended to the circular cone [12]. However, some other second-order cone properties fail to be satisfied for the circular cone [3, 12].

Recently, an interior point method [13] is proposed for the circular cone programming based on self-concordant barrier functions for its cones. However, in many science and engineering applications, real-time solutions of circular cone problems are often desired. For the large-scale problems, the traditional algorithm may not be efficient due to the complexity of the algorithm used. The artificial neural network based circuit implementation is an efficient approach for the real-time problem. At present, two types of neural networks are developed for the second-order cone programming and have shown some computational advantages. One type is the smooth neural network, including the merit function method derived from the Fischer-Burmeister function [14], the smoothed natural residual merit function method [15], and the merit function method based on the generalized Fischer-Burmeister function [16]. The other is the projection neural network by replacing the scalar projection function with the cone projection function [14, 17]. In paper [18], professor He proposed a neural network based on the simple projection and contraction technique for linear asymmetric variational inequalities. Inspired by the ideal projection and contraction technique and the new results about the projection conclusions on the circular cone [2], we can develop the projection neural network for the linear circular cone programming.

In this paper, we focus on neural network approach to the circular cone programming problem. The energy function is constructed by the distance function based on a cone projection function, whose solutions correspond to the KKT points of the circular cone programming. The dynamic differential equation is given by the descent direction of the

energy function based on the projection and contraction technique. The proposed neural network requires less state variables and leads to low complexity. Its Lyapunov stability and global convergence are proved under some conditions. Finally, we test the projection neural network by some numerical examples and the optimal grasping manipulation problems for multifingered robots and also compare the neural network with the second neural network for some second-order cone programming problems in paper [14]. Simulation results demonstrate the effectiveness of the proposed neural network.

## 2. Preliminaries

In this section, firstly, we introduced some concepts about the Lyapunov stability of the first-order differential equation [16, 19]. Then, we briefly introduce some properties of circular cone and the projection on the circular cone, which are proposed in paper [2].

*2.1. Lyapunov Stability of the First-Order Differential Equation.* Given a mapping  $f : R^n \rightarrow R^n$ , the following first-order differential equation is

$$\begin{aligned} \frac{du}{dt} &= f(u(t)), \\ u(t_0) &= u_0 \in R^n. \end{aligned} \quad (10)$$

Next, we give the definition of Lyapunov stability [19].

*Definition 1* ([19]).

- (1) For (10), a point  $u^* \in R^n$  is called an equilibrium point of (10) if  $f(u^*) = 0$ .
- (2) If there is a neighborhood  $N \subseteq R^n$  of  $u^*$  such that  $f(u^*) = 0$  and  $f(u) \neq 0$  for  $u \in N$ , then  $u^*$  is called an isolated equilibrium point.

*Definition 2* ([19]). Let  $u(t)$  be a solution of (10). For an isolated equilibrium point  $u^* \in R^n$ , if, for any  $u(t_0) = u_0$  and  $\epsilon > 0$ , there exists a  $\delta > 0$  such that

$$\|u_0 - u^*\| < \delta \implies \|u(t) - u^*\| < \epsilon \quad \text{for } t \geq t_0, \quad (11)$$

then  $u^*$  is Lyapunov stable.

*Definition 3* (Lyapunov function [19]). Let  $N \subseteq R^n$  be an open neighborhood of  $\tilde{u}$ . A continuously differential function  $g : R^n \rightarrow R$  is said to be a Lyapunov function (or energy function) at the state  $\tilde{u}$  for (10) if

$$\begin{aligned} g(\tilde{u}) &= 0, \\ g(\tilde{u}) &> 0, \quad \forall u \in N \setminus \tilde{u}; \\ \frac{dg(u(t))}{dt} &\leq 0, \quad \forall u \in N. \end{aligned} \quad (12)$$

The relationship between stabilities and a Lyapunov function is given as follows, which is proposed in paper [20].

**Lemma 4** ([20]). *An isolated equilibrium point  $u^*$  is Lyapunov stable if there exists a Lyapunov function over some neighborhood  $N$  of  $u^*$ .*

2.2. *The Properties and Projection on the Circular Cone.* Let

$$T_i = \begin{bmatrix} I_{n_i-1} & 0 \\ 0 & \tan \theta_i \end{bmatrix}, \quad (13)$$

for  $i = 1, 2, \dots, N$ , where  $I_{n_i-1}$  is the  $n_i - 1$  dimension unit matrix. The following lemma describes the relationship between the second-order cone  $K^{n_i}$  and the circular cone  $L_{\theta_i}$ , where  $i = 1, 2, \dots, N$ .

**Lemma 5** ([2]). *Let  $L_{\theta_i}$  and  $K^{n_i}$  be defined as in (1) and (2) for  $i = 1, 2, \dots, N$ . Then we obtain*

- (1)  $L_{\theta_i} = T_i^{-1}K^{n_i}$  and  $K^{n_i} = T_i L_{\theta_i}$ ,
- (2)  $T_i K^{n_i} = L_{\pi/2-\theta_i}$  and  $L_{\pi/2-\theta_i} = T_i^2 L_{\theta_i}$ ,
- (3)  $L_{\theta_i}^* = L_{\pi/2-\theta_i}$  and  $(L_{\theta_i}^*)^* = L_{\theta_i}$ .

From Lemma 5 and the definition of  $L_{\theta_i}$ ,  $L_{\theta_i}^*$ ,  $i = 1, 2, \dots, N$ , we have

$$x_i \in L_{\theta_i} \iff T_i x_i \in K^{n_i},$$

$$\text{for } x_i = \begin{bmatrix} x_{i1} \\ x_{i0} \end{bmatrix} \in R^{n_i-1} \times R \quad (14)$$

and

$$z_i \in L_{\theta_i}^* \iff T_i^{-1} z_i \in K^{n_i},$$

$$\text{for } z_i = \begin{bmatrix} z_{i1} \\ z_{i0} \end{bmatrix} \in R^{n_i-1} \times R. \quad (15)$$

Let

$$T = \begin{bmatrix} T_1 & & 0 \\ & \ddots & \\ 0 & & T_N \end{bmatrix}. \quad (16)$$

Then, from the conclusion above, we have

$$x \in L_{\theta} \iff Tx \in K, \quad (17)$$

for  $x = [x_1, \dots, x_N] \in R^{n_1} \times \dots \times R^{n_N}$

and

$$x \in L_{\theta}^* \iff T^{-1}x \in K, \quad (18)$$

for  $z = [z_1, \dots, z_N] \in R^{n_1} \times \dots \times R^{n_N}$ .

Let  $x_i = \begin{bmatrix} x_{i1} \\ x_{i0} \end{bmatrix} \in R^{n_i-1} \times R$ ,  $i = 1, 2, \dots, N$ . Then the spectral decomposition of  $x_i$  on circular cone  $L_{\theta_i}$  can be given as [2]

$$x_i = \lambda_1(x_i) c_1(x_i) + \lambda_2(x_i) c_2(x_i), \quad (19)$$

where

$$\lambda_1(x_i) = x_{i0} - \|x_{i1}\| \cot \theta_i, \quad (20)$$

$$\lambda_2(x_i) = x_{i0} + \|x_{i1}\| \tan \theta_i$$

and

$$c_1(x_i) = \frac{1}{1 + \cot^2 \theta_i} \begin{bmatrix} \cot \theta_i I_{n_i-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -w \\ 1 \end{bmatrix}, \quad (21)$$

$$c_2(x_i) = \frac{1}{1 + \tan^2 \theta_i} \begin{bmatrix} \tan \theta_i I_{n_i-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w \\ 1 \end{bmatrix}$$

with  $w = -x_{i1} / \|x_{i1}\|_2$  if  $x_{i1} \neq 0$  and any vector in  $R^{n_i-1}$  satisfying  $\|w\| = 1$  if  $x_{i1} = 0$ .

Next we introduce the projection on the circular cone [2].

**Lemma 6** ([2]). *For any  $x_i = \begin{bmatrix} x_{i1} \\ x_{i0} \end{bmatrix} \in R^{n_i-1} \times R$ ,  $i = 1, 2, \dots, N$ , let  $P_{L_{\theta_i}}(x_i)$  be the projection of  $x_i$  on the circular cone  $L_{\theta_i}$ . Then we obtain*

$$P_{L_{\theta_i}}(x_i) = (\lambda_1(x_i))_+ c_1(x_i) + (\lambda_2(x_i))_+ c_2(x_i), \quad (22)$$

where  $s_+ := \max(0, s)$  for  $s \in R$ .

Let  $x = [x_1, \dots, x_N] \in R^{n_1} \times \dots \times R^{n_N}$ . Then the projection  $P_{L_{\theta}}(x)$  of  $x$  on cone  $L_{\theta}$  is described as

$$P_{L_{\theta}}(x) = [P_{L_{\theta_1}}(x_1), \dots, P_{L_{\theta_N}}(x_N)] \in R^{n_1} \times \dots \times R^{n_N}. \quad (23)$$

In particular, when  $\theta_i = \pi/4$ , the projection on second-order cone  $K^{n_i}$  can be obtained, which have been proposed in papers [21–23]:

$$P_{K^{n_i}}(x_i) = (x_{i0} - \|x_{i1}\|)_+ \frac{1}{2} \begin{bmatrix} -w \\ 1 \end{bmatrix} + (x_{i0} + \|x_{i1}\|)_+ \frac{1}{2} \begin{bmatrix} w \\ 1 \end{bmatrix}, \quad (24)$$

where  $w = -x_{i1} / \|x_{i1}\|$  when  $x_{i1} \neq 0$  and any vector in  $R^{n_i-1}$  satisfying  $\|w\| = 1$  when  $x_{i1} = 0$ . Then the projection  $P_K(x)$  of  $x$  on the cone  $K$  is described as

$$P_K(x) = [P_{K^{n_1}}(x_1), \dots, P_{K^{n_N}}(x_N)] \in R^{n_1} \times \dots \times R^{n_N}. \quad (25)$$

It is well known that any  $z \in R^n$  can be written as

$$z = P_{\Theta}(z) + P_{-\Theta^*}(z) \quad (26)$$

where  $\Theta$  denotes any closed convex cone and  $\Theta^*$  represents the dual cone of  $\Theta$ . Hence, when  $\Theta = L_{\theta}$  and  $\Theta = K$ , respectively, we have

$$z = P_{L_{\theta}}(z) + P_{-L_{\theta}^*}(z) = P_{L_{\theta}}(z) - P_{-L_{\theta}^*}(-z) \quad (27)$$

and

$$z = P_K(z) + P_{-K}(z) = P_K(z) - P_{(K)^*}(-z). \quad (28)$$

Since the second-order cone is self-dual, then  $K = K^*$ . We have

$$z = P_K(z) + P_{-K}(z) = P_K(z) - P_{-K}(-z). \quad (29)$$

### 3. A Projection-Based Neural Network Model for Circular Cone Programming

In this section, we build a projection-based neural network model for circular cone programming.

Firstly, an equivalent projection equation is built for the KKT condition (9). Since the circular cone is a non-self-dual cone, our analysis is based on the relationship of circular cone and second-order cone.

**Lemma 7.** Assume  $\bar{x} \in L_\theta$ ,  $\bar{z} \in L_\theta^*$ . Then  $\bar{x}^T \bar{z} = 0$  if and only if  $e(\bar{x}, \bar{z}) = 0$ , where  $e(\bar{x}, \bar{z}) = T\bar{x} - P_K(T\bar{x} - T^{-1}\bar{z})$ .

*Proof.* “ $\Rightarrow$ ” The projection on the closed convex set  $K$  has an important property [24],

$$\langle x - P_K(x), T\bar{x} - P_K(x) \rangle \leq 0, \quad \forall x \in R^n \quad (30)$$

where  $\langle \cdot \rangle$  denotes the inner product of two vectors.

Let  $x = T\bar{x} - T^{-1}\bar{z}$  in the inequality above. Then we have

$$0 \leq \|e(\bar{x}, \bar{z})\|^2 \leq \langle e(\bar{x}, \bar{z}), Tz \rangle. \quad (31)$$

Moreover, because

$$\begin{aligned} \langle e(\bar{x}, \bar{z}), T^{-1}\bar{z} \rangle &= \langle T\bar{x} - P_K(T\bar{x} - T^{-1}\bar{z}), T^{-1}\bar{z} \rangle \\ &= \langle \bar{x}, \bar{z} \rangle - \langle P_K(T\bar{x} - T^{-1}\bar{z}), T^{-1}\bar{z} \rangle \end{aligned} \quad (32)$$

and

$$\begin{aligned} \langle \bar{x}, \bar{z} \rangle &= 0, \\ \langle P_K(T\bar{x} - T^{-1}\bar{z}), \bar{z} \rangle &\geq 0, \end{aligned} \quad (33)$$

we have

$$\langle e(\bar{x}, \bar{z}), T\bar{z} \rangle \leq 0. \quad (34)$$

By (31) and (34),  $e(\bar{x}, \bar{z}) = 0$  is obtained.

“ $\Leftarrow$ ” It follows from  $e(\bar{x}, \bar{z}) = 0$  that

$$T\bar{x} = P_K(T\bar{x} - T^{-1}\bar{z}) \in K. \quad (35)$$

From (17), we have  $\bar{x} \in L_\theta$ .

Based on (29), we have

$$T\bar{x} - T^{-1}\bar{z} = P_K(T\bar{x} - T^{-1}\bar{z}) + P_{-K}(T\bar{x} - T^{-1}\bar{z}). \quad (36)$$

Then, we get

$$\begin{aligned} T^{-1}\bar{z} &= -P_{-K}(T\bar{x} - T^{-1}\bar{z}) = P_K(-(T\bar{x} - T^{-1}\bar{z})) \\ &\in K. \end{aligned} \quad (37)$$

From (18), it is easy to obtain  $\bar{z} \in L_\theta^*$ . For any  $Tx \in K$ , we have

$$\begin{aligned} \langle T\bar{x} - T^{-1}\bar{z} - P_K(T\bar{x} - T^{-1}\bar{z}), Tx \\ - P_K(T\bar{x} - T^{-1}\bar{z}) \rangle \leq 0. \end{aligned} \quad (38)$$

Since  $T\bar{x} = P_K(T\bar{x} - T^{-1}\bar{z})$ , we get

$$\langle Tx - T\bar{x}, T^{-1}\bar{z} \rangle \geq 0. \quad (39)$$

Substituting  $Tx = 0 \in K$  and  $Tx = 2T\bar{x} \in K$  in inequality (39), we have

$$\langle \bar{x}, \bar{z} \rangle = \bar{x}^T \bar{z} = 0. \quad (40)$$

The proof is completed.  $\square$

Let

$$e(u) := Wu - P_\Omega(Wu - W^{-1}(Mu + q)), \quad (41)$$

where  $P_\Omega(\cdot)$  denotes the projection on the set  $\Omega = K \times R^m$  and

$$\begin{aligned} u &= \begin{bmatrix} x \\ y \end{bmatrix}, \\ W &= \begin{bmatrix} T & 0 \\ 0 & I_m \end{bmatrix}, \\ M &= \begin{bmatrix} 0 & -A^T \\ A & 0 \end{bmatrix}, \\ q &= \begin{bmatrix} c \\ -b \end{bmatrix}. \end{aligned} \quad (42)$$

Obviously,  $M$  is a block asymmetric matrix; i.e.,  $M^T = -M$ . So, we have

$$e(u) = \begin{bmatrix} Tx - P_K(Tx - T^{-1}(c - A^T y)) \\ Ax - b \end{bmatrix}. \quad (43)$$

From Lemma 7, we know that the KKT condition (9) is equivalent to

$$e(u) = 0. \quad (44)$$

Let  $\Omega_\theta = L_\theta \times R^m$ . Then it is easy to prove that

$$u \in \Omega_\theta \iff Wu \in \Omega. \quad (45)$$

Next, we give the following neural network model for circular cone programming, which consists of the following energy function and dynamical system.

The energy function is given as follows:

$$H(u) = \frac{1}{2} \|u - u^*\|^2, \quad (46)$$

where  $u^* \in \Omega_\theta^*$ .

Here, a dynamical system is proposed to solve (44). The dynamical system is given as follows:

$$\frac{du}{dt} = F(u) = -r\rho(u)(W - W^{-1}M)e(u), \quad (47)$$

where  $r \in \{0, 1\}$  and

$$\rho(u) = \frac{\|e(u)\|^2}{\|(W - W^{-1}M)e(u)\|^2}. \quad (48)$$

The system described by (47) can be realized by a recurrent neural network with two-layer structure.

#### 4. Stability Analysis

In this section, the stability analysis of projection neural network for circular cone programming is given.

For the dynamical system (47), we have the following result.

**Lemma 8.**  $u^* \in \Omega_\theta^*$  if and only if  $u^*$  is an equilibrium point of network (47), where  $\Omega_\theta^*$  is the solution set of problem (9).

*Proof.* “ $\Rightarrow$ ” Since  $u^* \in \Omega_\theta^*$ , we have  $e(u^*) = 0$ . It is easy to know that

$$\left. \frac{du}{dt} \right|_{u=u^*} = 0, \quad (49)$$

so  $u^*$  is an equilibrium point of network (47).

“ $\Leftarrow$ ” Since  $u^*$  is an equilibrium point of network (47), we obtain  $F(u^*) = -r\rho(u)(W - W^{-1}M)e(u^*) = 0$ .

Consider

$$\begin{aligned} \|F(u^*)\|_2^2 &= r^2\rho(u^*)^2 [e(u^*)]^T (W - W^{-1}M)^T \\ &\cdot (W - W^{-1}M)e(u^*) = 0. \end{aligned} \quad (50)$$

Since  $(W - W^{-1}M)^T(W - W^{-1}M)$  is positive definite matrix,  $e(u^*) = 0$  is obtained. From (45), we know  $u^* \in \Omega_\theta^*$ .  $\square$

The following result guarantees the existence and uniqueness for the solution  $u(t)$  of neural network (47).

**Lemma 9.** For any  $u_0 \in \Omega$ , there exists a unique continuous solution  $u(t)$  of (47) with  $u(0) = u_0$  for all  $t \geq 0$ .

*Proof.* Because

$$e(u) = \begin{bmatrix} Tx - P_K(Tx - T^{-1}(c - A^T y)) \\ Ax - b \end{bmatrix}, \quad (51)$$

from the results in paper [11], we know that  $e(u)$  is semismooth. From all of the above, we conclude that  $F(u)$  is semismooth. Thus, there exists a unique solution  $u(t)$  for neural network (47).  $\square$

The results of Lemmas 8 and 9 indicate that neural network model (47) is well defined. Now, we give the Lyapunov stability of neural network (47).

**Theorem 10.** The solution of neural network (47), with initial point  $u_0$ , is Lyapunov stable.

*Proof.* From Lemma 9, there exists a unique continuous solution  $u(t)$  of (47) with  $u(0) = u_0$  for all  $t \geq 0$ . Since  $u^* \in \Omega_\theta^*$ , we have

$$\langle u^*, Mu^* + q \rangle = \langle Wu^*, W^{-1}(Mu^* + q) \rangle = 0. \quad (52)$$

In addition, for any  $v \in \Omega_\theta$ , we have

$$\langle v, Mu^* + q \rangle = \langle Wv, W^{-1}(Mu^* + q) \rangle \geq 0. \quad (53)$$

From (52) and (53), we have

$$\langle W(v - u^*), W^{-1}(Mu^* + q) \rangle \geq 0. \quad (54)$$

Setting  $Wv = P_\Omega(Wu - W^{-1}(Mu + q))$  in (54), it follows that

$$\begin{aligned} \langle W^{-1}(Mu^* + q), P_\Omega(Wu - W^{-1}(Mu + q)) \\ - Wu^* \rangle \geq 0 \end{aligned} \quad (55)$$

for any  $u \in R^{n+m}$ . For any  $v \in R^{n+m}$  and  $w \in \Omega_\theta$ , we have

$$\langle v - P_\Omega(v), P_\Omega(v) - Ww \rangle \geq 0. \quad (56)$$

Let  $v = Wu - W^{-1}(Mu + q)$ ,  $w = u^*$  in (30). We obtain

$$\begin{aligned} \langle e(u) - W^{-1}(Mu + q), P_\Omega(Wu - W^{-1}(Mu + q)) \\ - Wu^* \rangle \geq 0. \end{aligned} \quad (57)$$

Based on (55) and (57), we get

$$\langle e(u) - W^{-1}M(u - u^*), W(u - u^*) - e(u) \rangle \geq 0. \quad (58)$$

It follows that

$$\begin{aligned} \langle u - u^*, (W - W^{-1}M)e(u) \rangle \\ \geq \|e(u)\|^2 + \langle u - u^*, M(u - u^*) \rangle. \end{aligned} \quad (59)$$

Because  $M^T = -M$ , we have  $\langle u - u^*, M(u - u^*) \rangle = 0$ . Thus we get

$$\langle u - u^*, (W - W^{-1}M)e(u) \rangle \geq \|e(u)\|^2. \quad (60)$$

By the equation above, we obtain

$$\begin{aligned} \frac{d}{dt}H(u(t)) &= (u - u^*)^T \frac{du}{dt} \\ &= -r\rho(u)(u - u^*)^T (W - W^{-1}M)e(u) \\ &\leq -r\rho(u)\|e(u)\|^2 \leq 0, \end{aligned} \quad (61)$$

which shows that the solution of neural network (47) is Lyapunov stable.  $\square$

Next, we give the globally convergent result of the trajectory of neural network (47).

**Theorem 11.** *Let  $u^*$  is the equilibrium point of neural network (47). The solution trajectory of neural network (47) with initial point  $u_0$  is globally convergent to  $u^*$  and has finite convergence time.*

*Proof.* From (61), we know that the level set

$$L(u^0) := \{u \mid H(u) \leq H(u^0)\} \quad (62)$$

is bounded. Then, based on the Invariant Set Theorem [25], we have that the solution trajectory  $u(t)$  converges to  $\bar{u}$  as  $t \rightarrow +\infty$ , where  $\bar{u}$  is the largest invariant set in

$$\Lambda = \left\{ u \in L(u^0) \mid \frac{dH(u(t))}{dt} = 0 \right\}. \quad (63)$$

□

Now, we prove the result that  $(du/dt)|_{u=\bar{u}} = 0$  if and only if  $(dH(u(t))/dt)|_{u=\bar{u}} = 0$ , and then we can obtain the globe convergence of neural network (47).

If  $(du/dt)|_{u=\bar{u}} = 0$ , then  $(dH(u(t))/dt)|_{u=\bar{u}} = (\bar{u} - u^*)^T (du/dt)|_{u=\bar{u}} = 0$ .

If  $(dH(u(t))/dt)|_{u=\bar{u}} = 0$ , from (61), we have

$$\begin{aligned} 0 &= \frac{dH(u(t))}{dt} \Big|_{u=\bar{u}} = (\bar{u} - u^*)^T \frac{du}{dt} \Big|_{u=\bar{u}} \\ &\leq -r\rho(\bar{u}) \|e(u)\|^2 \leq 0. \end{aligned} \quad (64)$$

So  $e(\bar{u}) = 0$ . Then

$$\frac{du}{dt} \Big|_{u=\bar{u}} = -r\rho(\bar{u}) (W - W^{-1}M) e(\bar{u}) = 0. \quad (65)$$

From the conclusions above,  $u(t)$  converges globally to the equilibrium point  $u^*$ . Moreover, from Lemma 9 and the same argument as in [14, 26], the neural network (47) has finite convergence time.

## 5. Simulation Experiments

In this section, we give some examples to test the simulation performance of neural network (47). The neural network is run in the MATLAB 7.0 environment on an Intel Core processor 1.80GHZ personal computer with 4.0GB of RAM. The numerical examples are solved by ODE23 in the ode solver, which is a nonstiff medium order method. In the ode solver,  $r = 0.8$ ,  $RelTol = 10^{-6}$ , and  $AbsTol = 10^{-9}$  for the test examples.

*Example 12.* Consider the following problem with two 3-dimensional circular cones:

$$\begin{aligned} \min \quad & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\ \text{s.t.} \quad & x_1 + 2x_2 + x_6 = 9 \end{aligned}$$

$$x_1 + x_4 + 4x_5 = 20$$

$$x_2 + x_3 + x_5 = 6$$

$$x_1 + x_2 = 4$$

$$x_3 + 2x_5 = 8$$

$$\|(x_2, x_3)^T\| \leq \tan \theta_1 x_1$$

$$\|(x_5, x_6)^T\| \leq \tan \theta_2 x_4, \quad (66)$$

where

$$A = \begin{pmatrix} 1 & 2 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 4 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 & 0 \end{pmatrix},$$

$$b = \begin{pmatrix} 9 \\ 20 \\ 6 \\ 4 \\ 8 \end{pmatrix}, \quad (67)$$

$$c = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix},$$

$$\theta_1 = \frac{\pi}{3},$$

$$\theta_2 = \frac{\pi}{3}.$$

The optimal solution of Example 12 is

$$x^* = (2.288, 1.712, 0.575, 2.863, 3.712, 3.288)^T \quad (68)$$

and

$$y^* = (1.628, -0.640, -2.267, 0.012, 3.267)^T. \quad (69)$$

Figures 1 and 2 depict the trajectories of neural network (47) with the initial values  $x_0 = (0, 0, 0, 0, 0, 0)^T$  and  $y_0 = (0, 0, 0, 0, 0)^T$  converging to its solutions  $x^*$  and  $y^*$ , respectively.

*Example 13.* Consider the following problem with three 3-dimensional circular cones:

$$\begin{aligned} \min \quad & x_2 + 2x_3 + 3x_4 + x_5 + 2x_6 + 2x_7 + x_8 + x_9 \\ \text{s.t.} \quad & x_1 + x_2 + 2x_3 + 2x_5 + x_7 = 15 \end{aligned}$$

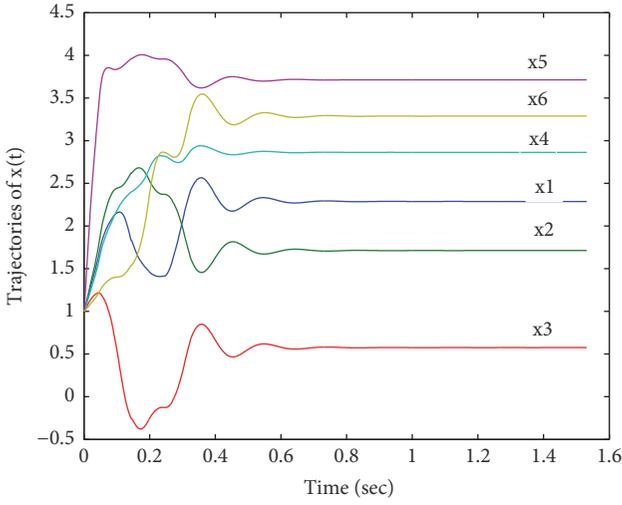


FIGURE 1: The  $x$  trajectories followed from Example 12.

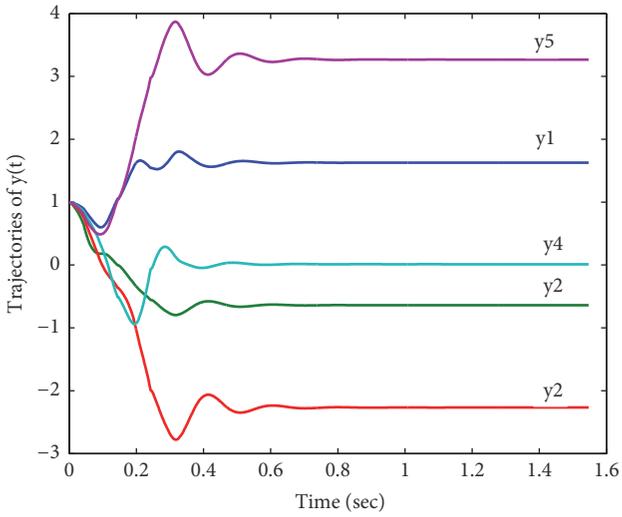


FIGURE 2: The  $y$  trajectories followed from Example 12.

$$\begin{aligned}
 4x_1 + 6x_2 + 3x_3 + x_5 + 2x_7 + x_8 &= 34 \\
 x_3 + x_4 + x_5 + x_6 + 3x_7 + x_8 + x_9 &= 24 \\
 2x_2 + x_3 + x_6 + x_7 + x_8 + x_9 &= 10 \\
 x_2 + x_5 + 2x_6 + 2x_7 + x_8 + 2x_9 &= 16 \\
 \|(x_2, x_3)^T\| &\leq \tan \theta_1 x_1 \\
 \|(x_5, x_6)^T\| &\leq \tan \theta_2 x_4 \\
 \|(x_8, x_9)^T\| &\leq \tan \theta_3 x_7,
 \end{aligned}$$

(70)

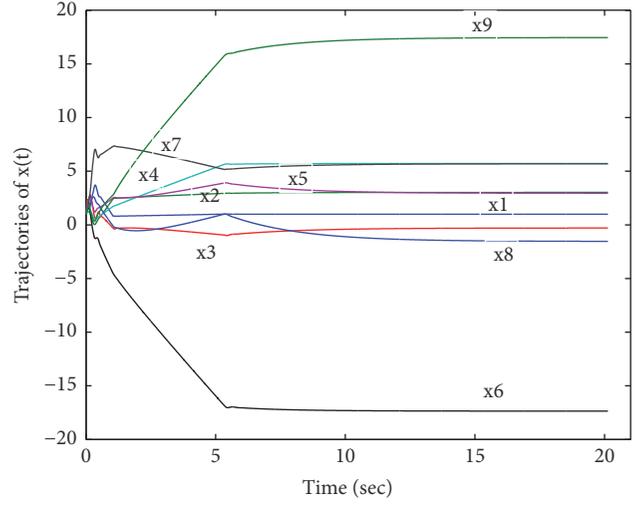


FIGURE 3: The  $x$  trajectories followed from Example 13.

where

$$\begin{aligned}
 A &= \begin{pmatrix} 1 & 1 & 2 & 0 & 2 & 0 & 1 & 0 & 0 \\ 4 & 6 & 3 & 0 & 1 & 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 3 & 1 & 1 \\ 0 & 2 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 2 & 2 & 1 & 2 \end{pmatrix}, \\
 b &= \begin{pmatrix} 15 \\ 34 \\ 24 \\ 10 \\ 16 \end{pmatrix}, \\
 c &= (0 \ 1 \ 2 \ 3 \ 1 \ 2 \ 2 \ 1 \ 1), \\
 \theta_1 &= 0.4\pi, \\
 \theta_2 &= 0.4\pi, \\
 \theta_3 &= 0.4\pi.
 \end{aligned}$$

(71)

The optimal solution of Example 13 is

$$\begin{aligned}
 x^* &= (0.987, 3.025, -0.297, 5.721, 2.945, \\
 &\quad -17.359, 5.692, -1.536, 17.450)^T
 \end{aligned}$$

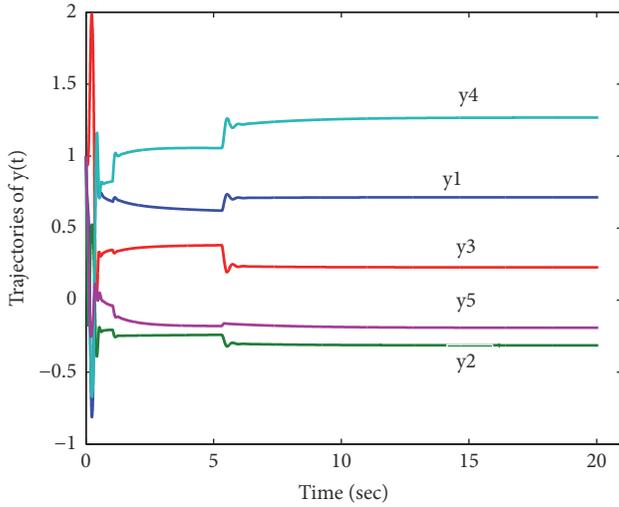
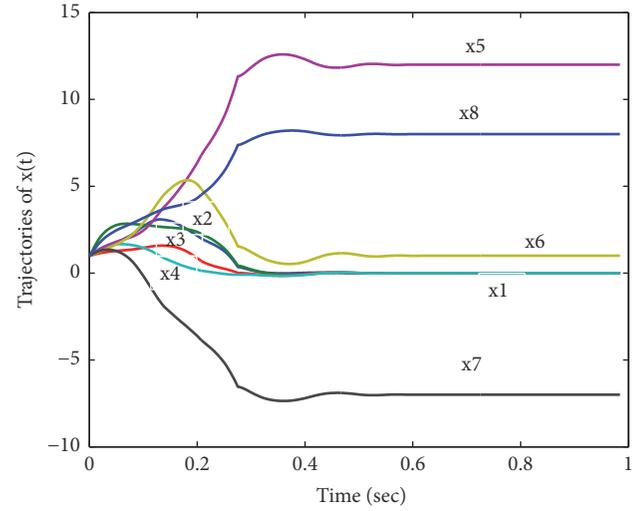
(72)

and

$$y^* = (0.714, -0.314, 0.228, 1.268, -0.192)^T.$$

(73)

Figures 3 and 4 depict the trajectories of neural network (47) with the initial values  $x_0 = (0, 0, 0, 0, 0, 0, 0, 0, 0)^T$  and  $y_0 = (0, 0, 0, 0, 0)^T$  converging to its solutions  $x^*$  and  $y^*$ , respectively.

FIGURE 4: The  $y$  trajectories followed from Example 13.FIGURE 5: The  $x$  trajectories followed from Example 14.

*Example 14.* Consider the problem with two 4-dimensional circular cones as follows:

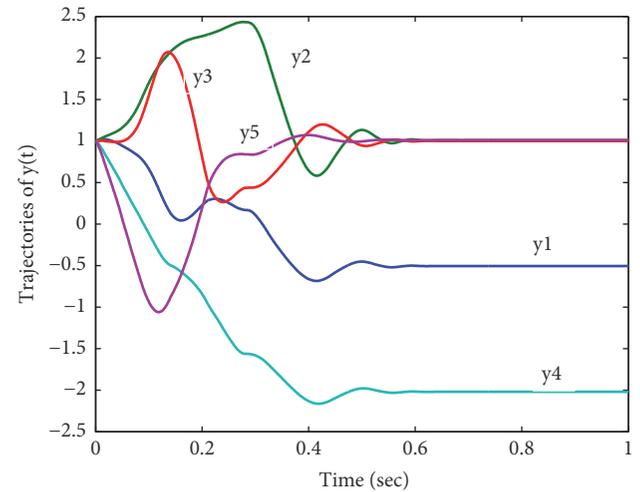
$$\begin{aligned}
 \min \quad & 6x_1 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 \\
 \text{s.t.} \quad & 3x_2 + x_4 + 2x_8 = 16 \\
 & 2x_2 + x_3 + x_4 + x_5 = 12 \\
 & x_1 + x_5 + x_6 + x_7 + x_8 = 14 \\
 & x_1 + x_4 + x_5 + x_7 = 5 \\
 & x_2 + x_5 + 2x_7 + x_8 = 6 \\
 & \|(x_2, x_3, x_4)^T\| \leq \tan \theta_1 x_1 \\
 & \|(x_6, x_7, x_8)^T\| \leq \tan \theta_2 x_5,
 \end{aligned} \tag{74}$$

where

$$A = \begin{pmatrix} 0 & 3 & 0 & 1 & 0 & 0 & 0 & 2 \\ 0 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 2 & 1 \end{pmatrix},$$

$$b = \begin{pmatrix} 16 \\ 12 \\ 14 \\ 5 \\ 6 \end{pmatrix},$$

$$c = (6 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1),$$

FIGURE 6: The  $y$  trajectories followed from Example 14.

$$\theta_1 = \frac{\pi}{4},$$

$$\theta_2 = \frac{\pi}{3}.$$

(75)

The optimal solution of Example 14 is

$$x^* = (0, 0, 0, 0, 12, 1, -7, 8)^T \tag{76}$$

and

$$y^* = (-0.505, 1.010, 1.000, -2.019, 1.010)^T. \tag{77}$$

Figures 5 and 6 depict the trajectories of neural network (47) with the initial values  $x_0 = (0, 0, 0, 0, 0, 0, 0, 0)^T$  and  $y_0 = (0, 0, 0, 0, 0)^T$  converging to its solutions  $x^*$  and  $y^*$ , respectively.

In paper [14], two neural networks are proposed for the second-order cone programs. The first neural network

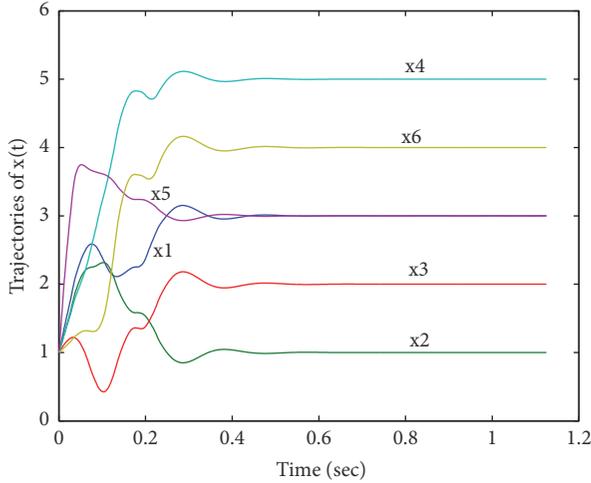


FIGURE 7: Transient behavior of the neural network (47) in Example 15.

uses the Fischer-Burmeister function to achieve an unconstrained minimization with a merit function. The second neural network utilizes the natural residual function with the cone projection (CP) function to achieve low computation complexity. The second neural network is a projection neural network. Here we compared our projection neural network method with the neural network based on the cone projection function in paper [14] by the next two test examples.

*Example 15.* In Example 12, let  $\theta_1 = \pi/4$ ,  $\theta_2 = \pi/4$ . Then the circular cone programming problem is converted into a second-order cone programming problem. This test problem is from Example 5.2 in paper [14]. This problem has an optimal solution  $x^* = (3, 1, 2, 5, 3, 4)^T$ .

Figures 7 and 8 depict the trajectories obtained using neural network (47) and the neural network with CP function in [14], respectively.

The simulation results show that both trajectories are convergent to  $x^*$ , but the neural network with the CP function yields the oscillating trajectory and has longer convergence time than the neural network (47).

*Example 16.* In Example 14, let  $\theta_1 = \pi/4$ ,  $\theta_2 = \pi/4$ . Then a second-order cone problem is obtained, which has an optimal solution  $x^* = (0, 0, 0, 0, 12, 1, -7, 8)^T$ .

Figures 9 and 10 depict the trajectories obtained using neural network (47) and the neural network with CP function in [14], respectively.

The simulation results show that both trajectories are convergent to  $x^*$ . In addition, neural network with the CP function yields the oscillating trajectory and has longer convergence time than the neural network (47).

*Example 17.* In this example, we use the grasping force optimization problem for the multifingered robotic hand [4, 8, 14] to demonstrate the effectiveness of the proposed neural network. The force optimization is to minimize the

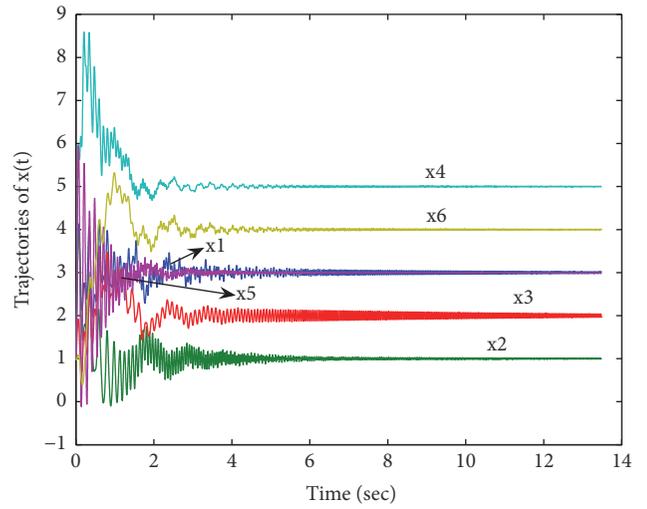


FIGURE 8: Transient behavior of the neural network with CP function in Example 15.

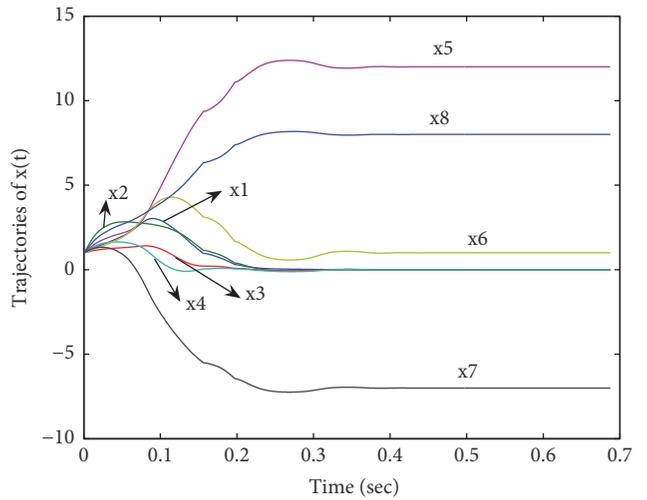


FIGURE 9: Transient behavior of the neural network (47) in Example 16.

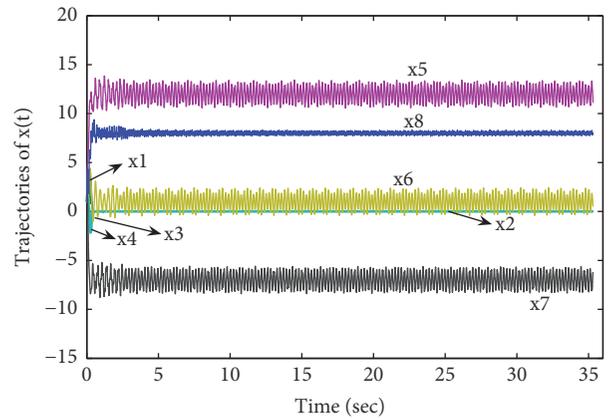


FIGURE 10: Transient behavior of the neural network with CP function in Example 16.

magnitude of grasping force from every finger applied to an object. For the robotic hand with  $m$  fingers, the optimization problem can be formulated as

$$\begin{aligned} \min \quad & \frac{1}{2} f^T f \\ \text{s.t.} \quad & Gf = -f_{ext} \\ & \|(f_{i1}, f_{i2})^T\| \leq \mu f_{i3} \quad (i = 1, 2, \dots, m), \end{aligned} \quad (78)$$

where  $f = [f_{11}, f_{12}, \dots, f_{m3}]$  is the grasping force,  $G$  is the grasping transformation matrix,  $f_{ext}$  is the time-varying external wrench, and  $\mu$  is the friction coefficient. Problem (78) is a convex quadratic circular cone programming problem. In [14], problem (78) is formulated as a convex quadratic second-order cone programming problem by variable transformation.

In this example, we consider a three-fingered grasping force optimization example [14]. The three-finger robot hand grasps a polyhedral with the grasp points  $[0, 1, 0]^T$ ,  $[1, 0.5, 0]^T$ , and  $[0, -1, 0]^T$ , and the robot hand moves along a vertical circular trajectory of radius  $r$  with a constant velocity  $v$ . Let  $x = [f_{13}, f_{11}, f_{12}, f_{23}, f_{21}, f_{22}, f_{33}, f_{31}, f_{32}]^T$ . Then problem (78) is reformulated as

$$\begin{aligned} \min \quad & \frac{1}{2} x^T Q x \\ \text{s.t.} \quad & Ax = b \\ & \|(x_2, x_3)\| \leq \tan(\theta_1) x_1 \\ & \|(x_5, x_6)\| \leq \tan(\theta_2) x_4 \\ & \|(x_8, x_9)\| \leq \tan(\theta_3) x_7, \end{aligned} \quad (79)$$

where  $Q = \text{diag}(1, 1, 1, 1, 1, 1, 1, 1, 1)$  and

$$A = \begin{pmatrix} 0 & 0 & 1 & -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & -0.5 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0.5 & 0 & -1 & 0 & 1 & 0 \end{pmatrix}, \quad (80)$$

$$b = \begin{pmatrix} 0 \\ -f_c \sin \theta(t) \\ M_1 g - f_c \cos \theta(t) \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$c = (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)^T,$$

$$\theta_1 = \theta_2 = \theta_3 = \text{actan}(\mu^{-1}).$$

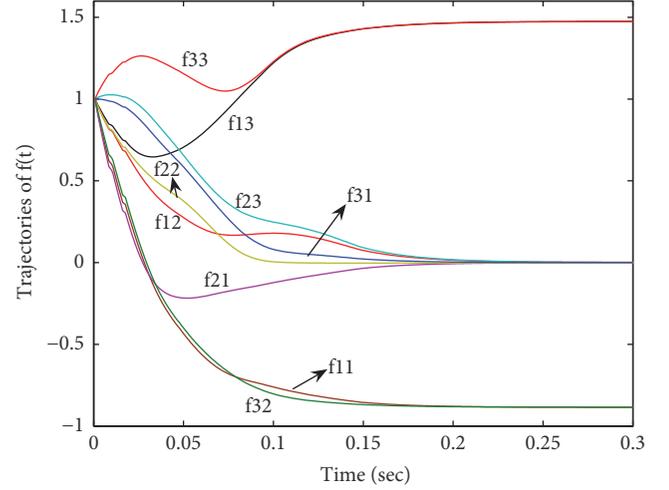


FIGURE 11: Transient behavior of the neural network (47) when  $t = 0.5s$ .

Here,  $M_1$  is the mass of the polyhedral,  $g = 9.8m/s^2$ ,  $f_c = Mv^2/r$  the centripetal force,  $t$  the time, and  $\theta(t) = vt/r \in [0, 2\pi]$ . In this example, we set the data as follows:  $M_1 = 0.1kg$ ,  $r = 0.2m$ ,  $n = 0.4\pi m/s$ , and  $\mu = 0.6$ . In the time-varying grasping force, we only test numerical examples when  $t = 0s$  and  $t = 0.5s$ . The other results are easily given when  $t \in (0, 0.5)$ .

Because the neural network (47) is designed to solve the linear circular cone programming problem, it cannot be used to solve problem (34) directly. But our neural network can be extended to convex quadratic circular cone programming problem easily. The extension method is only to change the residual function  $e(u)$  and the matrix  $M$  in neural network (47) as follows:

$$M = \begin{bmatrix} Q & -A^T \\ A & 0 \end{bmatrix}, \quad (81)$$

$$e(u) = \begin{bmatrix} Tx - P_K(Tx - T^{-1}(Qx + c - A^T y)) \\ Ax - b \end{bmatrix}.$$

The results of the projection neural networks (47) to solve Example 17 are shown in Figures 11 and 12 when  $t = 0s$  and  $t = 0.5s$ , respectively. The simulation results demonstrate that the neural networks are convergent for the grasping force optimization problem.

For these simulation examples, the proposed network in (47) with other initial points always converges to the theoretical optimal solution. The simulation results demonstrate that the neural networks are effective for the circular cone programming.

## 6. Conclusion

In the paper, a projection neural network method is proposed for the circular cone programming. The projection equation and the contraction technique are used to construct the energy function and dynamical system. In the method,

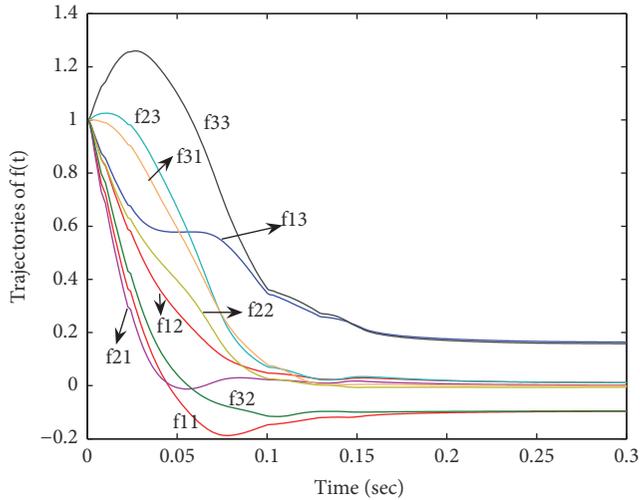


FIGURE 12: Transient behavior of the neural network (47) when  $t = 0s$ .

the projection and contraction technique accelerates the convergence of the neural network method. In addition, the proposed neural network requires less state variables, so the neural network method is simple and efficient.

**Data Availability**

The data used to support the findings of this study are available from the corresponding author upon request.

**Conflicts of Interest**

The authors declare that there are no conflicts of interest regarding the publication of this paper.

**Acknowledgments**

This work was supported by the National Science Foundations for Young Scientists of China (11101320, 61201297), National Science Basic Research Plan in Shaanxi Province of China (2015JM1031), and the Fundamental Research Funds for the Central Universities (JB150713).

**References**

[1] J. Dattorro, *Meboo Publishing*, Convex Optimization and Euclidean Distance Geometry. Meboo Publishin, Palo Alto, 2005.

[2] J. Zhou and J.-S. Chen, “Properties of circular cone and spectral factorization associated with circular cone,” *Journal of Nonlinear and Convex Analysis. An International Journal*, vol. 14, no. 4, pp. 807–816, 2013.

[3] J. Zhou, J.-S. Chen, and B. S. Mordukhovich, “Variational analysis of circular cone programs,” *Optimization. A Journal of Mathematical Programming and Operations Research*, vol. 64, no. 1, pp. 113–147, 2015.

[4] S. P. Boyd and B. Wegbreit, “Fast computation of optimal contact forces,” *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1117–1132, 2007.

[5] B. León, A. Morales, and J. Sancho-Bru, “Robot Grasping Foundations,” in *From Robot to Human Grasping Simulation*, vol. 19 of *Cognitive Systems Monographs*, pp. 15–31, Springer International Publishing, Cham, 2014.

[6] JF. Bonnans and HR. Cabrera, “Perturbation analysis of second-order cone programming problems,” *Math Program*, vol. 104, pp. 205–227, 2005.

[7] A. Ruszczyński, *Nonlinear Optimization*, Princeton University Press, New Jersey, 2006.

[8] F. Alizadeh and D. Goldfarb, “Second-order cone programming,” *Mathematical Programming*, vol. 95, no. 1, Ser. B, pp. 3–51, 2003.

[9] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, “Applications of second-order cone programming,” *Linear Algebra and its Applications*, vol. 284, no. 1-3, pp. 193–228, 1998.

[10] X. D. Chen, D. Sun, and J. Sun, “Complementarity functions and numerical experiments on some smoothing Newton methods for second-order-cone complementarity problems,” *Computational optimization and applications*, vol. 25, no. 1-3, pp. 39–56, 2003.

[11] C. Kanzow, I. Ferenczi, and M. Fukushima, “On the local convergence of semismooth Newton methods for linear and nonlinear second-order cone programs without strict complementarity,” *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 297–320, 2009.

[12] J. Zhou, J.-S. Chen, and H.-F. Hung, “Circular cone convexity and some inequalities associated with circular cones,” *Journal of Inequalities and Applications*, 2013:571, 17 pages, 2013.

[13] P. Ma, Y. Bai, and J.-S. Chen, “A self-concordant interior point algorithm for nonsymmetric circular cone programming,” *Journal of Nonlinear and Convex Analysis. An International Journal*, vol. 17, no. 2, pp. 225–241, 2016.

[14] C.-H. Ko, J.-S. Chen, and C.-Y. Yang, “Recurrent neural networks for solving second-order cone programs,” *Neurocomputing*, vol. 74, no. 17, pp. 3646–3653, 2011.

[15] X. Miao, J.-S. Chen, and C.-H. Ko, “A smoothed NR neural network for solving nonlinear convex programs with second-order cone constraints,” *Information Sciences*, vol. 268, pp. 255–270, 2014.

[16] X. Miao, J.-S. Chen, and C.-H. Ko, “A neural network based on the generalized FB function for nonlinear convex programs with second-order cone constraints,” *Neurocomputing*, vol. 203, pp. 62–72, 2016.

[17] Y. Xia, J. Wang, and L.-M. Fok, “Grasping-force optimization for multifingered robotic hands using a recurrent neural network,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 549–554, 2004.

[18] B. He and H. Yang, “A neural-network model for monotone linear asymmetric variational inequalities,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 11, no. 1, pp. 3–16, 2000.

[19] R. K. Miller and A. N. Michel, *Ordinary differential equations*, Academic Press, Inc. [Harcourt Brace Jovanovich, Publishers], New York-London, 1982.

[20] J. Zabczyk, *Mathematical Control Theory: An Introduction*, Birkhuser, Boston, Mass, USA, 1992.

[21] J. Faraut and A. Korányi, *Analysis on Symmetric Cones*, Oxford University Press, New York, NY, USA, 1994.

[22] J. r. Outrata and D. Sun, “On the coderivative of the projection operator onto the second-order cone,” *Set-Valued Analysis*, vol. 16, no. 7-8, pp. 999–1014, 2008.

- [23] LC. Kong, L. Tuncel, and NH. Xiu, "Clarke Generalized Jacobian of the Projection onto Symmetric Cones," *Set-Valued and Variational Analysis*, vol. 17, pp. 135–151, 2009.
- [24] D. G. Luenberger, *Introduction to Linear and Nolinear Programming*, Addison-wesley Publishing Company, Boston, 1973.
- [25] R. M. Golden, *Mathematical methods for neural network analysis and design*, MIT Press, Cambridge, 1996.
- [26] Y. Xia and J. Wang, "A recurrent neural network for solving nonlinear convex programs subject to linear constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 16, no. 2, pp. 379–386, 2005.

## Research Article

# A New Generalized Deep Learning Framework Combining Sparse Autoencoder and Taguchi Method for Novel Data Classification and Processing

Ahmad M. Karim,<sup>1</sup> Mehmet S. Güzel ,<sup>2</sup> Mehmet R. Tolun ,<sup>3</sup>  
Hilal Kaya,<sup>4</sup> and Fatih V. Çelebi<sup>4</sup>

<sup>1</sup>Electrical and Electronics Engineering, Aksaray University, Aksaray 68100, Turkey

<sup>2</sup>Computer Engineering Department, Ankara University, Ankara 06830, Turkey

<sup>3</sup>Computer Engineering Department, Baskent University, Ankara 06790, Turkey

<sup>4</sup>Computer Engineering Department, AYBU, Ankara 06830, Turkey

Correspondence should be addressed to Mehmet S. Güzel; [mguzel@ankara.edu.tr](mailto:mguzel@ankara.edu.tr)

Received 12 March 2018; Revised 3 May 2018; Accepted 13 May 2018; Published 7 June 2018

Academic Editor: Nivaldo Rodríguez

Copyright © 2018 Ahmad M. Karim et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Deep autoencoder neural networks have been widely used in several image classification and recognition problems, including hand-writing recognition, medical imaging, and face recognition. The overall performance of deep autoencoder neural networks mainly depends on the number of parameters used, structure of neural networks, and the compatibility of the transfer functions. However, an inappropriate structure design can cause a reduction in the performance of deep autoencoder neural networks. A novel framework, which primarily integrates the Taguchi Method to a deep autoencoder based system without considering to modify the overall structure of the network, is presented. Several experiments are performed using various data sets from different fields, i.e., network security and medicine. The results show that the proposed method is more robust than some of the well-known methods in the literature as most of the time our method performed better. Therefore, the results are quite encouraging and verified the overall performance of the proposed framework.

## 1. Introduction

Machine learning (ML) is a popular branch of artificial intelligence (AI) that does not need to be explicitly programmed but allows machines to obtain new skills and predict results with high accuracy. Deep learning (DL) is a new version of ML which recently have been applied in many fields from computer vision to high dimension data processing. DL achieved the state-of-the-art results [1, 2]. Essentially, DL achieves great improvement in solving problems that have resisted the trials of the AI society for more than three decades. It should be noted that DL can predict comprehensive outcomes by requiring little engineering, which cannot be compared by the conventional AI based approaches. DL will be applied to different fields in the near future due to its flexible and generic structure. Development

of innovative learning algorithms and new structures for deep neural networks will merely speed up this progress [3]. Recently, deep autoencoders have shown state-of-the-art achievement on different machine learning tasks which relies on unsupervised learning algorithms [4]. Deep autoencoders have been widely used in different fields from image recognition to computer network, etc. Lore et al. (2017) proposed a deep autoencoder-based method to separate features of signal from images having low light and also modify glare images without over saturating the lighter accessories in images with a high variety [5]. K. Sun et al. proposes that, a divergence of the stacked sparse denoising autoencoder, synthetic data used for training it, the new proposed extreme learning machine autoencoder (ELM-AE) called generalized extreme learning machine autoencoder (GELM-AE) adds the forked regularization to the aim of ELM-AE [6]. In

[7] Yihui Xiong et al. trained an autoencoder network to encode and remodel a geochemical pattern population with strange complex multivariate probability division. In [8] Lyle D. Burgoonet et al. trained the autoencoder to predict estrogenic chemical substances (APECS). APECS consists of two deep autoencoder models which is less convoluted than the USEPA's method and performs at least the same achievement. However, proposed idea implements accuracies of 91% versus 86% and 93% versus 93% on the in vitro and in vivo datasets used in validating the US EPA method. Chaoqun Hong et al. proposed a new pose retrieval technique which focuses on multimodal integration feature extraction and backpropagation deep neural network by using multilayered deep neural network with nonlinear mapping [9]. In [10] Tzu-Hsi Song et al. focused on bone marrow trepan biopsy images and proposed a hybrid deep autoencoder (HDA) network with Curvature Gaussian method for active and exact bone marrow hematopoietic stem cell detection via related high-level feature correspondence. In [11] Yosuke Suzuki et al. proposed a collaborative filtering based recommendation algorithm that employs the variation of similarities among users derived from different layers in stacked denoising autoencoders. Yu-Dong Zhang et al. presented a novel system counting on susceptibility-weighted imaging as computer-aided detection application which increased in the last years. Unsupervised feature learning was done by using SAE. Then, a deep autoencoder neural network was formed using the learned features and stacked autoencoders for training all of them together as supervised learning. The proposed approach produced a sensitivity of "93.20±1.37%", a specificity of "93.25±1.38%", and an accuracy of "93.22±1.37%", the results obtained over "10x10-fold" cross validation [12]. As presented above, deep autoencoders have gathered lots of attention from researchers recently.

Taguchi Method is a statistical technique proposed by Taguchi and Konishi, which was essentially proposed for optimizing the quality manufacturing process development [13]. Especially in recent years, this method is used in number of critical studies to design experiment with best performance by different disciplines such as Engineering, Biotechnology, and Computer Science. For instance, Mei-Ling Huang et al. (2014) combined a feature selection technique with SVM recursive feature elimination approach to validate the classification accuracy for Dermatology and Zoo databases [14]. In this study, the Taguchi Method was adapted and combined with a SVM classifier so as to increase the overall classification accuracy by optimizing 'C' and 'γ' parameters respectively. Authors claim that the proposed method can produce more than 95% accuracy for Dermatology and Zoo databases. A study includes multistage metal forming process by considering workability and also employs Taguchi Method for optimization [15]. For this study, the Taguchi Method is combined with artificial neural network to minimize the objective functions with respect to the forming process that the combinations of parameters used in finite element simulation are determined by orthogonal array in statistical design of experiments. The train data for artificial neural networks are obtained from orthogonal array and the result of simulation process. Huimin Wang

et al. (2014) adopted the Taguchi Method to analyze the effect of "inertia weight", "acceleration coefficients", "population size", "fitness evaluations", and population topology on particle swarm optimization algorithm (PSO) and to determine the best mix of them for various optimization problems. The experimental results illustrate that all the benchmark functions have their optimum solutions after the tuning process. Furthermore, acceptable results are also presented by the article when dealing with the optimization design of a "Halbach permanent magnet" motor. The paper concludes that the PSO based Taguchi Method is quite appropriate for such popular engineering problems [16]. A recent study published in (2016) proposed a new predictive modelling of material removal rate (MRR) by employing Taguchi-entropy weight based GRA to optimize an artificial neural network [17]. Further recent studies using Taguchi Method can be also seen in the corresponding articles [18–22].

This paper introduces a novel deep autoencoder based architecture optimized by Taguchi Method (see Section 2.1). The proposed architecture was employed in four different fields to show its performance, the presented architecture shows satisfactory results, and this encouraged authors to employ this framework in other different fields. The structure of paper consists of the proposed framework, experimental results, and conclusion.

## 2. The Proposed Framework

This study proposes a new method for optimizing deep autoencoders structure for processing data. The proposed deep learning architecture employs stacked autoencoders supported by Taguchi Method for parameter optimization in a reasonable amount of time. First a brief explanation of the stacked sparse autoencoder and Taguchi Method is presented respectively. Afterwards the proposed architecture, shown in Figure 2, is discussed.

*2.1. Stacked Sparse Autoencoder.* Supervised learning is one of the most powerful tools of AI. The stacked sparse autoencoder (SSAE) is essentially a neural network consisting of multiple layers of sparse autoencoders and mainly used as an unsupervised feature extraction method that automatically learns from unlabelled data. Output of each layer is wired to the inputs of the succeeding layer. Having a trained autoencoder essentially refers to estimate optimal parameters by reducing the divergence between input 'x' and output 'x̂'. An example autoencoder is illustrated in Figure 1. The mapping between input 'x' and output 'x̂'. is given following equations:

$$\hat{x} = f(x) \quad (1)$$

$$n_1^{(1)} = M_f \left( w_{11}^{(1)} x_1 + \cdots + w_{15}^{(1)} x_{5+} + b_1^{(1)} \right) \quad (2)$$

$$n_i^{(1)} = M_f \left( w_{i1}^{(1)} x_1 + \cdots + w_{i5}^{(1)} x_{5+} + b_i^{(1)} \right) \quad (3)$$

where  $M()$  is an activation using sigmoid logistic function.

TABLE I: Orthogonal array selection table.

		Number of Parameters (NoP)							
		2	3	4	5	6	7	8	9
Number of Levels	2	L4	L4	L8	L8	L12	L12	L12	L16
	3	L9	L9	L9	L18	L18	L18	L27	L27
	4	L16	L16	L16	L16	L32	L32	L32	L32
	5	L25	L25	L25	L25	L25	L50	L50	L50

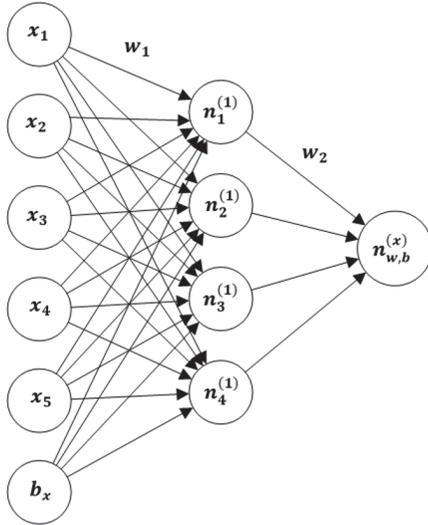


FIGURE 1: A simple example of autoencoder.

The final expression can be shown as follows:

$$n_{w,b}(x) = M_f \left( w_{11}^{(2)} n_1^{(2)} + \dots + w_{15}^{(2)} n_5^{(2)} + \dots + b_1^{(2)} \right) \quad (4)$$

The discrepancy between the input ' $x$ ' and output ' $x'$ '. is defined by using a cost function. This functions' first term refers to the MSE whereas the second one is the regularization term. Different algorithms are preferred to solve the optimal parameters of the network; the details can be seen in [45].

**2.2. Taguchi Method.** Taguchi Method is a statistical robust design method that was first proposed to improve the quality of manufactured product and more recently also applied to a variety of fields from engineering to marketing [13, 22, 46]. Three concepts were considered by the Taguchi concepts, namely, Taguchi loss function, offline quality control, and orthogonal arrays for experimental design. Taguchi Method offers a methodology for designing of experiments. For instance, if an experiment is aimed at heating of wire by passing the electricity through it, then different control parameters from material type to diameter of wire are considered. Those parameters may have various values. DOE allows you to obtain the parameters and their values in an efficient manner. An example orthogonal selection table is illustrated in Table 1.

Essentially those arrays tend to adopt a methodical way to permute and combine the collaboration among different

parameters. Besides, unlike the full factorial experiment, there is no need to carry out each experiment respectively. To obtain the objective value or best accuracy, Taguchi Method decreases the number of necessary experiments by using orthogonal arrays (OA). This reduces the number of experiments to be performed and also reduces the overall cost. This arrays are essentially predefined matrices, including control parameters and number of experiments. The purpose of the Taguchi Method is to design an experiment that reduces the effect of the operator that cannot be controlled with a least amount of experiments [46, 47]. The selection of an appropriate orthogonal array is mainly based on the number of control parameters and corresponding levels. Orthogonal arrays are varied from L4 to L50 (see Table 1). The more numbers of control parameters yield the higher the numbers after "L". Design of experiments is performed by employing the defined orthogonal array [47]. The iterations of experiments can be performed once the OA is carefully chosen. The number of iterations is then confirmed based on the complexity of the experiments. As aforementioned, the purpose of Taguchi Method to design an experiment is to reduce the effect of the operator that cannot be controlled with a least amount of experiments [47]. Taguchi Method is a powerful technique for supplying the best set among different stages of various parameters. The measure used in Taguchi Method is signal-to-noise (S/N) ratio to measure and esteem the superiority features that is the ratio of signal (S) to the operator of noise (N). Various S/N ratios were presented but three of them are considered standard [48]. The first standard is "smaller-is-better", when the objective account of the quality variable  $y$  is zero. In this case, the S/N ratio can be defined as follows:

$$\eta = -10 \log \sum \frac{x^2}{k} \quad (5)$$

In (1),  $x$  is the account of the experimental control and  $k$  is the number of experiments. The second standard is "larger-is-better" when the zero account of the quality variable  $y$  is unlimited and in this case, the S/N ratio can be realized as follows:

$$\eta = -10 \log \sum \frac{(1/x^2)}{k} \quad (6)$$

Here,  $x$  is experimental surveillance account and  $k$  is the number of experiments. The last standard is "nominal-is-best": in these styles of problems, the objective account of the

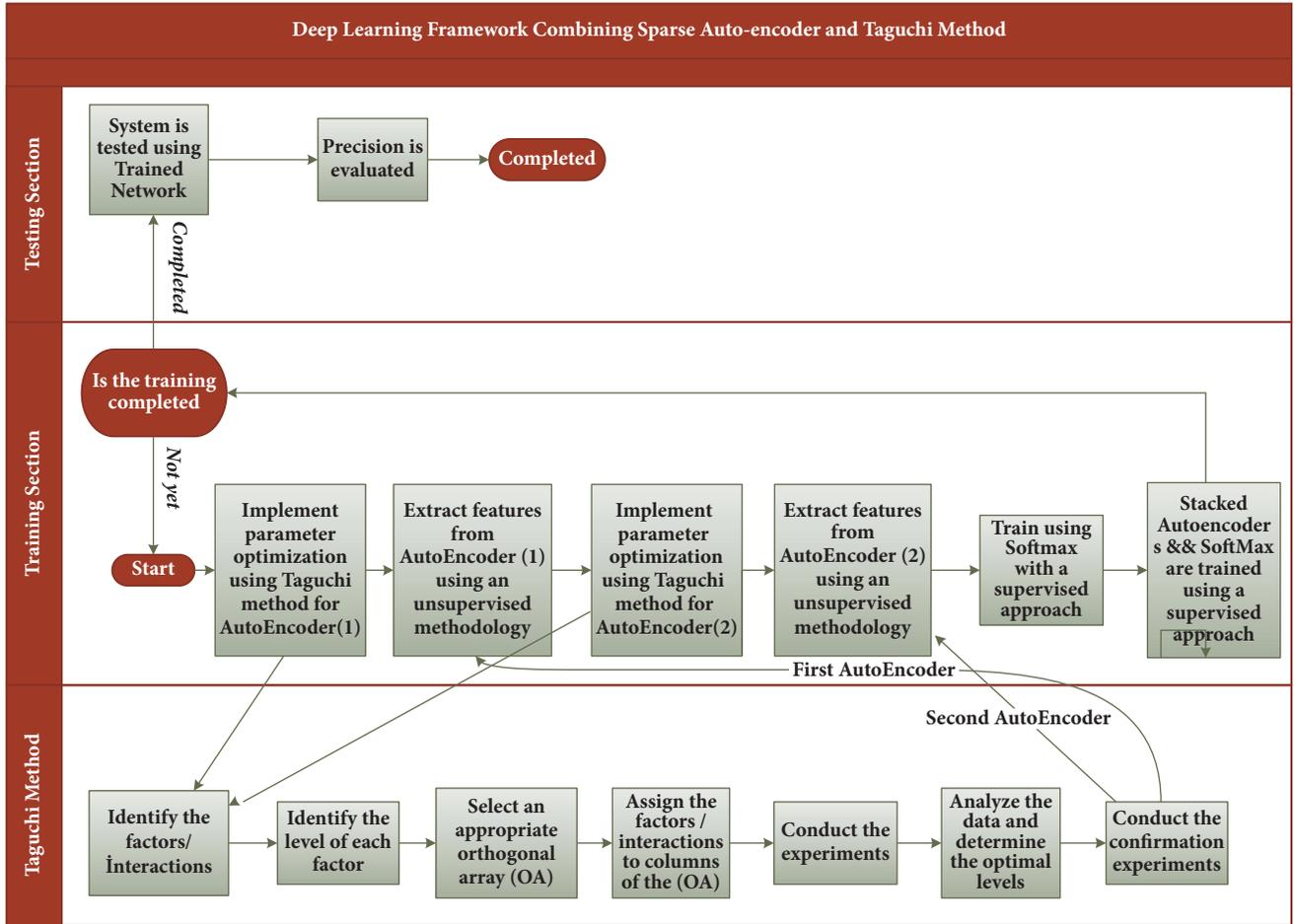


FIGURE 2: Deep learning framework combining sparse autoencoder and Taguchi Method.

quality variable  $x$  is specific. According to which, the S/N ratio can be realized as follows:

$$\eta = 10 \log \sum \frac{x^{-2}}{\sigma} \quad (7)$$

Here,  $x$  is the average account for the experimental surveillance and  $\sigma$  is the criterion variation of the experimental surveillance [49, 50].

Overall, the average values of the signal-to-noise (S/N) ratio for each level of each of the parameter are calculated. The maximum and minimum values of differences are presented that the appropriate S/N ratio is decided based on the experimental strategy. This principally has a great influence on assessing the experiments.

**2.3. Deep Learning Framework Combining Sparse Autoencoder and Taguchi Method.** As illustrated in Figure 1, deep neural network is designed from two autoencoders and SoftMax layers, each one of them was trained alone as unsupervised training without using labelled data; the purpose of these first two layers is essentially to extract appropriate features; automatic feature extraction is one of the powerful characteristics of deep learning based architectures. The following

section will briefly introduce the Taguchi Method whereas the following subsection will introduce the proposed method and the corresponding deep learning based architecture. The third layer is the SoftMax layer, which is one of the leading feature classifiers and is responsible for classifying the features that are extracted from the previous layers. The final layer is to stack all layers and train them together by using labelled data in supervised fashion. This basically allows converting an unsupervised learning architecture into a supervised learning architecture. To obtain the best performance from the first autoencoder, Taguchi Method is integrated into the model aiming to estimate optimized combination of five parameters of first autoencoder, namely, *L2 Weight Regularization*, *Sparsity Regularization*, *Sparsity Proportion*, *Hidden Size*, and *Max Epochs*. The effect of an L2 regularizer for the weights of the network is controlled by *L2 Weight Regularization* but not control the biases.

*L2 Weight Regularization* parameter should be very small and is represented in the following:

$$\Omega_{\text{weight}} = \frac{1}{2} \sum_l \sum_j^n \sum_i^k w_{ji}^{(l)2} \quad (8)$$

where number of hidden layers is represented by  $L$ , the number of observations is represented by  $n$ , and the training data variables number is represented by  $k$ .

The sparsity regularizer effect is controlled by a Sparsity Regularization parameter, dealing to force a chain on the sparsity of the output from the hidden layers. This is different from applying a sparsity regularizer to the weights that Sparsity Regularization term can be the Kullback-Leibler divergence (KL) function as illustrated in the following:

$$\begin{aligned}\Omega_{\text{sparsity}} &= \sum_{i=1}^{D(1)} KL(\rho \parallel \hat{\rho}_i) \\ &= \sum_{i=1}^{D(1)} \rho \log(\rho \parallel \hat{\rho}_i) + (1 - \rho) \log\left(\frac{1 - \rho}{1 - \hat{\rho}_i}\right)\end{aligned}\quad (9)$$

where  $\rho$  represents the desired value,  $\hat{\rho}_i$  represents the average output activation of a neuron  $i$ , and  $KL$  is the function that measures the variation between two probabilities distribution through the same data. As it can be inferred that the equation result value gets close to zero between  $\rho$  and  $\hat{\rho}_i$  when input and output data resemble each other. On the other hand, when those values are not close to each other, the sparsity will take a larger value [20].

Alternatively, sparsity regularizer parameter is controlled by Sparsity Proportion (SP) parameter. The sparsity of the output from each hidden layer is controlled by the Proportion parameter. A low value for SP normally leads all neurons in the hidden layer specialized by only producing a high output value for a small amount of training examples. For instance, if SP value is selected as "0.2", an average output for each neuron becomes "0.2" in the hidden layer over the training examples. The optimum value of SP varies depending on the nature of the problem between 0 and 1. Therefore, the technique for selecting the optimal value is very significant to improve the overall performance of the sparse autoencoder [21]. In addition, Hidden Size (HS) is a parameter which controls the size of the feature on each layer so; it affects the performance of the autoencoder. The last parameter is Maximum Epochs; one epoch represents one entire training cycle on the training data. Every sample in the training data is seen once, you start with the 2<sup>nd</sup> epoch. However, the Maximum Epochs mean, for example, if maximum epoch equals 10, this means the weights will be updated at least 10 times.

All previously defined parameters are employed in the training phase and directly influence the success of the training process. The cost function of training sparse autoencoder is also illustrated in (10). The training algorithm tries to reduce the cost function by finding the optimal parameters that essentially aims to reduce the value of  $E$ .

$$\begin{aligned}E &= \frac{1}{N} \sum_{n=1}^N \sum_{k=1}^k (x_{kn} - \hat{x}_{kn})^2 + \lambda * \Omega_{\text{weights}} + \beta \\ &\quad * \Omega_{\text{sparsity}}\end{aligned}\quad (10)$$

Here,  $E$  represented the loss rate (error rate),  $x$  is represented the input features,  $\hat{x}$  is the reconstructed features,  $\lambda$  is

the coefficient for the L2 *Weight Regularization*, and  $\beta$  is coefficient for the Sparsity Regularization.

The given problem includes two autoencoders. Each of those autoencoders has 5 parameters and each parameter can be defined with 5 different levels. Consequently, the traditional method for finding best combination of parameters for two autoencoders requires  $5^5 + 5^5 = 3125 + 3125 = 6250$  trails so as to test all parameter combinations by using full factorial design. This means that each autoencoder entails  $5^5$  trails to obtain the best combination of parameters. Hence, a more optimized approach has been proposed in this study. According to which Taguchi Method was utilized for finding the optimal parameters for the system by performing only 25 experiments, select L25 orthogonal index (5 parameters and 5 levels in each parameter); see Section 2.2. As the first autoencoder is performed by doing 25 experiments, the most optimum parameters were also determined by Taguchi Method and best performance for the second autoencoder as well. This means that the total experiments for the first two layers in our system are  $25 + 25 = 50$ . As mentioned above, at the last step, all three components are stacked and trained in a supervised fashion by using backpropagation on multilayer network for improving the network performance. In order to validate the performance of the proposed system, a series of experiments were conducted.

### 3. Experimental Results

A computer with Intel Core i7-6700 CPU @ 2.60-GHz and 8-GB RAM is used for running the proposed framework which is used in several applications to detect computer network attacks including DDoS and IDS attacks and Epileptic Seizure Recognition and Handwritten Digit classification. The results obtained with the proposed method are compared to a number of studies in the respective field. In addition, some of the techniques implemented in this paper to compare the results with our proposed method are SVM, neural network, SoftMax and stacked sparse autoencoder based support vector machine (SSAE-SVM). Each dataset and corresponding result will be detailed in the following subsections respectively.

*3.1. DDoS Detection Using the Proposed Framework.* Distributed Denial of Service attack is an offensive and threatening intrusive threats to online servers, websites, networks, and clouds. The purpose of DDoS attack is to exhaust exchequer and to expend bandwidth of a network system. Due to the harmonious nature of DDoS attack, an attacker can generate massive amount of attack traffic using a huge number of compromised machines to smash a system or website [51, 52]. Many organizations such as Amazon, eBay, CNN, and Yahoo were the victims of DDoS attacks in the recent past. In this paper, our new framework was used to detect DDoS attack proposed in [23], which presented four attacks types (Smurf, UDP Flood, SIDDOS, HTTP Flood, and normal). This dataset consists of 27 features (SRC ADD, DES ADD, PKT ID, FROM NODE, TO NODE, PKT TYPE, PKT SIZE, FLAGS, FID, SEQ NUMBER, NUMBER

TABLE 2: Autoencoder 1 upper and lower level values of factors for DDOS detection.

Factors	Lower Limit	Upper Limit
Hidden Size (HS)	19	22
Max Epochs (ME)	300	500
L2 Weight Regularization (L2)	0.0035	0.0045
Sparsity Regularization (SR)	4	6
Sparsity Proportion (SP)	0.13	0.16

OF PKT, NUMBER OF BYTE, NODE NAME FROM, NODE NAME TO, PKT IN, PKTOUT, PKTR, PKT DELAY NODE, PKTRATE, BYTE RATE, PKT AVG SIZE, UTILIZATION, PKT DELAY, PKT SEND TIME, PKT RESEVED TIME, FIRST PKT SENT, LAST PKT RESEVED). In Table 2, parameters are classified into 5 classes and we recognize the upper and lower boundaries of the parameters. The upper and lower boundaries of these parameters are determined by using trial and error approach. This approach considers the results of the predefined experiments and studies.

As mentioned above, dataset consists of five classes that each class consists of 800 samples. 50% of them were used for training, and also the other % 50 were used for testing. Consequently, the proposed framework was trained by employing 2000 samples and then it was tested by employing another 2000 samples. Moreover, in Table 3, the operators' level values are presented. Minitab program experiments results are presented in Table 4. The error accounts obtained from stratifying the parameters to the autoencoder 1 are represented in Table 5. Root mean square error (RMSE) is used to measure the performance of the autoencoder 1, the smallest value which is closed to zero means that the performance is well.

$$x = \frac{\sqrt{\sum_{i=1}^n (X_{(Obs,i)} - X_{(model,i)})^2}}{n} \quad (11)$$

where spotted rate is represented by  $X_{Obs}$  and modelled rate represented by  $X_{Model}$  at time/place 'i'. The experiment results acquired by using the Taguchi experimental design were estimated by transforming them into S/N ratios. The results acquired by using the Taguchi experimental design were predestined by transforming the results into signal/noise (S/N) ratios (See Table 5).

Now, Table 6 presents the best parameters for autoencoder 1 which represented the first layer in the deep autoencoder neural network. The parameters of autoencoder 2 which represented the second layer can be obtained by using the same steps in different ranges for each parameter Table 7.

By following Tables 8, 9, and 10, the best parameters are obtained in Figure 4 and Table 11; the same procedures in Table 3, Table 4, Table 5, respectively, were used to find the best parameters that are represented in Figure 3 and Table 5. This means that the best parameters of each autoencoder were determined in minimum number of tests.

After finding the best parameters, for each autoencoder, this leads to obtaining the best performance for training each autoencoder by using the best parameters. On the other hand,

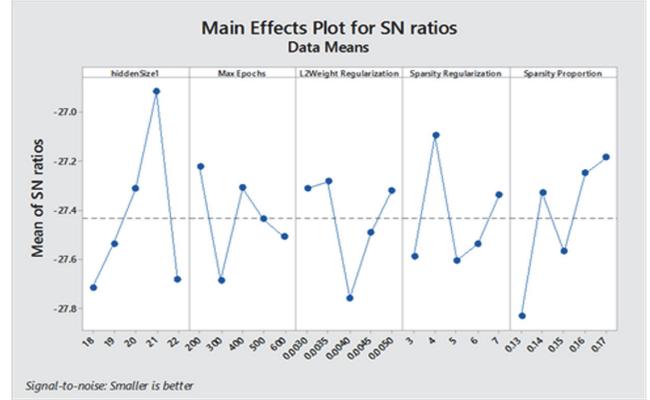


FIGURE 3: Autoencoder 1 main effect of experimental parameters on the S/N ratio for DDoS Attack Detection.

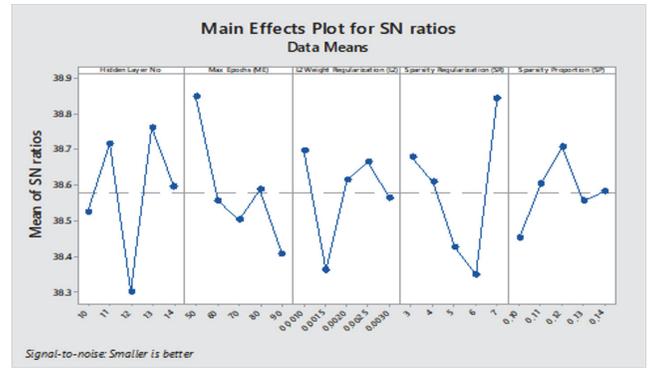


FIGURE 4: The main effect of experimental parameters on the S/N ratio for autoencoder 2.

the results that were obtained from the system presented by using confusion matrix for detailed analysis for each type of DDos attack is seen in Figure 5. The experimental results show that proposed method has satisfactory results when compared to other methods.

Detection accuracy of 99.6% makes the proposed method slightly better than the other methods as shown in Table 12. The other feature of proposed method is that this system can learn effectively by using only 2000 samples which is very little when compared to previous methods. Data collection is very difficult and expensive procedure so that the system that learns faster by using less number of data sample is more practical from others. The confusion matrix notation is used to present results in a more detailed fashion and to be more understandable. The proposed framework results is compared with number of methods proposed in [23], also with number of methods proposed by us to detect DDos attacks such as SSAE-SVM [24], SVM, and SoftMax classifiers. Table 12 illustrates that the proposed framework produces the best results compared with the state-of-the-art methods for this problem.

3.2. *IDS Attack.* In computer security systems, Intrusion Detection Systems (IDS) have become a necessity because of the growing demand in unlawful access and attacks. In

TABLE 3: Autoencoder (1) upper and lower level values of factors for DDOS attack detection.

Factors	Level 1	Level 2	Level 3	Level 4	Level 5
HS	18	19	20	21	22
ME	200	300	400	500	600
L2	0.003	0.0035	0.004	0.0045	0.005
SR	3	4	5	6	7
SP	0.13	0.13	0.15	0.16	0.17

TABLE 4: Autoencoder 1 parameters values obtained by using Taguchi Method for DDOS attack detection.

Levels	HS	ME	L2	SR	SP	RMSE
1	18	200	0.0030	3	0.13	25.4501
2	18	300	0.0035	4	0.14	24.7128
3	18	400	0.0040	5	0.15	24.0935
.	.	.	.	.	.	.
.	.	.	.	.	.	.
24	22	500	0.0040	4	0.13	25.1571
25	22	600	0.0045	5	0.14	25.2895

TABLE 5: Autoencoder 1 S/N ratios obtained in the Taguchi experimental design for DDOS attack detection.

Levels	HS	ME	L2	SR	SP
1	-27.72	-27.72	-27.33	-27.59	-27.83
2	-27.54	-27.69	-27.28	-27.09	-27.33
3	-27.31	-27.31	-27.49	-27.54	-27.25
4	-26.91	-27.44	-27.49	-27.54	-27.25
5	-27.68	-27.51	-27.32	-27.34	-27.18
Delta	0.80	0.46	0.48	0.51	0.65
Rank	1	5	4	3	2

TABLE 6: Autoencoder 1 parameters set obtained through optimization for DDOS detection.

Factors	Value
HR	21
ME	200
L2	0.0035
SR	4
SP	0.17

TABLE 7: Autoencoder 2 upper and lower level values of factors for DDOS detection.

Factors	Lower limit	Upper limit
HS	18	22
ME	200	400
L2	0.003	0.005
SR	3	7
SP	0.13	0.17

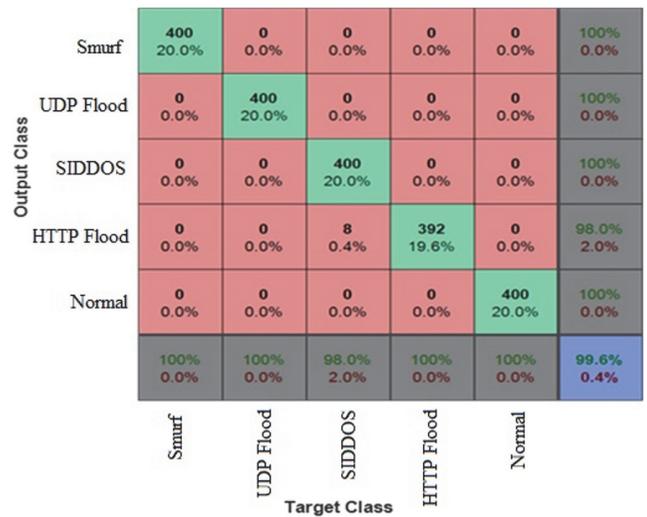


FIGURE 5: Confusion matrix for DDos detection results.

computer security systems, IDS is a prime part that can be classified as Host-based Intrusion Detection System (HIDS) which superheats a confirmed host or system and Network-based Intrusion detection system (NIDS), which superheats

a network of hosts and systems. In this paper, our framework is used to detect IDS attack by using new dataset [53], which consists of 47 features and 10 attack types. We will examine the UNSW-NB15 intrusion dataset in our research, as well

TABLE 8: Autoencoder 2 upper and lower level values of factors for DDOS detection.

Factors	Level 1	Level 2	Level 3	Level 4	Level 5
HS	10	11	12	13	14
ME	50	60	70	80	90
L2	0.0010	0.0015	0.0020	0.0025	0.0030
SR	3	4	5	6	7
SP	0.10	0.11	0.12	0.13	0.14

TABLE 9: Autoencoder 2 parameters values obtained by using Taguchi Method for DDOS attack detection.

Levels	HS	ME	L2	SR	SP	RMSE
1	10	50	0.0010	3	0.10	0.011309
2	10	60	0.0015	4	0.11	0.012012
3	10	70	0.0020	5	0.12	0.012019
.	.	.	.	.	.	.
24	14	80	0.0020	4	0.10	0.011987
25	14	90	0.0025	5	0.11	0.011979

TABLE 10: Autoencoder 2S/N ratios obtained in the Taguchi experimental design for DDOS attack detection.

Levels	HS	ME	L2	SR	SP
1	38.53	38.85	38.70	38.68	38.45
2	38.72	38.56	38.36	38.61	38.60
3	38.30	38.50	38.62	38.43	38.71
4	38.76	38.59	38.66	38.35	38.56
5	38.60	38.41	38.56	38.84	38.58
Delta	0.46	0.44	0.33	0.49	0.25
Rank	2	3	4	1	5

TABLE 11: Autoencoder 2 parameters set obtained through optimization for DDOS detection.

Factors	Value
HR	14
ME	50
L2	0.0010
SR	3
SP	0.13

TABLE 13: Autoencoder 1 parameters set obtained through optimization for IDS detection.

Factors	Value
HR	31
ME	250
L2	0.0040
SR	4
SP	0.16

TABLE 12: DDOS detection methods results comparison.

Methods	Accuracy %
MLP [23]	98.63
Random Forest [23]	98.02
Naïve Bayes [23]	96.91
SVM	97.29
SoftMax	93.14
SSAE-SVM [24]	97.65
<b>Proposed Framework</b>	<b>99.60</b>

as real-time captured dataset. This dataset is a hybrid of intrusion data collected from real modernistic normal and abnormal activities of the network traffic. This dataset is newer and more efficient than KDD98, KDDCUP99, and

NSLKDD which are the common and older features datasets because they were generated two decades ago. By following the same procedures in the Figure 1, and the tables such as in the DDoS detection procedures, the best parameters were determined as shown in the Tables 13 and 14 for each autoencoders 1 and 2 to find the best parameters that produces the best performance to detect IDS attacks. 10000 data points were used to train and test the system (5000 data used for training and 5000 for testing). Dividing half of the data for testing is also a challenging issue that previous studies employ more than %50 data for training. However, in order to reduce overall training time, training data percentage is pulled down. Experimental results for this dataset and configuration is illustrated in Figure 5. According to those results, the framework detection rate reaches 99.70% success

TABLE 14: Autoencoder 2 parameters set obtained through optimization for IDS detection.

Factors	Value
HR	14
ME	80
L2	0.0012
SR	4
SP	0.14

TABLE 15: A comparison amongst IDS detection methods results.

Methods	Accuracy %
DT [25]	85.56
LR [25]	83.15
NB [25]	82.07
ANN [25]	81.34
Ramp-KSVCR [25]	93.52
GA-LR [26]	81.42
SSAE-SVM [24]	84.71
SVM	83.16
SoftMax	80.13
<b>Proposed Framework</b>	<b>99.70</b>

rate which is satisfactory when compared with previous studies, as illustrated in Table 15. This proves that even such a small percentage training set is employed for this problem. Satisfactory results can be obtained. Figure 6 also demonstrates results based on the corresponding confusion matrix of the output results.

3.3. *Epileptic Seizure Recognition.* According to the latest results, 1-2% inhabitants of the world suffer from epilepsy which is a neurological trouble [54]. It is distinguished by surprised frequent and evanescent troubles of perception or demeaning our produce from immoderate coincidence of cortical neural networks. Epileptic Seizure is a neurologic status which is caused by detonation of electrical discharges in the brain. The epileptic seizures mean lineament of epilepsy is recurrent seizures. Observation of brain performance over the EEG has become a serious agent in the detection of epilepsy [55]. There are two kinds of abnormal actions: interictal, abnormal EEG recorded between epileptic crisis and ictal that occurs in the patient’s EEG records. The EEG subscription of an interictal action is accidental passing waveforms, as either separated trainer, sharp waves, or spike wave complexes [56]. Commonly, veteran physicians by visual surveying of EEG records for interictal and ictal actions can detect the epilepsy crises. However, visual survey of the huge size of EEG data has business-like disadvantages and weaknesses. Visual search is very time-consuming and inactive, essentially in the situation of long size of data [57]. In addition, contention among physicians on the many EEG results in some time leading to individual decision of the analysis due to the set of interictal spikes morphology. Therefore, computer-aided systems are developed to detect blood diseases [58], heart disease recognition [59], and epilepsy

TABLE 16: Autoencoder 1 parameters set obtained through optimization for Epileptic Seizure Recognition.

Factors	Value
HR	2004
ME	350
L2	0.0035
SR	5
SP	0.16

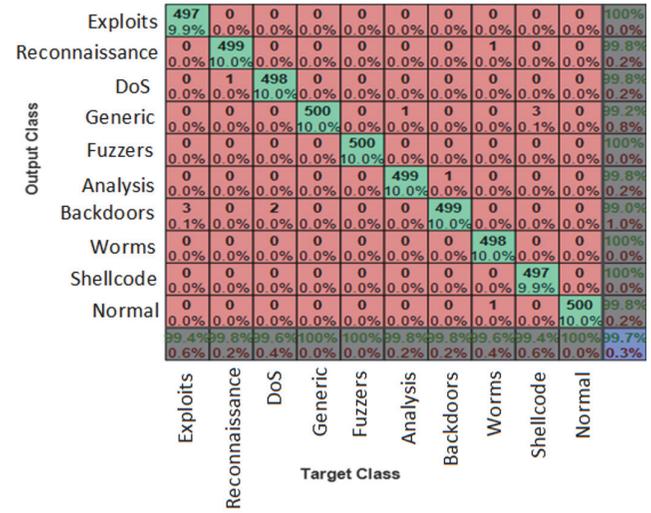


FIGURE 6: Confusion matrix for IDS detection results.

detection systems which are listed in Table 15. Epileptic dataset [60] is used to train and test in the proposed method. Two parts vector matrices are generated with the size of (100 × 4096) datasets, A representing (healthy) and E representing the (epileptic activity condition). A, E are divided into two parts, each of them is 50% of the vector matrices, and then two (50 × 4096) vector matrices are generated for training and another one for testing. Epileptic Seizure dataset consists of 4096 features by using 2 autoencoders, the first one reduces the number of features to 2004 and 103 in the second autoencoder which means reducing the time consumption. The best parameters for autoencoder 1 and autoencoder 2 that were obtained from our system are listed in Tables 16 and 17. This leads to obtaining the best results for Epileptic Seizure Recognition which is represented in Figure 7. The proposed method results compared with previous results in Epileptic Seizure Recognition are presented in Table 17. SVM, Nlp, and SoftMax were implemented by us to obtain results that are compared with our proposed method.

The comparison in Table 18 shows that there are a number of methods that have same accuracies with proposed method such as Tzallas et al. [28] and Srinivasan [30], but our proposed method has a good feature which uses deep learning techniques that give advantage when there are huge numbers of instances of epilepsy data for classification and uses only 50% of data in training when other methods used 60%.

TABLE 17: Autoencoder 2 parameters set obtained through optimization for Epileptic Seizure Recognition.

Factors	Value
HR	103
ME	160
L2	0.002
SR	4
SP	0.01

TABLE 18: A comparison of epileptic seizure recognition results.

Methods	Accuracy %
Srinivasan et al. [27]	99.60
Subasi and Ercelebi [28]	92.00
Subasi [29]	94.5
Kannathal et al [30]	92.22
Tzallas et al. [31]	100
Polat et al. [32]	98.72
Acharya et al. [33]	99.00
Acharya et al [34]	99.70
Musa Peker et al.[35]	100
SoftMax	87.13
SSAE-SVM [24]	98.80
SVM	92.09
MLP	94.11
<b>Proposed Framework</b>	<b>100</b>

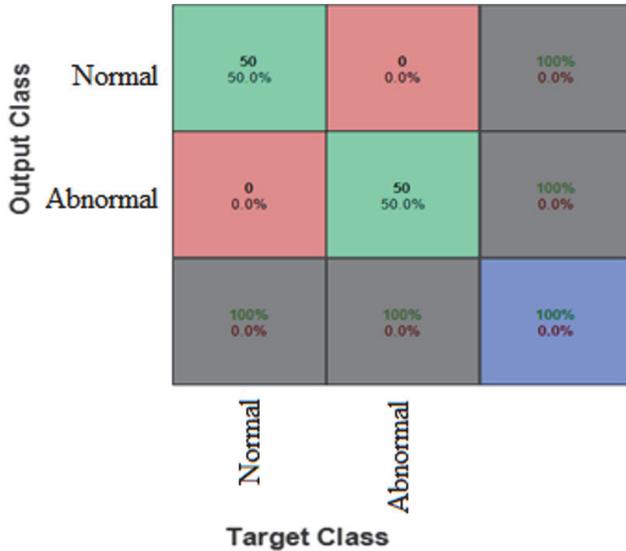


FIGURE 7: Confusion matrix for epileptic seizure recognition results.

3.4. *Handwritten Digit Classification.* The proposed framework is finally tested by employing MNSIT dataset which was proposed for handwritten digit classification problem [40]. The framework is trained by using “5000” images that is “500” for each example. Each image consists of “28x28” pixels, meaning there are “784” values for each image when

TABLE 19: Autoencoder 1 parameters set obtained through optimization for handwritten digit classification.

Factors	Value
HR	100
ME	400
L2	0.004
SR	4
SP	0.15

TABLE 20: Autoencoder 2 parameters set obtained through optimization for handwritten digit classification.

Factors	Value
HR	50
ME	100
L2	0.002
SR	4
SP	0.1

converted to vectors to build the matrices of vectors. In the second stage, the matrix of arrays becomes input to the first autoencoder in which parameters are also optimized by using Taguchi Method, as illustrated in Table 19. Besides Table 20 illustrates the optimized parameters for the second autoencoder. According to the characteristics of the proposed framework, extracted features from the second autoencoder are conveyed to the SoftMax layer that classify them into ten separate classes. Overall, the two autoencoders and SoftMax layer are stacked and trained in a supervised manner. The confusion matrix of the system obtained according to the experimental results is illustrated in Figure 8. These results are compared with the state-of-the-art studies regarding this problem and satisfactory results are obtained, as illustrated in Table 21.

#### 4. Conclusion

This paper proposes a new deep learning framework that essentially combines sparse autoencoder and Taguchi Method, which is an organized approach for parameter optimization in a reasonable amount of time. Experimental results reveal that applying this method allows the proposed framework to optimize numerous factors and extract more quantitative data from fewer experimental trials simultaneously. This novel framework is tested with different experimental data sets and compared to state-of-the-art methods and studies in terms of overall accuracy. For instance, proposed framework achieves satisfactory results: 99.6% in DDoS Detection, 99.7% for IDS Attack, 100% in Epileptic Seizure Recognition, and finally 99.8% precision result for handwritten digit classification problem. The results verify the validity of the proposed framework. Also authors are encouraged to improve overall performance of this architecture for more complex problems such as 3D image processing and real-time robotic system. Accordingly, different

TABLE 21: A comparison of handwritten digit classification results.

Reference	Methods	Accuracy %
Anupama Kaushik et al. [36]	J48	70.0
Anupama Kaushik et al. [36]	NaiveBayes	72.65
Anupama Kaushik et al. [36]	SMO	89.95
Olarik Surinta et al. [37]	Hotspot + SVM	92.70
U Ravi Babu et al. [38]	Hotspot + k-NN	96.94
Hinton GE et al. [39]	Deep Belief Network	98.75
LeCun Y et al. [40]	Deep Conv. Net LeNet-5	99.05
Wan L [41]	Deep Conv. Net (dropconnect)	99.43
Zelier MD [42]	Deep Conv. Net (stochastic pooling)	99.53
Goodfellow IJ [43]	Deep Conv. Net (maxout units and dropout)	99.55
Lee CY [44]	Deep Conv. Net (deeply-supervised)	99.61
<b>Proposed Framework</b>	<b>Deep Autoencoder based on Taguchi Method</b>	<b>99.80</b>

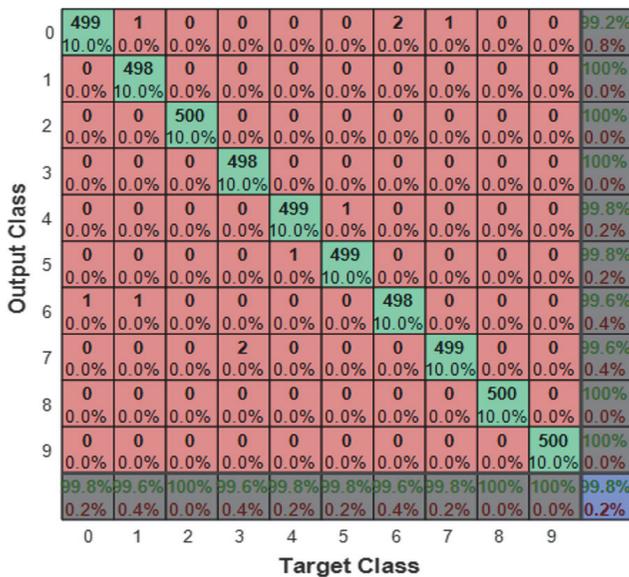


FIGURE 8: Confusion matrix for handwritten digit classification.

heuristic optimization algorithm, including genetic algorithms, particle swarm optimization, or colony optimization algorithms, will be used to estimate autoencoder parameters and compared with the Taguchi Method in future works. It is also noticed that the proposed architecture can also be employed for comprehensive recognition and estimation problems, including gesture recognition, URL reputation, and SMS spam collection.

**Data Availability**

The IDS attack data that support the findings of this study are available with “UNSW-NB15” reference name at “https://www.unsw.adfa.edu.au/australian-centre-for-cybersecurity/cybersecurity/ADFA-NB15-Datasets/”. Epilepsy recognition dataset that also support the findings of this study with “SETS

A and B” references is available at “http://epileptologiebonn.de/cms/front\_content.php?idcat=193&lang=3”. The Digit Classification dataset with “MNIST” reference is available at “http://http://yann.lecun.com/exdb/mnist/”. The DDoS detection dataset that support this study are available at “https://www.researchgate.net/publication/292967044\_Dataset-\_Detecting\_Distributed\_Denial\_of\_Service\_Attacks\_Using\_Data\_Mining\_Technique”.

**Conflicts of Interest**

The authors declare that there are no conflicts of interest regarding the publication of this paper.

**Acknowledgments**

The authors gratefully acknowledge the support to this work by Ankara Yıldırım Beyazıt University.

**References**

- [1] B. Chandra and R. K. Sharma, “Deep learning with adaptive learning rate using laplacian score,” *Expert Systems with Applications*, vol. 63, pp. 1–7, 2016.
- [2] S. Ding, N. Zhang, X. Xu, L. Guo, and J. Zhang, “Deep extreme learning machine and its application in EEG classification,” *Mathematical Problems in Engineering*, vol. 2015, Article ID 129021, 11 pages, 2015.
- [3] G. Urban et al., *Do Deep Convolutional Nets Really Need to be Deep and Convolutional?*, 2016.
- [4] T. orde, “Grozdića.pdf”.
- [5] K. G. Lore, A. Akintayo, and S. Sarkar, “LLNet: A deep autoencoder approach to natural low-light image enhancement,” *Pattern Recognition*, vol. 61, pp. 650–662, 2017.
- [6] K. Sun, J. Zhang, C. Zhang, and J. Hu, “Generalized extreme learning machine autoencoder and a new deep neural network,” *Neurocomputing*, vol. 230, pp. 374–381, 2017.
- [7] Y. Xiong and R. Zuo, “Recognition of geochemical anomalies using a deep autoencoder network,” *Computers & Geosciences*, vol. 86, pp. 75–82, 2016.

- [8] L. D. Burgoon, "Autoencoder Predicting Estrogenic Chemical Substances (APECS): An improved approach for screening potentially estrogenic chemicals using in vitro assays and deep learning," *Computational Toxicology*, vol. 2, pp. 45–49, 2017.
- [9] C. Hong, J. Yu, J. Wan, D. Tao, and M. Wang, "Multimodal Deep Autoencoder for Human Pose Recovery," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5659–5670, 2015.
- [10] T.-H. Song, V. Sanchez, H. Eidaly, and N. M. Rajpoot, "Hybrid deep autoencoder with Curvature Gaussian for detection of various types of cells in bone marrow trephine biopsy images," in *Proceedings of the 14th IEEE International Symposium on Biomedical Imaging, ISBI 2017*, pp. 1040–1043, aus, April 2017.
- [11] Y. Suzuki and T. Ozaki, "Stacked Denoising Autoencoder-Based Deep Collaborative Filtering Using the Change of Similarity," in *Proceedings of the 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 498–502, Taipei, Taiwan, March 2017.
- [12] Y. Zhang, X. Hou, Y. Lv, H. Chen, Y. Zhang, and S. Wang, "Sparse Autoencoder Based Deep Neural Network for Voxelwise Detection of Cerebral Microbleed," in *Proceedings of the 2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 1229–1232, Wuhan, December 2016.
- [13] S. Athreya and Y. D. Venkatesh, *Application Of Taguchi Method For Optimization Of Process Parameters In Improving The Surface Roughness Of Lathe Facing Operation*, vol. 1, no. 3, pp. 13–19, 2012.
- [14] M.-L. Huang, Y.-H. Hung, W. M. Lee, R. K. Li, and B.-R. Jiang, "SVM-RFE based feature selection and taguchi parameters optimization for multiclass SVM Classifier," *The Scientific World Journal*, vol. 2014, Article ID 795624, 2014.
- [15] D.-C. Ko, D.-H. Kim, and B.-M. Kim, "Application of artificial neural network and Taguchi method to preform design in metal forming considering workability," *The International Journal of Machine Tools and Manufacture*, vol. 39, no. 5, pp. 771–785, 1999.
- [16] H. Wang, Q. Geng, and Z. Qiao, "Parameter tuning of particle swarm optimization by using Taguchi method and its application to motor design," in *Proceedings of the 2014 4th IEEE International Conference on Information Science and Technology (ICIST)*, pp. 722–726, Shenzhen, China, April 2014.
- [17] G. K. Dhuria, R. Singh, and A. Batish, "Application of a hybrid Taguchi-entropy weight-based GRA method to optimize and neural network approach to predict the machining responses in ultrasonic machining of Ti-6Al-4V," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 39, no. 7, pp. 2619–2634, 2017.
- [18] H.-F. Yang, T. S. Dillon, and Y.-P. P. Chen, "Optimized Structure of the Traffic Flow Forecasting Model with a Deep Learning Approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2371–2381, 2017.
- [19] L. Wang, C. Wang, W. Du et al., "Parameter optimization of a four-legged robot to improve motion trajectory accuracy using signal-to-noise ratio theory," *Robotics and Computer-Integrated Manufacturing*, vol. 51, pp. 85–96, 2018.
- [20] R. Huang, C. Liu, G. Li, and J. Zhou, "Adaptive Deep Supervised Autoencoder Based Image Reconstruction for Face Recognition," *Mathematical Problems in Engineering*, vol. 2016, 2016.
- [21] D. Jha and G. Kwon, "Alzheimer's Disease Detection Using Sparse Autoencoder, Scale Conjugate Gradient and Softmax Output Layer with Fine Tuning," *International Journal of Machine Learning and Computing*, vol. 7, no. 1, pp. 13–17, 2017.
- [22] F. Cui, Y. Su, S. Xu, F. Liu, and G. Yao, "Optimization of the physical and mechanical properties of a spline surface fabricated by high-speed cold roll beating based on taguchi theory," *Mathematical Problems in Engineering*, vol. 2018, Article ID 8068362, pp. 1–12, 2018.
- [23] M. Alkasassbeh, G. Al-Naymat, A. B.A., and M. Almseidin, "Detecting Distributed Denial of Service Attacks Using Data Mining Techniques," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 1, 2016.
- [24] Y. Ju, J. Guo, and S. Liu, "A Deep Learning Method Combined Sparse Autoencoder with SVM," in *Proceedings of the 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 257–260, Xi'an, China, September 2015.
- [25] S. M. Hosseini Bamakan, H. Wang, and Y. Shi, "Ramp loss K-Support Vector Classification-Regression; a robust and sparse multi-class approach to the intrusion detection problem," *Knowledge-Based Systems*, vol. 126, pp. 113–126, 2017.
- [26] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," *Computers & Security*, vol. 70, pp. 255–277, 2017.
- [27] V. Srinivasan, C. Eswaran, and A. N. Sriraam, "Artificial neural network based epileptic detection using time-domain and frequency-domain features," *Journal of Medical Systems*, vol. 29, no. 6, pp. 647–660, 2005.
- [28] A. Subasi and E. Erçelebi, "Classification of EEG signals using neural network and logistic regression," *Computer Methods and Programs in Biomedicine*, vol. 78, no. 2, pp. 87–99, 2005.
- [29] A. Subasi, "EEG signal classification using wavelet feature extraction and a mixture of expert model," *Expert Systems with Applications*, vol. 32, no. 4, pp. 1084–1093, 2007.
- [30] N. Kannathal, M. L. Choo, U. R. Acharya, and P. K. Sadasivan, "Entropies for detection of epilepsy in EEG," *Computer Methods and Programs in Biomedicine*, vol. 80, no. 3, pp. 187–194, 2005.
- [31] A. T. Tzallas, M. G. Tsipouras, and D. I. Fotiadis, "Automatic seizure detection based on time-frequency analysis and artificial neural networks," *Computational Intelligence and Neuroscience*, vol. 2007, Article ID 80510, 13 pages, 2007.
- [32] K. Polat and S. Güneş, "Classification of epileptiform EEG using a hybrid system based on decision tree classifier and fast Fourier transform," *Applied Mathematics and Computation*, vol. 187, no. 2, pp. 1017–1026, 2007.
- [33] U. Rajendra Acharya, S. Vinitha Sree, A. P. C. Alvin, and J. S. Suri, "Use of principal component analysis for automatic classification of epileptic EEG activities in wavelet framework," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9072–9078, 2012.
- [34] U. R. Acharya, S. V. Sree, P. C. A. Ang, R. Yanti, and J. S. Suri, "Application of non-linear and wavelet based features for the automated identification of epileptic EEG signals," *International Journal of Neural Systems*, vol. 22, no. 2, Article ID 1250002, pp. 565–579, 2012.
- [35] M. Peker, B. Sen, and D. Delen, "A novel method for automated diagnosis of epilepsy using complex-valued classifiers," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 1, pp. 108–118, 2016.
- [36] A. Kaushik, H. Gupta, and D. S. Latwal, "Impact of Feature Selection and Engineering in the Classification of Handwritten Text," pp. 2598–2601, 2016.
- [37] L. Schomaker and M. A. Wiering, "Handwritten Character Classification Using The Hotspot Feature Extraction Technique," in *Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods*, pp. 261–264, Vilamoura, Algarve, Portugal, February 2012.

- [38] U. R. Babu, Y. Venkateswarlu, and A. K. Chintha, "Handwritten digit recognition using K-nearest neighbour classifier," in *Proceedings of the World Congress on Computing and Communication Technologies (WCCCT '14)*, pp. 60–65, Trichirappalli, India, March 2014.
- [39] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, in *Neural Computation*, vol. 18, pp. 1527–1554, 2006.
- [40] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.
- [41] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus, "Regularization of neural networks using DropConnect," in *Proceedings of the 30th International Conference on Machine Learning*, vol. 28, Atlanta, Georgia, USA, 2013.
- [42] M. F. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep CNN," in *Proceedings of the In proc. International Conference on Learning Representations*, Scottsdale, USA, 2013.
- [43] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, Mass, USA, 2016.
- [44] C. Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," *Deep Learning and Representation Learning Workshop*, 2014.
- [45] Y. Yan, Z. Tan, N. Su, and C. Zhao, "Building extraction based on an optimized stacked sparse autoencoder of structure and training samples using LIDAR DSM and optical images," *Sensors*, vol. 17, no. 9, 2017.
- [46] Q.-C. Hsu and A. T. Do, "Minimum porosity formation in pressure die casting by taguchi method," *Mathematical Problems in Engineering*, vol. 2013, 2013.
- [47] M. Peker, "A new approach for automatic sleep scoring: Combining Taguchi based complex-valued neural network and complex wavelet transform," *Computer Methods and Programs in Biomedicine*, vol. 129, pp. 203–216, 2016.
- [48] M. Peker, B. Şen, and P. Y. Kumru, "An efficient solving of the traveling salesman problem: The ant colony system having parameters optimized by the Taguchi method," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 21, no. 1, pp. 2015–2036, 2013.
- [49] M. Nandhini, B. Suchithra, R. Saravana-thamizhan, and D. G. Prakash, "Optimization of parameters for dye removal by electro-oxidation using Taguchi Design," *Journal of Electrochemical Science and Engineering*, vol. 4, no. 4, 2014.
- [50] L. Ivanović, B. Stojanović, J. Blagojević, G. Bogdanović, and A. Marinković, "Analysis of the flow rate and the volumetric efficiency of the trochoidal pump by application of taguchi method," *Tehnički vjesnik*, vol. 24, pp. 265–270, 2017.
- [51] N. Hoque, H. Kashyap, and D. K. Bhattacharyya, "Real-time DDoS attack detection using FPGA," *Computer Communications*, vol. 110, pp. 48–58, 2017.
- [52] B. B. Gupta, "Predicting number of zombies in DDoS attacks using pace regression model," *Journal of Computing and Information Technology*, vol. 20, no. 1, pp. 33–39, 2012.
- [53] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proceedings of the Military Communications and Information Systems Conference*, pp. 1–6, 2015.
- [54] S. Yücel, P. Terzioğlu, and D. Özçimen, "World™'s largest Science, Technology & Medicine Open Access book publisher c," *RFID Technol. Secur. Vulnerabilities, Countermeas*, 2016.
- [55] F. Mormann, R. G. Andrzejak, C. E. Elger, and K. Lehnertz, "Seizure prediction: the long and winding road," *Brain*, vol. 130, no. 2, pp. 314–333, 2007.
- [56] P. Evidence, B. Eb, and P. Neprovociranom, "Evidence Base (Eb) Approach to The First Unprovoked".
- [57] C. J. James and B. E. Eng, *Detection of epileptiform activity in the electroencephalogram using artificial neural networks*, 1997.
- [58] A. M. Karim, F. V. Çelebi, and A. S. Mohammed, "Software Development for Blood Disease Expert System," *Lecture Notes on Empirical Software Engineering*, vol. 4, no. 3, pp. 179–183, 2016.
- [59] A. Khemphila and V. Boonjing, "Heart disease classification using neural network and feature selection," in *Proceedings of the 21st International Conference on Systems Engineering (ICSEng '11)*, pp. 406–409, IEEE Computer Society, Las Vegas, Nev, USA, August 2011.
- [60] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: dependence on recording region and brain state," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 64, no. 6, Article ID 061907, 8 pages, 2001.

## Research Article

# An Objective Penalty Function-Based Method for Inequality Constrained Minimization Problem

Shujun Lian , Sitong Meng, and Yiju Wang 

School of Management Science, Qufu Normal University, Rizhao, Shandong 276800, China

Correspondence should be addressed to Shujun Lian; [lsjsd2003@126.com](mailto:lsjsd2003@126.com)

Received 15 March 2018; Accepted 10 May 2018; Published 6 June 2018

Academic Editor: Nivaldo Rodríguez

Copyright © 2018 Shujun Lian et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For inequality constrained minimization problem, we first propose a new exact nonsmooth objective penalty function and then apply a smooth technique to the penalty function to make it smooth. It is shown that any minimizer of the smoothing objective penalty function is an approximated solution of the original problem. Based on this, we develop a solution method for the inequality constrained minimization problem and prove its global convergence. Numerical experiments are provided to show the efficiency of the proposed method.

## 1. Introduction

Consider the following inequality constrained minimization problem:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0, \quad i \in I = \{1, 2, \dots, m\} \end{aligned} \quad (P)$$

where  $f, g_i : R^n \rightarrow R, i \in I$  are continuously differentiable functions. Throughout this paper, we use  $X = \{x \in R^n \mid g_i(x) \leq 0, i \in I\}$  to denote the feasible solution set.

The problem finds applications in fields such as economics, mathematical programming, transportation, and regional science [1–5], and it has received much attention from researchers; see, e.g., [6–16].

Due to the involvement of the inequality constraint in the problem, it is very hard to solve directly. Hence, some researchers turn to indirect methods such as the penalty function method, the SQP method, and the feasible direction method [17]. Among these methods, the penalty function method is a popular one. Its main idea is to combine the objective function and constraints into a penalty function and then attack problem (P) by solving a sequence of unconstrained problems. Generally, if the solution of the original problem is the solution of the penalty problem or the solution of the penalty problem is the solution of the original problem, then the penalty function is called exact [17]. For

this, Zangwill [18] proposed the following classical  $l_1$  exact penalty function:

$$F(x, \rho) = f(x) + \rho \sum_{i \in I} g_i^+(x), \quad (1)$$

where  $\rho > 0$  is a penalty parameter and  $g_i^+(x) = \max\{g_i(x), 0\}, i \in I$ .

Obviously, the penalty function given above is not smooth and the researchers considered its smoothing version [19–25]. In [26], the lower-order penalty function,

$$\varphi_{\rho,k}(x) = f(x) + \rho \sum_{i=1}^m (\max\{g_i(x), 0\})^k, \quad k \in (0, 1), \quad (2)$$

was introduced and its exact property and its smoothing were investigated [27, 28].

To improve the performance of the penalty function when solving the inequality constrained optimization problem, the following objective penalty function is introduced [29, 30]:

$$F_1(x, M) = (f(x) - M)^p + \sum_{i \in I} g_i(x)^p, \quad (3)$$

where  $M \in R$  is an objective penalty parameter and  $p > 0$ . Assume that  $x^*$  is an optimal solution and  $f_0(x^*)$  is the optimal objective function value of the original problem (P).

For this function, it is shown that the minimizers  $x(M^k)$  of the problem  $\min F_1(x, M^k)$  tend to  $x^*$  when a convergent sequence  $M^k$  tends to  $f_0(x^*)$ .

Later, Meng et al. [31] considered the following objective penalty function:

$$F_2(x, M) = (f(x) - M)^2 + \sum_{i \in I} g_i^+(x)^p, \quad (4)$$

where  $p > 1$  and  $M \in R$  is an objective penalty parameter. The objective penalty function is smooth and its exact property was proved for the objective penalty function.

Li et al. [32] proposed the following objective penalty function for solving minmax programming problems with equality and inequality constraints:

$$E(x, t; M, \rho) = \frac{1}{2} [(t - M)^+]^2 + \frac{\rho}{2} \left\{ \sum_{i \in I} [(f_i(x) - t)^+]^2 + \sum_{i \in J} [g_j^+(x)]^2 + \sum_{i \in L} h_i^2(x) \right\} \quad (5)$$

by combining the objective penalty and constraint penalty.

In this paper, we will propose a new exact nonsmooth objective penalty function which is different from the functions defined by (3) and (4). Then motivated by the smoothing technique of the  $l_1$  exact penalty function in [23–25], we make a smoothing to the nonsmooth objective penalty function so that the nonsmooth objective penalty function can be numerically minimized by methods such as gradient-type method or Newton-type method.

The remainder of this paper is organized as follows. In Section 2, we propose a new exact nonsmooth objective penalty function and then make a second-order smoothing approximation to it. Error bound estimations among the optimal objective values of the nonsmooth objective penalty problem and the smoothed objective penalty problem are presented. Based on the second-order differentiable smoothing objective penalty function, we develop a solution method in Section 3 and prove its global convergence. Some numerical experiments are made in Section 4 to show the efficiency of the proposed method.

## 2. A New Objective Penalty Function and Its Smoothing

In this section, we consider the following objective penalty function:

$$G(x, M) = (f(x) - M)^2 + \sum_{i \in I} g_i^+(x). \quad (6)$$

Correspondingly, the associated optimization problem is as follows:

$$\begin{aligned} \min \quad & G(x, M) \\ \text{s.t.} \quad & x \in Y \end{aligned} \quad (P_M)$$

where  $X \subset Y \subset R^n$ .

For this problem, we have the following conclusion on the relationship between the optimal solution of  $(P_M)$  and  $(P)$ .

**Theorem 1.** *If  $x^*$  is an optimal solution of problem  $(P)$ , then  $x^*$  is also an optimal solution of problem  $(P_M)$  with  $M = f(x^*)$ .*

*Proof.* Since  $x^*$  is an optimal solution to  $(P)$  and  $M = f(x^*)$ , it holds that

$$G(x^*, M) = (f(x^*) - M)^2 + \sum_{i \in I} g_i^+(x^*) = 0. \quad (7)$$

It is easy to see that  $G(x, M) \geq 0$  for any  $x \in R^n$ . Hence  $x^*$  is an optimal solution to  $(P_M)$ .  $\square$

**Theorem 2.** *Let  $X$  be a connected and compact set and  $f : R^n \rightarrow R$  be a continuous function. Set  $M_* = \min_{x \in X} f(x)$  and  $M^* = \max_{x \in X} f(x)$ . Suppose  $x_M^*$  is an optimal solution to  $(P_M)$  for some  $M$ . Then*

(i) *if  $G(x_M^*, M) = 0$ , then  $x_M^*$  is a feasible solution to  $(P)$  and  $M_* \leq M \leq M^*$ ;*

(ii) *if  $G(x_M^*, M) > 0$  and  $M \leq M^*$ , then  $M < M_*$ .*

*Proof.* (i) It follows from  $G(x_M^*, M) = 0$  that  $g_i(x_M^*) \leq 0$ ,  $\forall i \in I$ , and  $f(x_M^*) - M = 0$ , so  $M_* \leq M = f(x_M^*) \leq M^*$ . The conclusion is proved.

(ii) If  $M_* \leq M$ , then  $M_* \leq M \leq M^*$ . Since  $f$  is continuous, there exists  $x \in X$  such that  $M = f(x)$ . Hence  $G(x, M) = 0$ . On the other hand, since  $x_M^*$  is optimal to  $(P_M)$ , it holds that  $G(x_M^*, M) \leq G(x, M) = 0$ , which is contradict with  $G(x_M^*, M) > 0$ . Therefore,  $M < M_*$ .  $\square$

**Theorem 3.** *Let  $X$  be a connected and compact set,  $f : R^n \rightarrow R$  be a continuous function,  $M_* = \min_{x \in X} f(x)$  and  $M^* = \max_{x \in X} f(x)$ , and  $x^*$  be an optimal solution to  $(P)$ . Suppose  $x_M^*$  is an optimal solution to  $(P_M)$  for some  $M$ ,  $G(x_M^*, M) > 0$ , and  $M \leq M^*$ . Then*

(i) *if  $x_M^*$  is not feasible to  $(P)$ , then  $M < M_*$  and  $f(x_M^*) < M_*$ ;*

(ii) *if  $x_M^*$  is a feasible solution to  $(P)$ , then  $x_M^*$  is an optimal solution to  $(P)$  and  $M$  is an exact value of the objective penalty parameter.*

*Proof.* (i) By (ii) in Theorem 2,  $M < f(x^*) = M_*$ . If  $f(x_M^*) \leq M$ , then  $f(x_M^*) \leq M < M_*$ . On the other hand, if  $f(x_M^*) > M$ , and from (6), one has

$$\begin{aligned} 0 < (f(x_M^*) - M)^2 &< G(x_M^*, M) \leq G(x^*, M) \\ &= (f(x^*) - M)^2. \end{aligned} \quad (8)$$

Since  $f(x_M^*) - M > 0$  and  $f(x^*) - M > 0$ , it follows that  $f(x_M^*) - M < f(x^*) - M$ . Hence,  $f(x_M^*) < f(x^*) = M_*$ .

(ii) It follows from the assumption and (6) that

$$\begin{aligned} 0 < (f(x_M^*) - M)^2 &= G(x_M^*, M) \leq G(x, M) \\ &= (f(x) - M)^2, \quad \forall x \in X. \end{aligned} \quad (9)$$

Since  $x_M^*$  is feasible to  $(P)$ , by (ii) in Theorem 2, one has  $M < M_* < f(x_M^*)$  and  $M < M_* < f(x)$ ,  $\forall x \in X$ . Hence,  $f(x_M^*) - M > 0$  and  $f(x) - M > 0$ . Then,

$$f(x_M^*) - M \leq f(x) - M, \quad \forall x \in X. \quad (10)$$

Therefore,

$$f(x_M^*) \leq f(x), \quad \forall x \in X. \quad (11)$$

This means that  $x_M^*$  is an optimal solution to  $(P)$ .  $\square$

Theorems 2 and 3 provide a way to solve problem  $(P)$ . However, the objective penalty function  $G(x, M)$  is not smooth. Now, we use a smoothing technique to make it twice continuously differentiable which can be minimized by methods such as Newton-type method. The obtained smooth objective penalty function is much different from the functions given in [23–25].

Let  $p(t) = \max\{t, 0\}$ , and define

$$p_\varepsilon(t) = \begin{cases} \frac{1}{4}\varepsilon e^{2t/\varepsilon}, & t \leq 0, \\ t + \frac{1}{4}\varepsilon e^{-2t/\varepsilon}, & t > 0. \end{cases} \quad (12)$$

Then

$$p'_\varepsilon(t) = \begin{cases} \frac{1}{2}e^{2t/\varepsilon}, & t \leq 0, \\ 1 - \frac{1}{2}e^{-2t/\varepsilon}, & t > 0, \end{cases} \quad (13)$$

and

$$p''_\varepsilon(t) = \begin{cases} -\frac{1}{\varepsilon}e^{2t/\varepsilon}, & t \leq 0, \\ \frac{1}{\varepsilon}e^{-2t/\varepsilon}, & t > 0. \end{cases} \quad (14)$$

It is easy to see that function  $p_\varepsilon(t)$  is twice continuously differentiable on  $R$  and

$$\lim_{\varepsilon \rightarrow 0^+} p_\varepsilon(t) = p(t). \quad (15)$$

Based on this, we consider the following second-order smoothing approximation:

$$G(x, M, \varepsilon) = (f(x) - M)^2 + \sum_{i \in I} p_\varepsilon(g_i(x)), \quad (16)$$

where  $\lim_{\varepsilon \rightarrow 0^+} G(x, M, \varepsilon) = G(x, M)$ .

The corresponding optimization problem to  $G(x, M, \varepsilon)$  is as follows:

$$\begin{aligned} \min \quad & G(x, M, \varepsilon) \\ \text{s.t.} \quad & x \in Y. \end{aligned} \quad (P'_M)$$

For problems  $(P_M)$  and  $(P'_M)$ , we have the following conclusion.

**Lemma 4.** For any  $x \in R^n$  and  $\varepsilon > 0$ , it holds that

$$0 \leq G(x, M, \varepsilon) - G(x, M) \leq \frac{1}{4}m\varepsilon. \quad (17)$$

*Proof.* From the definition of  $p(t)$  and  $p_\varepsilon(t)$ , one has

$$p_\varepsilon(t) - p(t) = \begin{cases} \frac{1}{4}\varepsilon e^{2t/\varepsilon}, & t \leq 0, \\ \frac{1}{4}\varepsilon e^{-2t/\varepsilon}, & t > 0. \end{cases} \quad (18)$$

Hence,

$$0 \leq p_\varepsilon(t) - p(t) \leq \frac{1}{4}\varepsilon. \quad (19)$$

Thus, for any  $x \in R^n$ , it holds that

$$0 \leq p_\varepsilon(g_i(x)) - p(g_i(x)) \leq \frac{1}{4}\varepsilon, \quad \forall i \in I, \quad (20)$$

which means that

$$0 \leq \sum_{i \in I} p_\varepsilon(g_i(x)) - \sum_{i \in I} p(g_i(x)) \leq \frac{1}{4}m\varepsilon. \quad (21)$$

It follows from (6) and (16) that

$$0 \leq G(x, M, \varepsilon) - G(x, M) \leq \frac{1}{4}m\varepsilon. \quad (22)$$

$\square$

**Theorem 5.** Suppose positive sequence  $\{\varepsilon_j\}$  converges to 0 as  $j \rightarrow \infty$ ,  $x^j$  is a solution to  $\min_{x \in Y} G(x, M, \varepsilon_j)$ , and  $\bar{x}$  is an accumulating point of sequence  $\{x^j\}$ . Then  $\bar{x}$  is an optimal solution to  $\min_{x \in Y} G(x, M)$ .

*Proof.* Since  $x^j$  is a solution to  $\min_{x \in Y} G(x, M, \varepsilon_j)$ , one has

$$G(x^j, M, \varepsilon_j) \leq G(x, M, \varepsilon_j). \quad (23)$$

It follows from Lemma 4 that

$$G(x^j, M) \leq G(x^j, M, \varepsilon_j), \quad (24)$$

and

$$G(x, M, \varepsilon_j) \leq G(x, M) + \frac{1}{4}m\varepsilon_j. \quad (25)$$

From (23), (24), and (25), one has

$$\begin{aligned} G(x^j, M) &\leq G(x^j, M, \varepsilon_j) \leq G(x, M, \varepsilon_j) \\ &\leq G(x, M) + \frac{1}{4}m\varepsilon_j. \end{aligned} \quad (26)$$

Letting  $j \rightarrow \infty$  yields

$$G(\bar{x}, M) \leq G(x, M). \quad (27)$$

Thus  $\bar{x}$  is an optimal solution to  $\min_{x \in Y} G(x, M)$ .  $\square$

**Theorem 6.** Let  $x^*$  be an optimal solution of  $(P_M)$  and  $\bar{x}$  be an optimal solution of  $(P'_M)$ . Then

$$0 \leq G(\bar{x}, M, \varepsilon) - G(x^*, M) \leq \frac{1}{4}m\varepsilon. \quad (28)$$

*Proof.* By Lemma 4 and the assumption, one has

$$\begin{aligned} 0 &\leq G(\bar{x}, M, \varepsilon) - G(\bar{x}, M) \\ &\leq G(\bar{x}, M, \varepsilon) - G(x^*, M), \end{aligned} \quad (29)$$

and

$$\begin{aligned} G(\bar{x}, M, \varepsilon) - G(x^*, M) &\leq G(x^*, M, \varepsilon) - G(x^*, M) \\ &\leq \frac{1}{4}m\varepsilon. \end{aligned} \quad (30)$$

Then

$$0 \leq G(\bar{x}, M, \varepsilon) - G(x^*, M) \leq \frac{1}{4}m\varepsilon. \quad (31)$$

□

Theorem 6 means that the optimal solution to  $(P'_M)$  is also an approximately optimal solution to  $(P_M)$  when  $\varepsilon$  is sufficiently small.

### 3. A Smoothing Method

In this section, we will propose an algorithm for solving problem  $(P)$  based on the smoothed objective penalty function  $G(x, M, \varepsilon)$ . The following algorithm is based on the relationship between  $M$  and  $M_*$  given in Theorems 2 and 3.

*Algorithm 7.*

*Step 1.* Take  $x^0 \in R^n$ ,  $\varepsilon_1 > 0$ ,  $0 < \eta < 1$ ,  $a_1 < \min_{x \in X} f(x)$  and  $\min_{x \in X} f(x) < b_1 < \max_{x \in X} f(x)$ . Let  $M_1 = (a_1 + b_1)/2$ ,  $j = 1$ .

*Step 2.* Solve  $\min_{x \in Y} G(x, M_j, \varepsilon_j)$  starting at  $x^{j-1}$ . Let  $x^j$  be the global optimal solution. ( $x^j$  is obtained by a quasi-Newton method.)

*Step 3.* If  $G(x^j, M_j) = 0$ , let  $a_{j+1} = a_j$ ,  $b_{j+1} = M_j$ ,  $M_{j+1} = (a_{j+1} + b_{j+1})/2$ ,  $\varepsilon_{j+1} = \eta\varepsilon_j$ ,  $j = j + 1$  and go to Step 2.

*Step 4.* If  $x^j$  is not feasible to  $(P)$ , let  $b_{j+1} = b_j$ ,  $a_{j+1} = M_j$ ,  $M_{j+1} = (a_{j+1} + b_{j+1})/2$ ,  $\varepsilon_{j+1} = \eta\varepsilon_j$ ,  $j = j + 1$  and go to Step 2. Otherwise, if  $x^j$  is feasible to  $(P)$ ,  $x^j$  is the approximate optimal solution to  $(P)$ .

For Algorithm 7, we always assume that  $a_1 < \min_{x \in X} f(x)$  and  $\min_{x \in X} f(x) < b_1 < \max_{x \in X} f(x)$  can be satisfied. Under this condition, we can establish the global convergence of Algorithm 7.

**Theorem 8.** Suppose that  $\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty$  and  $\{x^j\}$  is an infinite sequence generated by Algorithm 7. Then the following hold:

(1) If the algorithm terminates at step  $\bar{j}$ , then  $x^{\bar{j}}$  is an optimal solution to  $(P)$ .

(2) If the algorithm generates an infinite sequence  $\{x^j\}$ , then it is bounded and its any limit point  $x^*$  is an optimal solution to  $(P)$ .

*Proof.* First, we claim that sequences  $\{a_j\}$  and  $\{b_j\}$  defined in Algorithm 7 are such that  $\{a_j\}$  is an increasing sequence and  $\{b_j\}$  is a decreasing sequence with

$$a_j \leq M_j \leq b_j, \quad j = 1, 2, \dots \quad (32)$$

and

$$b_{j+1} - a_{j+1} = \frac{b_j - a_j}{2}, \quad j = 1, 2, \dots \quad (33)$$

The following proof is by induction.

For  $j = 1$ , it follows from Algorithm 7 that  $a_1 \leq M_1 = (a_1 + b_1)/2 \leq b_1$ ,  $b_2 - a_2 = (b_1 - a_1)/2$ . For the induction step, let the hypothesis hold for  $j-1$ . For  $j$ , we let  $a_j = a_{j-1}$ ,  $b_j = M_{j-1}$ , and  $M_j = (a_j + b_j)/2$  in Step 3. By  $a_{j-1} \leq M_{j-1} \leq b_{j-1}$ , one has

$$\begin{aligned} a_j = a_{j-1} &= \frac{a_{j-1} + a_{j-1}}{2} \leq M_j = \frac{a_{j-1} + M_{j-1}}{2} \\ &\leq \frac{M_{j-1} + M_{j-1}}{2} = M_{j-1} = b_j. \end{aligned} \quad (34)$$

In Step 4, let  $b_j = b_{j-1}$ ,  $a_j = M_{j-1}$ , and  $M_j = (a_j + b_j)/2$ . By  $a_{j-1} \leq M_{j-1} \leq b_{j-1}$ , one has

$$\begin{aligned} a_j = M_{j-1} &= \frac{M_{j-1} + M_{j-1}}{2} \leq M_j = \frac{M_{j-1} + b_{j-1}}{2} \\ &\leq \frac{b_{j-1} + b_{j-1}}{2} = b_{j-1} = b_j. \end{aligned} \quad (35)$$

By induction, (32) holds for all  $j$ .

Consider the next iteration.

In Step 3, let  $a_{j+1} = a_j$ ,  $b_{j+1} = M_j$ , then  $b_{j+1} - a_{j+1} = M_j - a_j = (b_j - a_j)/2$ .

In Step 4, let  $b_{j+1} = b_j$ ,  $a_{j+1} = M_j$ , then  $b_{j+1} - a_{j+1} = b_j - M_j = (b_j - a_j)/2$ .

By induction, (33) holds for all  $j$ .

From Algorithm 7, it is easy to see that  $\{a_j\}$  is increasing and  $\{b_j\}$  is decreasing. Then sequences  $\{a_j\}$  and  $\{b_j\}$  are both convergent. Let  $a_j \rightarrow a^*$  and  $b_j \rightarrow b^*$ . It follows from (32) and (33) that  $a^* = b^*$ . Therefore,  $\{M_j\}$  also converges to  $a^*$ .

Now, we are at the position to prove the main conclusion in the section.

For (1), if Algorithm 7 terminates at the  $\bar{j}$ th iteration, it must terminate at Step 4;  $x^{\bar{j}}$  is feasible to  $(P)$ . By Theorem 3,  $x^{\bar{j}}$  is an optimal solution to  $(P)$ .

For (2), we first show that the sequence  $\{x^j\}$  is bounded. For the sake of contradiction, suppose that the sequence  $\{x^j\}$  is unbounded.

Since  $x^j$  is an optimal solution to  $\min_{x \in Y} G(x, M_j, \varepsilon_j)$ , for any fixed  $\bar{x} \in X$ ,

$$\begin{aligned} G(x^j, M_j, \varepsilon_j) &\leq G(\bar{x}, M_j, \varepsilon_j) \\ &= (f(\bar{x}) - M_j)^2 + \sum_{i \in I} \frac{1}{4} \varepsilon_j e^{2g_i(\bar{x})/\varepsilon_j}, \end{aligned} \quad (36)$$

$$j = 1, 2, \dots$$

TABLE 1: Numerical results of Algorithm 7 on Example 11 with different starting points.

$j$	$x^0$	$x^j$	$g_1(x^j)$	$g_2(x^j)$	$g_3(x^j)$	$f(x^j)$
1	(-10, -10, -10, -10)	(0.16893, 0.83692, 2.0083, -0.96546)	$-5.3237e - 12$	$-2.0108e - 12$	-1.8767	-44.2338
1	(10, 10, -10, -10)	(0.1691, 0.83694, 2.0082, -0.9555)	$-1.3398e - 10$	$-5.0255e - 11$	-1.8767	-44.2338
1	(-10, -10, 10, 10)	(0.16924, 0.83683, 2.0082, -0.96555)	$-1.7465e - 10$	$-5.0245e - 11$	-1.8772	-44.2338
1	(10, 10, 10, 10)	(0.16903, 0.83701, 2.0082, -0.96556)	$-1.339e - 10$	$-5.0243e - 11$	-1.8763	-44.2338
1	(5, 5, 5, 5)	(0.16899, 0.83713, 2.0081, -0.96562)	$-2.5527e - 11$	$-9.9867e - 12$	-1.8758	-44.2338
1	(20, 20, 20, 20)	(0.16897, 0.83705, 2.0082, -0.96557)	$-2.6782e - 11$	$-1.0057e - 11$	-1.8761	-44.2338

Due to  $M_j \rightarrow a^*$  and  $\varepsilon_j \rightarrow 0$  as  $j \rightarrow \infty$ , we conclude that there is some  $L > 0$  such that

$$L > G(x^j, M_j, \varepsilon_j) \geq (f(x^j) - M_j)^2, \quad j = 1, 2, \dots \quad (37)$$

Since  $\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty$ , we arrive at a contradiction, which shows that the sequence  $\{x^j\}$  is bounded.

Let  $M_* = \min_{x \in X} f(x)$ . Without loss of generality, we assume  $x^j \rightarrow x^*$  as  $j \rightarrow \infty$ . By Theorems 2 and 3 and Algorithm 7, we know that  $a_j < M_* \leq b_j$ . It follows from (32) that  $a^* = M_*$ . Let  $y^*$  be an optimal solution to (P). Then  $M_* = f(y^*)$ . Note that

$$\begin{aligned} G(x^j, M_j, \varepsilon_j) &\leq G(y^*, M_j, \varepsilon_j) \\ &= (f(y^*) - M_j)^2 + \sum_{i \in I} \frac{1}{4} \varepsilon_j e^{2g_i(y^*)/\varepsilon_j}. \end{aligned} \quad (38)$$

Letting  $j \rightarrow \infty$  yields that

$$G(x^*, M_*) \leq 0, \quad (39)$$

which implies  $x^* \in X$  and  $M_* = f(x^*)$ . Therefore,  $x^*$  is an optimal solution to (P).  $\square$

#### 4. Numerical Experiments

In this section, we will make some numerical experiments to show the efficiency of Algorithm 7. Based on the different objective penalty functions, we give different algorithms to make a comparison. The algorithms based on the objective functions (6) or (4) are described below.

*Algorithm 9.*

*Step 1.* Take  $x^0 \in R^n$ ,  $a_1 < \min_{x \in X} f(x)$ , and  $\min_{x \in X} f(x) < b_1 < \max_{x \in X} f(x)$ . Let  $M_1 = (a_1 + b_1)/2$ ,  $j = 1$ .

*Step 2.* Solve  $\min_{x \in Y} G(x, M_j)$  starting at  $x^{j-1}$ . Let  $x^j$  be the global optimal solution.

*Step 3.* If  $G(x^j, M_j) = 0$ , let  $a_{j+1} = a_j$ ,  $b_{j+1} = M_j$ ,  $M_{j+1} = (a_{j+1} + b_{j+1})/2$ ,  $j = j + 1$  and go to Step 2.

*Step 4.* If  $x^j$  is not feasible to (P), let  $b_{j+1} = b_j$ ,  $a_{j+1} = M_j$ ,  $M_{j+1} = (a_{j+1} + b_{j+1})/2$ ,  $j = j + 1$  and go to Step 2. Otherwise, if  $x^j$  is feasible to (P),  $x^j$  is the approximate optimal solution to (P).

*Algorithm 10.*

*Step 1.* Take  $x^0 \in R^n$ ,  $a_1 < \min_{x \in X} f(x)$ , and  $\min_{x \in X} f(x) < b_1 < \max_{x \in X} f(x)$ . Let  $M_1 = (a_1 + b_1)/2$ ,  $j = 1$ .

*Step 2.* Solve  $\min_{x \in Y} F_2(x, M_j)$  with  $p = 1.5$  starting at  $x^{j-1}$ . Let  $x^j$  be the global optimal solution.

*Step 3.* If  $F_2(x^j, M_j) = 0$ , let  $a_{j+1} = a_j$ ,  $b_{j+1} = M_j$ ,  $M_{j+1} = (a_{j+1} + b_{j+1})/2$ ,  $j = j + 1$  and go to Step 2.

*Step 4.* If  $x^j$  is not feasible to (P), let  $b_{j+1} = b_j$ ,  $a_{j+1} = M_j$ ,  $M_{j+1} = (a_{j+1} + b_{j+1})/2$ ,  $j = j + 1$  and go to Step 2. Otherwise, if  $x^j$  is feasible to (P),  $x^j$  is the approximate optimal solution to (P).

*Example 11.* Consider the following problem considered in [33]:

$$\begin{aligned} \min \quad & f(x) \\ &= x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 \\ &\quad + 7x_4 \\ \text{s.t.} \quad & g_1(x) = 2x_1^2 + x_2^2 + x_3^2 + 2x_1 + x_2 + x_4 - 5 \\ &\leq 0 \\ & g_2(x) \\ &= x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 \\ &\leq 0 \\ & g_3(x) = x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10 \\ &\leq 0. \end{aligned} \quad (40)$$

Let  $a_1 = -2 \times 10^6$ ,  $b_1 = 0$ ,  $M_1 = -10^6$ . The numerical results of Algorithm 7 on example 4.1 with  $\varepsilon_1 = 0.05$ ,  $\eta = 0.1$  and different starting points are shown in Table 1.

The numerical results given in Table 1 show that all algorithms are completed in the first iteration and the numerical result of Algorithm 7 does not depend on the selection of the starting points for this example.

Let  $a_1 = -2 \times 10^6$ ,  $b_1 = 0$ ,  $M_1 = -10^6$ . The numerical results of Algorithm 9 or Algorithm 10 on this example with different starting point are shown in Tables 2 and 3.

TABLE 2: Numerical results of Algorithm 9 on Example 11 with different starting points.

$j$	$x^0$	$x^j$	$g_1(x^j)$	$g_2(x^j)$	$g_3(x^j)$	$f(x^j)$
1	(-10, -10, -10, -10)	(0.16989, 0.83553, 2.0085, -0.96503)	$-1.3381e - 10$	$-4.9983e - 11$	-1.8833	-44.2338
1	(10, 10, -10, -10)	(0.16936, 0.83591, 2.0086, -0.96502)	$-8.2839e - 10$	$-5.7484e - 10$	-1.8813	-44.2338
1	(-10, -10, 10, 10)	(0.16963, 0.8355, 2.0086, -0.96489)	$-2.6753e - 11$	$-1.0071e - 11$	-1.8833	-44.2338
1	(-10, -10, -10, -10)	(0.16942, 0.83559, 2.0087, -0.96487)	$-5.4548e - 11$	$-2.5295e - 11$	-1.8828	-44.2338

TABLE 3: Numerical results of Algorithm 10 on Example 11 with different starting points.

$j$	$x^0$	$x^j$	$g_1(x^j)$	$g_2(x^j)$	$g_3(x^j)$	$f(x^j)$
1	(10, 10, -10, -10)	(0.16947, 0.83564, 2.0086, -0.9649)	$-5.3531e - 12$	$-2.0135e - 12$	-1.8826	-44.2338
1	(-10, -10, 10, 10)	(0.16972, 0.83541, 2.0086, -0.96488)	$-2.6757e - 11$	$-1.007e - 11$	-1.8837	-44.2338

TABLE 4: Numerical results of Algorithm 7 on Example 12 with different starting points.

$j$	$x^0$	$x^j$	$g_1(x^j)$	$g_2(x^j)$	$f(x^j)$
1	(1, 1)	(0.00012003, 0.00012002)	-0.00012	-0.00012003	$2.28812e - 08$
1	(0, 0)	(0.00036527, 0.00036547)	-0.00036533	-0.00036527	$2.6699e - 07$

TABLE 5: Numerical results of Algorithm 9 on Example 12 with different starting points.

$j$	$x^0$	$x^j$	$g_1(x^j)$	$g_2(x^j)$	$f(x^j)$
1	(1, 1)	(0.00018226, 0.00018233)	-0.0001823	-0.00018226	$6.646e - 08$
1	(0, 0)	(0.00040823, 0.00040826)	-0.00040809	-0.00040823	$3.3333e - 07$

TABLE 6: Numerical results of Algorithm 10 on Example 12 with different starting points.

$j$	$x^0$	$x^j$	$g_1(x^j)$	$g_2(x^j)$	$f(x^j)$
1	(1, 1)	(0.00081614, 0.00081663)	-0.00081596	-0.00081614	$1.333e - 06$
1	(0, 0)	(0.00081597, 0.00081656)	-0.00081589	-0.00081597	$1.3326e - 06$

From the numerical results on Example 11, we can see that Algorithms 7, 9, and 10 can obtain almost the same approximate optimal solution. From the numerical results given in [33], we know that the optimal solution of Example 11 is (0.170160, 0.835886, 2.008125, -0.965392) with the objective function value -44.233828. Hence, the numerical results show that Algorithm 7 is efficient in this example.

*Example 12.* Consider the following problem considered in [24]:

$$\begin{aligned}
 \min \quad & f(x) = x_1^2 + x_2^2 \\
 \text{s.t.} \quad & x_1^2 - x_2 \leq 0 \\
 & -x_1 \leq 0.
 \end{aligned} \tag{41}$$

For this example, we let  $a_1 = -400$ ,  $b_1 = 100$ ,  $M_1 = -150$ . The numerical results of Algorithm 7 on Example 12 with  $\varepsilon_1 = 10^{-5}$ ,  $\eta = 0.1$  and different starting point are shown in Table 4.

Let  $a_1 = -400$ ,  $b_1 = 100$ ,  $M_1 = -150$ . The numerical results of Algorithm 9 or Algorithm 10 on Example 12 with different starting point are shown in Tables 5 and 6.

From Tables 4–6, we can see that Algorithm 7 has better numerical stability than Algorithms 9 and 10 for the optimal

solution and objective function value in this example. In fact, the given solution for Example 12 is (0, 0) with the objective function value 0.

## 5. Concluding Remarks

In this paper, we proposed a method for smoothing the non-smooth objective penalty function for inequality constrained optimization. Further, we showed the global convergence of the method under mild conditions. The given numerical experiments exhibit the efficiency of the proposed method.

## Data Availability

The data used in our numerical experiments are taken from [24, 33], and all used data released in this paper can be used directly.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Acknowledgments

This work was supported by National Natural Science Foundation of China (71371107, 61373027, and 11671228)

and Natural Science Foundation of Shandong Province (ZR2016AM10).

## References

- [1] M. Sun, Y. Wang, and J. Liu, "Generalized Peaceman-Rachford splitting method for multiple-block separable convex programming with applications to robust PCA," *Calcolo. A Quarterly on Numerical Analysis and Theory of Computation*, vol. 54, no. 1, pp. 77–94, 2017.
- [2] Y. Wang, L. Caccetta, and G. Zhou, "Convergence analysis of a block improvement method for polynomial optimization over unit spheres," *Numerical Linear Algebra with Applications*, vol. 22, no. 6, pp. 1059–1076, 2015.
- [3] K. Zhang and Y. Wang, "An  $H$ -tensor based iterative scheme for identifying the positive definiteness of multivariate homogeneous forms," *Journal of Computational and Applied Mathematics*, vol. 305, pp. 1–10, 2016.
- [4] Y. Wang, K. Zhang, and H. Sun, "Criteria for strong  $H$ -tensors," *Frontiers of Mathematics in China*, vol. 11, no. 3, pp. 577–592, 2016.
- [5] H. Chen, L. Qi, and Y. Song, "Column sufficient tensors and tensor complementarity problems," *Frontiers of Mathematics in China*, vol. 13, no. 2, pp. 255–276, 2018.
- [6] G. Wang, G. Zhou, and L. Caccetta, "Z-eigenvalue inclusion theorems for tensors," *Discrete and Continuous Dynamical Systems - Series B*, vol. 22, no. 1, pp. 187–198, 2017.
- [7] G. Wang, "Existence-stability theorems for strong vector set-valued equilibrium problems in reflexive Banach spaces," *Journal of Inequalities and Applications*, vol. 239, pp. 1–14, 2015.
- [8] G. Wang, X. Q. Yang, and T. C. Cheng, "Generalized Levitin-Polyak well-posedness for generalized semi-infinite programs," *Numerical Functional Analysis and Optimization*, vol. 34, no. 6, pp. 695–711, 2013.
- [9] J. Li, C. Wu, Z. Wu, and Q. Long, "Gradient-free method for nonsmooth distributed optimization," *Journal of Global Optimization*, vol. 61, no. 2, pp. 325–340, 2015.
- [10] J. Li, G. Chen, Z. Dong, and Z. Wu, "Distributed mirror descent method for multi-agent optimization with delay," *Neurocomputing*, vol. 177, pp. 643–650, 2016.
- [11] G. Wang and H.-t. Che, "Generalized strict feasibility and solvability for generalized vector equilibrium problem with set-valued map in reflexive Banach spaces," *Journal of Inequalities and Applications*, vol. 66, pp. 1–11, 2012.
- [12] H. Che, Y. Wang, and M. Li, "A smoothing inexact Newton method for  $P_0$  nonlinear complementarity problem," *Frontiers of Mathematics in China*, vol. 7, no. 6, pp. 1043–1058, 2012.
- [13] H. Zhang and Y. Wang, "A new CQ method for solving split feasibility problem," *Frontiers of Mathematics in China*, vol. 5, no. 1, pp. 37–46, 2010.
- [14] Y. Wang, L. Qi, and X. Zhang, "A practical method for computing the largest  $M$ -eigenvalue of a fourth-order partially symmetric tensor," *Numerical Linear Algebra with Applications*, vol. 16, no. 7, pp. 589–601, 2009.
- [15] D. Feng, M. Sun, and X. Wang, "A family of conjugate gradient methods for large-scale nonlinear equations," *Journal of Inequalities and Applications*, vol. 236, 2017.
- [16] G. Wang and H.-t. Che, "Generalized strict feasibility and solvability for generalized vector equilibrium problem with set-valued map in reflexive Banach spaces," *Journal of Inequalities and Applications*, vol. 1, pp. 1–11, 2012.
- [17] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, NY, USA, 1999.
- [18] W. I. Zangwill, "Non-linear programming via penalty functions," *Management Science*, vol. 13, pp. 344–358, 1967.
- [19] S. P. Han and O. L. Mangasarian, "Exact penalty functions in nonlinear programming," *Mathematical Programming*, vol. 17, no. 3, pp. 251–269, 1979.
- [20] E. Rosenberg, "Exact penalty functions and stability in locally Lipschitz programming," *Mathematical Programming*, vol. 30, no. 3, pp. 340–356, 1984.
- [21] G. Di Pillo and L. Grippo, "Exact penalty functions in constrained optimization," *SIAM Journal on Control and Optimization*, vol. 27, no. 6, pp. 1333–1360, 1989.
- [22] S. Lian and L. Zhang, "A simple smooth exact penalty function for smooth optimization problem," *Journal of Systems Science & Complexity*, vol. 25, no. 3, pp. 521–528, 2012.
- [23] S.-j. Lian, "Smoothing approximation to  $l_1$  exact penalty function for inequality constrained optimization," *Applied Mathematics and Computation*, vol. 219, no. 6, pp. 3113–3121, 2012.
- [24] X. Xu, Z. Meng, J. Sun, L. Huang, and R. Shen, "A second-order smooth penalty function algorithm for constrained optimization problems," *Computational optimization and applications*, vol. 55, no. 1, pp. 155–172, 2013.
- [25] Z. Y. Wu, H. W. Lee, F. S. Bai, and L. S. Zhang, "Quadratic smoothing approximation to  $l_1$  exact penalty function in global optimization," *Journal of Industrial and Management Optimization*, vol. 1, no. 4, pp. 533–547, 2017.
- [26] Z. Y. Wu, F. S. Bai, X. Q. Yang, and L. S. Zhang, "An exact lower order penalty function and its smoothing in nonlinear programming," *Optimization. A Journal of Mathematical Programming and Operations Research*, vol. 53, no. 1, pp. 51–68, 2004.
- [27] S. Lian and J. Han, "Smoothing approximation to the square-order exact penalty functions for constrained optimization," *Journal of Applied Mathematics*, vol. 2013, Article ID 568316, 7 pages, 2013.
- [28] S. Lian and Y. Duan, "Smoothing of the lower-order exact penalty function for inequality constrained optimization," *Journal of Inequalities and Applications*, vol. 185, 2016.
- [29] D. D. Morrison, "Optimization by least squares," *SIAM Journal on Numerical Analysis*, vol. 5, pp. 83–88, 1968.
- [30] R. Fletcher, "Penalty functions," in *Mathematical Programming: The State of The Art (Bonn, 1982)*, A. Bachem, M. Grottschel, and B. Korte, Eds., pp. 87–114, Springer, Berlin, 1983.
- [31] Z. Meng, Q. Hu, C. Dang, and X. Yang, "An objective penalty function method for nonlinear programming," *Applied Mathematics Letters*, vol. 17, no. 6, pp. 683–689, 2004.
- [32] J. Li, Z. Wu, and Q. Long, "A new objective penalty function approach for solving constrained minimax problems," *Journal of the Operations Research Society of China*, vol. 2, no. 1, pp. 93–108, 2014.
- [33] Z. Meng, C. Dang, M. Jiang, X. Xu, and R. Shen, "Exactness and algorithm of an objective penalty function," *Journal of Global Optimization*, vol. 56, no. 2, pp. 691–711, 2013.

## Research Article

# Filled Function Method for Nonlinear Model Predictive Control

Hajer Degachi <sup>1</sup>, Bechir Naffeti,<sup>2</sup> Wassila Chagra,<sup>1</sup> and Moufida Ksouri<sup>1</sup>

<sup>1</sup>Tunis El Manar University, National Engineering School of Tunis, Tunisia

<sup>2</sup>Carthage University, Faculty of Sciences of Bizerte, Tunisia

Correspondence should be addressed to Hajer Degachi; degachihajer@gmail.com

Received 15 December 2017; Accepted 7 May 2018; Published 4 June 2018

Academic Editor: Guillermo Cabrera-Guerrero

Copyright © 2018 Hajer Degachi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new method is used to solve the nonconvex optimization problem of the nonlinear model predictive control (NMPC) for Hammerstein model. Using nonlinear models in MPC leads to a nonlinear and nonconvex optimization problem. Since control performances depend essentially on the results of the optimization method, in this work, we propose to use the filled function as a global optimization method to solve the nonconvex optimization problem. Using this method, the control law can be obtained through two steps. The first step consists of determining a local minimum of the objective function. In the second step, a new function is constructed using the local minimum of the objective function found in the first step. The new function is called the filled function; the new constructed function allows us to obtain an initialization near the global minimum. Once this initialization is determined, we can use a local optimization method to determine the global control sequence. The efficiency of the proposed method is proved firstly through benchmark functions and then through the ball and beam system described by Hammerstein model. The results obtained by the presented method are compared with those of the genetic algorithm (GA) and the particle swarm optimization (PSO).

## 1. Introduction

Model predictive control (MPC) is a powerful control tool. This control strategy is formulated as the repeated solution of an open loop control problem [1]. At each sampling time, a control sequence is determined where only the first component is applied to the system and the optimization task is repeated at the next sampling instant.

The MPC algorithm presents the major advantage to efficiently handle constraints on input and output [2–4]. MPC is also able to control a wide variety of processes starting from systems that present a simple behavior like linear process [5] as well as those that exhibit more complex behavior like nonlinear [6, 7] and multivariable process [8].

The MPC theory was successfully applied in many areas of application such as chemical, petrochemical, pulp and paper, aerospace and defense, and food processing [9].

Firstly, MPC was well developed for linear models. Since almost real processes have a nonlinear behavior, this fact motivates the development of the NMPC strategy [4].

The NMPC is considered as a purely optimization-based algorithm. In fact, the minimization problem of the NMPC

problem is nonlinear and nonconvex due to the nonlinear nature of the model. The determination of the control sequence, solution of the NMPC optimization problem, should be done using a global optimization algorithm to ensure good control performances. For this, a variety of solutions were proposed in literature to solve this kind of problem. Many works were focused on online linearization of the nonlinear model [10–12]. A second solution was proposed in some other works that consists of reversing the nonlinear block to remove nonlinearity. This solution is only addressed to block-oriented models [13] where the nonlinear block is described by a polynomial. The two mentioned solutions allow obtaining a quadratic cost function at each sampling time where the global solution can be easily determined.

Since solving a nonlinear optimization problem at each sampling time is a hard-computational task, [14] has proposed to describe the nonlinear process by a set of uncertain linear models to overcome the online computational complexity of the NMPC. Since the used model in the NMPC is nonlinear and since real process must operate under rigorous conditions, the nonlinear characteristic of the model should be kept without simplification during the online computation

of the control sequence to get performances with sufficient accuracy.

For this, genetic algorithm (GA) [15], particle swarm optimization (PSO) [16], and neural network (NN) [17] were used as global optimization methods to solve the nonlinear optimization problem. Also, deterministic method such as Generalized Geometric Programming (GGP) was used as global optimization method for the NMPC [18, 19].

In this work, we will use the filled function as a global deterministic optimization method to solve the NMPC optimization problem. This method is based on the construction of a new function starting from the first local minimum found of the original function to be minimized and carry out the search of the global minimum using a local optimization method. This function allows determining the global minimum by finding an appropriate initialization.

In this work, the NMPC strategy is based on Hammerstein model. The Hammerstein model belongs to block-oriented models that is described by a nonlinear static block followed by a linear dynamic one. The special structure of the Hammerstein model was widely exploited to describe a variety of processes such as heat exchanger [17], SOFC [20], and pH neutralization process [21].

The rest of the paper is organized as follows. In Section 2, we describe the Hammerstein model. The NMPC based on Hammerstein model is presented in Section 3. In Section 4, we outline the optimization method used to solve the NMPC minimization problem. Simulation results on benchmark functions based on the global optimization method are given in Section 5. The efficiency of the proposed method for NMPC is illustrated in Section 6.

## 2. Hammerstein Model

The Hammerstein model is composed of a static nonlinear block followed by a linear dynamic one as depicted in Figure 1.

The outputs of the two blocks are given by the following equation:

$$S : \begin{cases} s(k) = f(u(k)) \\ A(q^{-1})y(k) = B(q^{-1})s(k) \end{cases} \quad (1)$$

The polynomials  $A(q^{-1})$  and  $B(q^{-1})$  are defined as

$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + \dots + a_{n_a}q^{-n_a} \quad (2)$$

$$B(q^{-1}) = b_1q^{-1} + b_2q^{-2} + \dots + b_{n_b}q^{-n_b} \quad (3)$$

where  $q^{-1}$  is the unit delay operator and  $n_a$  and  $n_b$  define the order of polynomials  $A(q^{-1})$  and  $B(q^{-1})$ .

Various types of nonlinearity can be used to describe the nonlinear block of the Hammerstein model. In this work, the polynomial form is chosen, and the function  $f$  is defined as

$$f(u(k)) = c_1u(k) + c_2u(k)^2 + \dots + c_{r_1}u(k)^{r_1} \quad (4)$$

where  $n_c$  is the order of the polynomial.

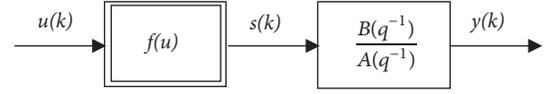


FIGURE 1: Hammerstein model.

## 3. Nonlinear Model Predictive Control Design

The aim of the NMPC strategy is to compute the control sequence  $U = [u(k), u(k+1), \dots, u(k+N_u-1)]$  by minimizing the criterion  $J$  defined as the difference between the desired trajectory  $y^{sp}(k+j)$  and the model output  $\hat{y}(k+j)$  over the prediction horizon  $N_p$  and penalizing the control increments over the control horizon  $N_u$

$$J = \sum_{i=1}^{N_p} (y^{sp}(k+i) - \hat{y}(k+i))^2 \quad (5)$$

$$+ \lambda \sum_{i=0}^{N_u-1} \Delta u(k+i)^2$$

$$\Delta u^{\min} \leq \Delta u(k+j) \leq \Delta u^{\max}, \quad j = 0, \dots, N_u - 1 \quad (6)$$

$$u^{\min} \leq u(k+j) \leq u^{\max}, \quad j = 0, \dots, N_u - 1$$

where  $\Delta u(k+i) = u(k+i) - u(k+i-1)$  and  $\lambda$  is a positive weighting coefficient.

Based on the receding horizon principle, only the first component will be applied to the system. Then the horizon will be shifted one step forward and the whole procedure will be repeated.

The predicted output of the Hammerstein model is defined as

$$\hat{y}(k+j) = -\sum_{i=1}^{n_a} a_i \hat{y}(k-i+j) + \sum_{i=1}^{n_b} b_i \hat{s}(k-i+j) \quad (7)$$

$\hat{s}(k+j)$  is the output of the nonlinear block defined as

$$\hat{s}(k+j) = \sum_{l=1}^{n_c} c_l u(k+j)^l \quad (8)$$

By using (7) and (8), the criterion expression (5) becomes

$$J(k) = \sum_{j=1}^{N_p} \left( y^{sp}(k+j) + \sum_{i=1}^{n_a} a_i \hat{y}(k-i+j) - \sum_{i=1}^{n_b} b_i \sum_{l=1}^{n_c} c_l u(k-i+j)^l \right)^2 + \lambda \sum_{j=0}^{N_u-1} \Delta u(k+j|k)^2 \quad (9)$$

Due to the nonlinear nature of Hammerstein model, the obtained objective function defined by (9) of the NMPC problem is nonconvex.

Solving the presented minimization problem using standard optimization methods will lead to suboptimal results in control. Deterministic optimization methods have the major benefit to offer the global solution. For this, we propose in this work to use the filled function method to solve the NMPC optimization problem. The present method was found to be easier to implement and offer an accurate solution but suffer from the high computation time.

#### 4. Optimization Method

In this work, we will use the filled function method to solve the NMPC optimization problem.

The concept of this global optimization method consists of determining a good initialization to obtain the global minimum. Therefore, the key of this method is to construct a function “ $P$ ” that allows moving from the local minimum of the objective function to obtain a more promising initialization point. This point presents a stationary point of the function  $P$ . The new initialization will allow finding easily the global minimum.

Thus, we consider a function  $f(x)$  to be minimized. The principle of the filled function method can be summarized as follows [22].

*Step 1.* Use a local optimization method to find a local minimum  $x_1$  of  $f(x)$ .

*Step 2.* Construct the filled function  $P(x)$  at the local minimum  $x_1$  of  $f(x)$ . Minimize the new function  $P$  using a local optimization method and so the minimum,  $t_1$ , of the filled function is found. The minimum  $t_1$  of the filled function is used as initialization to minimize again the function  $f$ , where a new better minimum will be found. Many filled functions are proposed in literature. Most of them are described using some parameters that can be difficult to be adjusted [23]. Other works minimize this number of parameters and reduce the filled function into a mono adjustable parameter function.

In this work, we will use the nonparametric filled function method proposed by [22] and defined as

$$P(t, x_1) = \begin{cases} -\int_{x_1}^x (f(t) - f(x_1)) dt & (x \geq x_1) \\ -\int_x^{x_1} (f(t) - f(x_1)) dt & (x \leq x_1) \end{cases} \quad (10)$$

The nonparametric filled function  $P$  should satisfy the following properties.

**Theorem 1.** *If  $x_1$  is an isolated minimizer of  $f : \Omega \subset \mathbb{R} \rightarrow \mathbb{R}$ , then  $x_1$  must be a maximizer of  $P(x, x_1)$ .*

**Theorem 2.** *If  $x_1$  is an isolated minimizer of  $f : \Omega \subset \mathbb{R} \rightarrow \mathbb{R}$ , then  $P(x, x_1)$  does not have any stationary point for  $f(x) > f(x_1)$ .*

**Theorem 3.** *If  $x_1$  is an isolated minimizer of  $f : \Omega \subset \mathbb{R} \rightarrow \mathbb{R}$ , then  $P(x, x_1)$  must have a stationary point for  $f(x) \leq f(x_1)$ . The flowchart of the considered method is given in Figure 2 that*

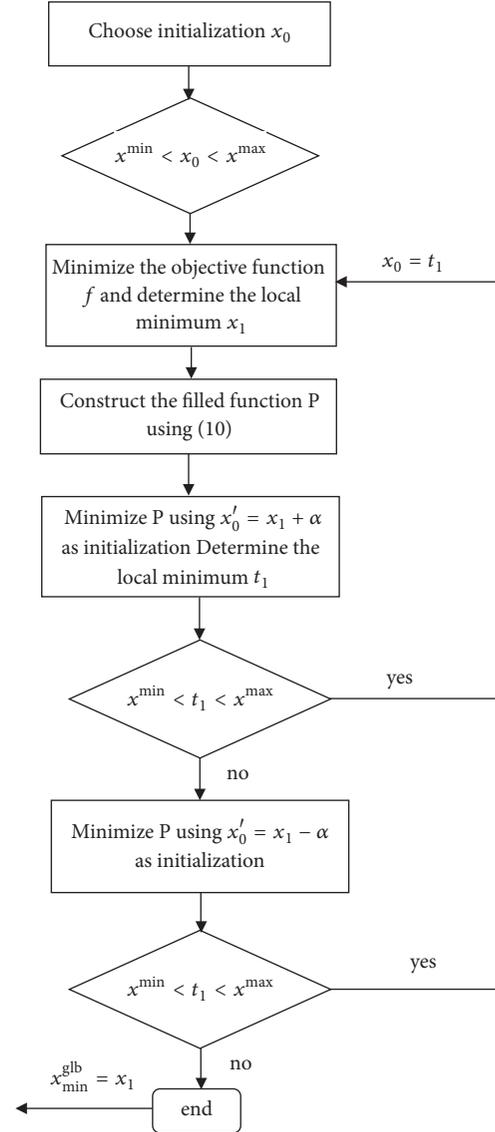


FIGURE 2: Flowchart of the filled function algorithm.

*details the different steps of the algorithm. The variable  $\alpha$  in the algorithm presents a small step within the interval  $[0, 1]$ .*

#### 5. Simulation Results on Benchmark Function

The efficiency of the filled function is demonstrated through benchmark functions defined in Table 1. We consider 3 multimodal functions as depicted in Figures 3, 4, and 5. The function  $f_1$  has a global minimum at  $x = 17.0392$  and two local minima at  $x = 5.3622$  and  $x = 10.4535$ . The function  $f_2$  has a global minimum at  $x = 5.1994$  and two local minima at  $x = 3.4392$  and  $x = 7.0678$ . The function  $f_3$  has a global minimum at  $x = -1$  and a local one  $x = 2$ . The minimization is carried out using the filled function and the local method illustrated by the “fmincon” function of the MATLAB environment.

Consider the first function  $f_1$ .

TABLE 1: Benchmark functions.

Function	Min	Research interval
$f_1(x) = \sin(x) + \sin\left(\frac{2x}{3}\right)$	17.03	[0, 20]
$f_2(x) = \sin(x) + \sin\left(\frac{10x}{3}\right) + \ln(x) - 0.84x$	5.36	[2.7, 7.5]
$f_3(x) = x^4 - 3x^3 - 1.5x^2 + 10x$	-1	[-2, 3]

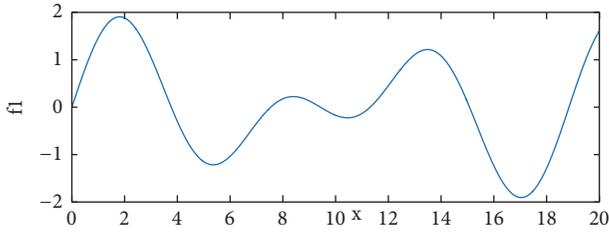


FIGURE 3: Evolution of the function  $f_1$ .

Iteration 1.

Step 1. The first step in the filled function optimization method consists of using a local optimization method to determine a local minimum of  $f$ . We choose as initialization  $x_0=2$ .

In this work, we use the gradient descent method to find a local minimum  $x_1$  of  $f$ . Using  $x_0=2$  as initialization and using the gradient we can get a local minimum,  $x_1 = 5.3622$ , as indicated in Figure 6

Step 2. Using  $x_1$  we construct the filled function  $P(t, x_1)$  using (10) as illustrated in Figure 6.

Step 3. Once we have  $P$ , we apply a local optimization method to determine a local minimum of  $P$ , and we set initialization as  $t_0 = x_1 + \alpha$ , where  $\alpha$  is a small step real value. The obtained minimum of  $P$  is  $t_1$

Iteration 2. If  $t_1$  belongs to the interval of variation of the manipulated variable, it will be used in the second iteration as initialization for  $f$ .

Step 1. Since this condition is true in our case,  $t_1$  presents a new initialization for  $f$  as depicted in Figure 6.

The new initialization presents a promising point to get a best minimum than that found in the first step.

The new minimum,  $x_1$ , of  $f$  found with initialization  $t_1$  is shown in Figure 7.

Step 2. Using the new found minimum  $x_1$  indicated in Figure 7 we construct the filled function  $P(t, x_1)$  using (10) as illustrated in Figure 7. We can conclude from Figure 7 that  $P$  does not have a stationary point in the considered interval. So, the global minimum of  $f_1$  is  $x_1=17.0328$ .

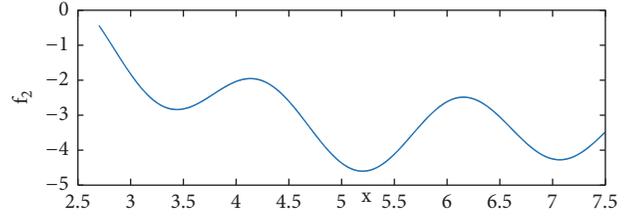


FIGURE 4: Evolution of the function  $f_2$ .

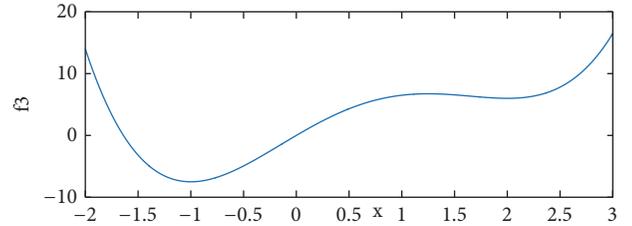


FIGURE 5: Evolution of the function  $f_3$ .

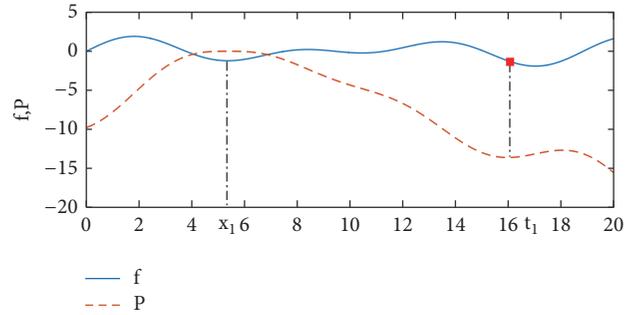


FIGURE 6: Evolution of  $f_1$  and  $P$  in the first iteration.

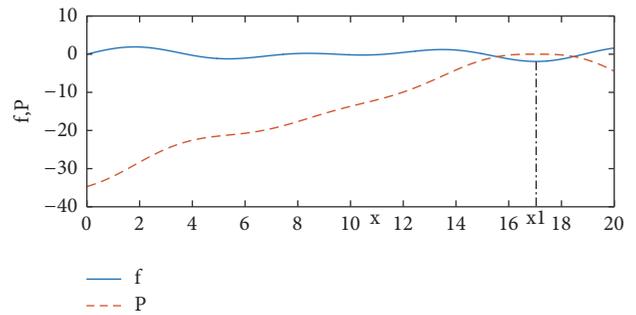


FIGURE 7: Evolution of  $f_1$  and  $P$  in the second iteration.

The minimization results for the considered function,  $f_1$ ,  $f_2$ , and  $f_3$ , are illustrated, respectively, in Tables 1, 2, and 3

Based on results given by Tables 2, 3, and 4, we can note that, for different initialization values, the convergence to the global minimum using the local optimization method is not always guaranteed. Contrariwise, we can conclude that whatever the value of the initialization the minimization of the considered function using the proposed method is always guaranteed.

TABLE 2: Minimization of  $f_1$  using global and local optimization method.

Initialization	Reached min (proposed method)	Reached min (local method)
3	17.0328	5.3622
7	17.0328	5.3622
9	17.0328	10.4535
14	17.0328	17.0328

 TABLE 3: Minimization of  $f_2$  using global and local optimization method.

Initialization	Reached min (proposed method)	Reached min (local method)
4.5	5.1995	3.4392
6	5.2006	7.0678
6.5	5.1994	5.1998
7	5.1994	7.0697

 TABLE 4: Minimization of  $f_3$  using global and local optimization method.

Initialization	Reached min (proposed method)	Reached min (local method)
2.5	-1	-1
1.5	-1	2
-1.5	-1	-1

## 6. Simulation Results

The system under consideration is a ball and beam which is composed of ball, ball position sensor, a center pivoted beam on which the ball rolls on, beam angle sensor, and the servomotor. The control objective is to fix the ball on a desired position. The ball and beam system is described by a Hammerstein model defined as [24]

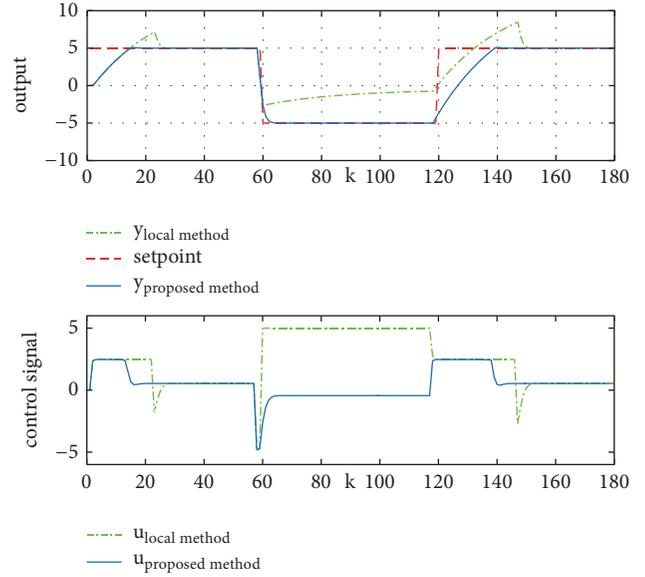
$$S : \begin{cases} y(k) = -a_1 y(k-1) + b_1 s(k-1) \\ s(k) = c_1 u(k) + c_2 u(k)^2 \end{cases} \quad (11)$$

The different parameters of the linear and nonlinear blocks are given by  $a_1 = -0.62$ ,  $b_1 = 1$ ,  $c_1 = 0.396$ , and  $c_2 = -0.036$ .

The prediction and the control horizons are fixed to  $N_p = 2$ ,  $N_u = 1$  and constraints on the manipulated variable  $u$  are fixed as  $-5 \leq u(k) \leq 5$ .

First, the NMPC algorithm for the ball and beam system is implemented using the filled function method to solve the minimization problem. The desired trajectory is fixed as -5 and 5 represented by three step changes. The NMPC algorithm based on the proposed method is compared to a local optimization method illustrated by "fmincon" function of the MATLAB environment. The simulation results are depicted in Figure 8.

We can note from Figure 8 that the NMPC performances based on the filled function method are significantly better


 FIGURE 8: Output tracking using the proposed method and a local method for  $\lambda = 0.1$ .

than those of the NMPC based on a local optimization method. The good ability of the filled function method to solve the optimization problem and offer the optimal control quality is justified by the fact that the NMPC based on the proposed method ensures the setpoint tracking with zero steady state error. Also, we can note that the response of the system based on the local optimization method is characterized by a high overshoot compared to the results given by the response of the system based on the filled function method.

In the sequel, the NMPC based on the proposed method is compared with the NMPC based on GA and PSO algorithms.

To compare the performances of the different algorithms we consider two performance indexes: the SSE and the SCV.

The SSE is defined as

$$SSE = \frac{1}{N} \sum_{i=1}^N (y^{sp}(k) - y(k))^2 \quad (12)$$

Since our goal in control is to reach the desired trajectory and ensure the good output tracking by determining the optimal control sequence that gives the minimum dissipated energy, we will consider the second performance index SCV defined as

$$SCV = \frac{1}{N} \sum_{k=1}^N u(k)^2 \quad (13)$$

We can note from Figure 9 that the three algorithms are able to reach the desired trajectory and ensure good output tracking as proved by the SSE value. The significant difference lays in the control quality. Based on Figure 9 we can note that each algorithm reaches the desired trajectory with a different control sequence. Table 5 shows that the best

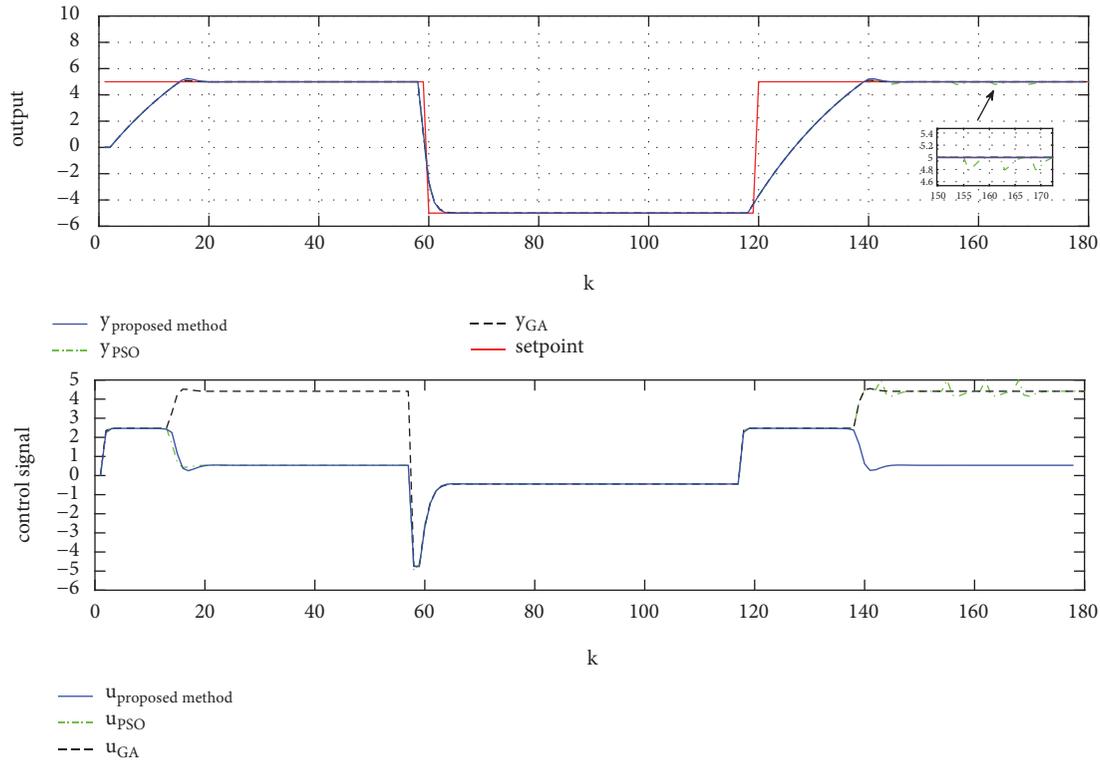


FIGURE 9: Output tracking using the GA, PSO, and the proposed method for  $\lambda = 0.1$ .

TABLE 5: The performance of the NMPC based on the three algorithms for  $\lambda = 0.1$ .

Method	Overshoot	SCV	SSE
Proposed method	0.0142	1.635	3.114
GA	0.0178	10.729	3.120
PSO	0.0178	5.990	3.112

TABLE 6: The performance of the NMPC based on the three algorithms for  $\lambda = 0.2$ .

Method	Overshoot	SCV	SSE
proposed method	0.0494	1.646	3.114
GA	0.0594	5.994	3.123
PSO	0.0594	10.681	3.124

control quality is given by the proposed method since it offers the small SCV. Also, we can remark that the filled function method offers the less overshoot as illustrated by Table 5. We can conclude that for the proposed method the setpoint is reached with the best control quality and the less overshoot. Although the results of setpoint tracking obtained by the GA, PSO, and the proposed method are quite similar, it is very clear that the control quality of GA and PSO is inferior to the filled function method. In fact, setting parameters of evolutionary algorithms as coding of individuals, population size, crossover and mutation rates, and stopping criterion has an important role in the determination of an accurate solution [25].

To show the effectiveness of our proposed method with different values of  $\lambda$  we suggest varying  $\lambda$  and carry out simulation for  $\lambda = 0.2$ .

We can notice from Figure 10 and Tables 5 and 6 that the increase of the weighting factor  $\lambda$  results in a high overshoot value and slow system response. Based on Table 4 we can conclude that the SCV offered by the filled function method is

better than that given by GA and PSO. This confirms the good ability of the filled function method to give the best control quality.

Tables 5 and 6 give the performance of the different algorithms for  $\lambda = 0.1$  and  $\lambda = 0.2$ . We can note that for different values of  $\lambda$  the less overshoot and SCV are given by the filled function method.

## 7. Conclusion

In this work, a nonlinear model predictive control for Hammerstein model is presented. Using nonlinear model in the MPC strategy complicates the solution of the optimization problem and leads to computational difficulties. Consequently, an efficient optimization algorithm should be used to overcome these difficulties and ensure good control results. The filled function, as a global optimization method, is used in this work to solve the nonconvex minimization problem of the NMPC. The proposed method proves its efficiency to give good control performances through nonlinear process

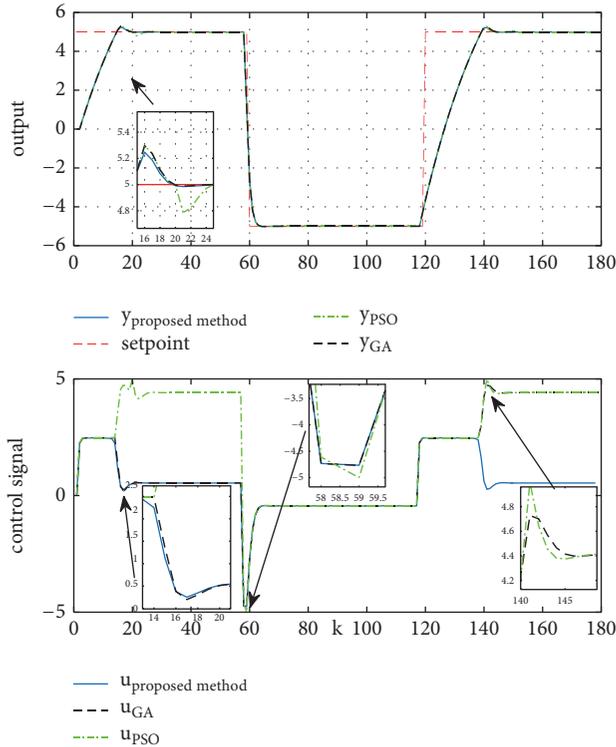


FIGURE 10: Output tracking using the GA, PSO, and the proposed method for  $\lambda = 0.2$ .

described by a ball and beam system. It offers a good output tracking and a less overshoot. The filled function method is then compared with two heuristic global optimization algorithms: GA and PSO. We can remark that the GA and the PSO algorithms are quite similar. We can also conclude that the NMPC with the proposed method is much better than the GA and PSO since it offers less overshoot and gives the best control sequence at each sampling time.

**Conflicts of Interest**

The authors declare that there are no conflicts of interest regarding the publication of this paper.

**References**

[1] F. Allgöwer, R. Findeisen, and Z. K. Nagy, “Nonlinear model predictive control: From theory to application,” *J. Chin. Inst. Chem. Engrs*, vol. 35, no. 3, pp. 299–315, 2004.

[2] P. Tatjewski, *Advanced control of industrial processes: structures and algorithms*, Springer Science & Business Media.

[3] D. Knuth, *The Art of Programming*, 2007, pp. 82–86.

[4] B. Kouvaritakis and M. Cannon, Eds., *Non-linear Predictive Control: theory and practice*, vol. 61, 2001.

[5] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, “Constrained model predictive control: stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[6] E. Hellström, M. Ivarsson, J. Åslund, and L. Nielsen, “Look-ahead control for heavy trucks to minimize trip time and fuel

consumption,” *Control Engineering Practice*, vol. 17, no. 2, pp. 245–254, 2009.

[7] P. Ortner, R. Bergmann, H. J. Ferreau, and L. Del Re, “Nonlinear model predictive control of a diesel engine airpath,” *IFAC Proceedings Volumes*, vol. 42, no. 2, pp. 91–96, 2009.

[8] A. Kheriji, F. Bouani, and M. Ksouri, “A GGP approach to solve non convex min-max predictive controller for a class of constrained MIMO systems described by state-space models,” *International Journal of Control, Automation, and Systems*, vol. 9, no. 3, pp. 452–460, 2011.

[9] S. J. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology,” *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.

[10] M. Ławryńczuk, “Nonlinear predictive control of dynamic systems represented by Wiener-Hammerstein models,” *Nonlinear Dynamics*, vol. 86, no. 2, pp. 1193–1214, 2016.

[11] M. Ławryńczuk, “Computationally efficient nonlinear predictive control based on neural Wiener models,” *Neurocomputing*, vol. 74, no. 1-3, pp. 401–417, 2010.

[12] M. Ławryńczuk, “Practical nonlinear predictive control algorithms for neural Wiener models,” *Journal of Process Control*, vol. 23, no. 5, pp. 696–714, 2013.

[13] S. J. Norquay, A. Palazoglu, and J. A. Romagnoli, “Application of Wiener model predictive control (WMPC) to a pH neutralization experiment,” *IEEE Transactions on Control Systems Technology*, vol. 7, no. 4, pp. 437–445, 1999.

[14] F. Khani and M. Haeri, “Robust model predictive control of nonlinear processes represented by Wiener or Hammerstein models,” *Chemical Engineering Science*, vol. 129, pp. 223–231, 2015.

[15] H. Al-Duwaish and W. Naeem, “Nonlinear model predictive control of Hammerstein and Wiener models using genetic algorithms,” in *Proceedings of the 2001 IEEE International Conference on Control Applications (CCA’01)*, pp. 465–469, Mexico City, Mexico.

[16] S. S. Kaddah, K. M. Abo-Al-Ez, and T. F. Megahed, “Application of nonlinear model predictive control based on swarm optimization in power systems optimal operation with wind resources,” *Electric Power Systems Research*, vol. 143, pp. 415–430, 2017.

[17] D. R. Ramírez, M. R. Arahall, and E. F. Camacho, “Min-max predictive control of a heat exchanger using a neural network solver,” *IEEE Transactions on Control Systems Technology*, vol. 12, no. 5, pp. 776–786, 2004.

[18] D. Hajer, C. Wassila, and K. Moufida, “Global optimization method for Min-Max MPC based on Wiener and Hammerstein model,” in *Proceedings of the 2015 7th International Conference on Modelling, Identification and Control (ICMIC)*, pp. 1–6, Sousse, Tunisia, December 2015.

[19] W. Chagra, H. Degachi, and M. Ksouri, “Nonlinear model predictive control based on Nelder Mead optimization method,” *Nonlinear Dynamics*, pp. 1–12, 2017.

[20] H.-B. Huo, X.-J. Zhu, W.-Q. Hu, H.-Y. Tu, J. Li, and J. Yang, “Nonlinear model predictive control of SOFC based on a Hammerstein model,” *Journal of Power Sources*, vol. 185, no. 1, pp. 338–344, 2008.

[21] E. F. Camacho and C. B. Alba, *Model predictive control*, Springer Science & Business Media, 2007.

[22] B. M. Ismail, K. W. Goh, and B. D. Yosza, “Solving global optimization using non-parametric filled function,” *Jurnal Mat Stat*, vol. 11, no. 2, 2011.

- [23] W.-B. Du, Y. Gao, C. Liu, Z. Zheng, and Z. Wang, "Adequate is better: particle swarm optimization with limited-information," *Applied Mathematics and Computation*, vol. 268, pp. 832–838, 2015.
- [24] A. Raees and M. B. Kadri, "Fuzzy Hammerstein model based generalized predictive control for ball and beam system," in *Proceedings of the 2012 International Conference on Emerging Technologies (ICET)*, pp. 1–6, Islamabad, Pakistan, October 2012.
- [25] F. J. Lobo, C. F. Lima, and Z. Michalewicz, Eds., *Parameter setting in evolutionary algorithms*, vol. 54, Springer Science & Business Media, 54, 2007.

## Research Article

# Multiobjective Optimization for a Wireless Ad Hoc Sensor Distribution on Shaped-Bounded Areas

Armando Céspedes-Mota <sup>1</sup>, Gerardo Castañón <sup>1</sup>,  
Alberto F. Martínez-Herrera <sup>1</sup> and Leopoldo Eduardo Cárdenas-Barrón <sup>2</sup>

<sup>1</sup>Department of Electrical and Computer Engineering, Tecnológico de Monterrey, 2501 Eugenio Garza Sada Sur, 64849 Monterrey, NL, Mexico

<sup>2</sup>School of Engineering and Sciences, Tecnológico de Monterrey, 2501 Eugenio Garza Sada Sur, 64849 Monterrey, NL, Mexico

Correspondence should be addressed to Leopoldo Eduardo Cárdenas-Barrón; [lecarden@tec.mx](mailto:lecarden@tec.mx)

Received 30 January 2018; Revised 6 April 2018; Accepted 2 May 2018; Published 3 June 2018

Academic Editor: Abhishek K. Gupta

Copyright © 2018 Armando Céspedes-Mota et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Resource efficiency in wireless ad hoc networks has become a widely studied NP-problem. This problem may be suboptimally solved by heuristic strategies, focusing on several features like the channel capacity, coverage area, and more. In this work, maximizing coverage area and minimizing energy consumption are suboptimally adjusted with the implementation of two of Storn/Price's Multiobjective Differential Evolution (DE) algorithm versions. Additionally, their extended representations with the use of random- $M$  parameter into the mutation operator were also evaluated. These versions optimize the initial random distribution of the nodes in different shaped areas, by keeping the connectivity of all the network nodes by using the Prim-Dijkstra algorithm. Moreover, the Hungarian algorithm is applied to find the minimum path distance between the initial and final node positions in order to arrange them at the end of the DE algorithm. A case base is analyzed theoretically to check how DE is able to find suboptimal solutions with certain accuracy. The results here computed show that the inclusion of random- $M$  and completion of the algorithm, where the area is pondered with 60% and the energy is pondered with 40%, lead to energy optimization and a total coverage area higher than 90%, by considering the best results on each scenario. Thus, this work shows that the aforementioned strategies are feasible to be applied on this problem with successful results. Finally, these results are compared against two typical bioinspired multiobjective algorithms, where the DE algorithm shows the best tradeoff.

## 1. Introduction

Nowadays, it is well known that the distribution of sensors in a wireless ad hoc sensor network (WAHSN) is a challenge in the research of wireless communications, because it is necessary to increase both lifetime and coverage area demand of the network. Furthermore, it is required to avoid a weak connectivity.

A WAHSN contains a number of geographically dispersed mobile sensors. Each sensor node has wireless communication, collaborative signal processing, and network capabilities subject to some technical constraints. The mobile sensor nodes are free to move and self-organize in an arbitrary way while the communication is made via radio transceivers. Therefore, a WAHSN is able to determine the

value of different parameters such as temperature and geographic coordinates of a given location, detect the occurrence of events, classify a detected object, and track an object. In consequence, it is necessary that the required data be disseminated to the proper end users [1, 2]. It is important to remark that wireless sensor networks (WSN) are a type of ad hoc network composed of a large number of sensor nodes, which are densely deployed either inside the sense perception area or very close to it. The nodes are not connected to each other. Basically, they are stationary or at most slowly moving [1]. In contrast, a mobile ad hoc network (MANET) is an autonomous collection of mobile users connected by one to one wireless links. The MANETs topology may change continuously in an unpredictably way over time. The mobile nodes in MANETs can renew, replace, or recharge

their batteries or source of energy. Although WAHSNs, WSNs, and MANETs involve multihop communications, many end-to-end routing schemes proposed for MANETs are inappropriate for WAHSN and WSNs for the previously mentioned reasons [3]. Nevertheless, sensor distribution schemes between WAHSNs and MANETs can be compared. A key factor that distinguishes the WAHSNs from MANETs is that the end goal is detection and estimation of some events of interest and not just communication [1, 2].

In spite of the fact that our work is applicable to WAHSNs, MANETs, and also WSNs under certain modifications, the analysis on WAHSNs due to some characteristics and resource constraints such as energy, coverage area, and connectivity optimization is made.

Sensor distribution is one of the fundamental problems in WAHSNs and regularly has been suboptimally solved by heuristic strategies such as genetic algorithms, among several others as shown by [4–8]. It is worth mentioning that sensor distribution with minimum energy connectivity is an NP-complete problem [9–12]. In the literature, different strategies for wireless ad hoc and sensor networks related to topology control [2], node localization [13, 14], distributed coverage [15], and network lifetime [16] are described to provide solutions for sensor distribution, coverage area, and energy consumption problems. Additional related problems in wireless communication such as frequency allocation and interference have been also solved by bioinspired algorithms as it is reported in [17, 18] or in [19], where the authors show an efficient way of solving the minimum number of hops among nodes to be communicated.

To optimize the sensor distribution, DE algorithm is used. DE is a very simple but very powerful stochastic global optimizer for a continuous search domain. It was proposed by [20] and represents a very complex evolution process. Cleverly using the differences between the populations, a simple but fast linear operator called differentiation is created, which makes DE unique. A survey of the applications of DE to different optimization problems shows that DE generally outperforms other evolutionary algorithms [21–23]. An updated reference with novel DE algorithm versions and additional applications are found at [24, 25]. DE exploits a population of potential solutions to effectively probe the search spaces. The algorithm is initialized with a population of random candidate solutions, which are conceptualized as individuals. In this work, an individual includes the coordinates of the sensor nodes  $(x, y)$  followed by its communication radii values. For each individual in the population, a descendant is created from three parents. One parent, called the main parent, is disturbed by the vector of the difference of the other two parents. If the descendant has a better performance as measured by the objective function, it replaces the individual; otherwise, the original individual is retained and is passed on to the next generation of the algorithm and the descendant is discarded. This process is repeated until it reaches the termination condition. For a complete theoretical analysis of the DE algorithm, the reader is referred to [20].

Node distribution optimization in mobile sensor networks is presented in [26], where they analyze the sensor network coverage area and its coverage redundant area using

DE. Moreover, the optimal power allocation in WSNs using DE is given in [27]. Also, in [28], DE is applied to maximize the lifetime of the WSNs using disjoint sets of sensors to cover a set of targets. However, it is important to mention that [26–28] do not consider ad hoc sensor networks. They use the DE algorithm only for WSNs. On the other hand, a DE based topology control mechanism in MANETs is presented in [29] to deploy the nodes with the maximum coverage area. Here, the topology control is managed with the use of a Markov chain model. A similar approach is followed in [30] to evaluate the node distribution, but now based on Voronoi regions for each node. However, [29, 30] do not consider optimization of energy consumption for maintaining the network connectivity and coverage area simultaneously in a multiobjective fitness function.

The approach followed in [31] analyzed the coverage area of sensor networks, where the main goal is to maximize the coverage area in presence of obstacles such as walls. In [32] the same approach of maximizing the coverage area in the presence of obstacles is extended, including the energy consumed in the objective function. The redundant area is also considered in the objective function. In [33], an improved DEA version based on a best-optimal-solution sorting technique is compared with the classical DEA and the Nondominated Sorting Genetic Algorithm II (NSGA-II), showing that the first one has the best performance when computing the coverage area. Additionally, the number of nodes needed to obtain the maximum coverage area is diminished to determine the minimum feasible. In [34], the coverage area is optimized by using the improved versions of DEA such as the Self-Adaptive Differential Evolution (jDE) and the Adaptive Differential Evolution (JADE). As expected, the jDE and JADE have better performance than the classical DEA versions. In [35], DEA is applied for node position estimation in convex and nonconvex configurations.

Other aspects of wireless sensor networks can be tackled by implementing other bioinspired algorithms. For instance, in [6], the coverage area is optimized with Particle Swarm Optimization (PSO). Coverage area and energy are considered in a multiobjective version of PSO as shown in [5], where the parameters are controlled with a clustering approach. For routing management, Ant Colony is implemented in [36] as well as in [8]. Routing is also solved by the Strength Pareto Evolutionary algorithm (SPEA2) as shown in [37]. Optimal path problem is treated in [38], where DEA is the best solution compared to the classical PSO and GA strategies. Node anchors can be deployed with bioinspired techniques as shown in [39] with Ant Colony. Such scenario is found when RFID readers are deployed to detect tags. In [40], a combination of GA and PSO is generated to do so.

Additionally, in [41], two multiobjective algorithms are compared, the Multiobjective Simulated Annealing (MOSA) and NSGA-II. Such comparison is done to determine the algorithm that gives the best distribution output with fewer nodes. In [42], two algorithms are compared to solve WSN scenarios: SPEA2 and NSGA-II. The objective function includes the energy consumption, the coverage area, and the network reliability, where the energy component is analyzed against the elapsed time. For scenarios where the complexity

is increased, SPEA2 is better than NSGA-II and vice versa. In [43] different WSN scenarios are tested, where NSGA-II and Learning Automata (LA) are combined to create a hybrid multiobjective algorithm. This hybrid version outperforms NSGA-II. The main goal is to minimize the number of active sensors that cover the maximum possible area and minimize the energy consumption while maintaining the network connectivity. In [44] the Multiobjective Evolution algorithm based on Decomposition (MOEA/D) and NSGA-II are compared to determine which is the best option to maximize the coverage area and minimize the energy consumption in WSN, where it is claimed that MOEA/D gives better results than NSGA-II. In [45], three optimization strategies are evaluated, the Multiobjective Particle Swarm Optimization (MOPSO), NSGA-II, and a proposed optimization denoted as Heuristic 3-Phases (H3P). H3P is formed by 3 steps: clustering, Pareto construction, and disassembly. The goal is to minimize the number of sensors when maximizing the number of targets to be detected, where H3P presents the best performance among the presented optimization strategies. In [46], NSGA-II is implemented and compared with MOEA/D to obtain the best routing combination by minimizing the energy consumption while detecting a large number of targets as possible, where it is shown that NSGA-II is better than MOEA/D. In [47], a discrete DEA version and NSGA-II are implemented to reach the best compromise among three goals: the delay of a packet from the origin to the destiny in the network, the packet delivery ratio, and the flow conservation. It is done to help the routing process with time efficiency, where it is shown that DEA outperforms NSGA-II. A detailed survey for more available techniques applied on WSN can be consulted in [48].

Compared to the existing bioinspired techniques in the open literature [26, 32] and the method here presented, the followed approach is based on a simplified version of the MOEA/D techniques presented in [49], where a set of weights is assigned on each function to be optimized and the summation of them is equal to 1. Such variant is known as *Weighted Sum Approach* [49, 50]. For the sake of clarity, the method here implemented is denoted as Multiobjective Differential Evolution algorithm (MODEA).

MODEA is applied to the optimization of network energy consumption to maintain the network connectivity and network coverage area in WAHSNs, following a convex model [35]. The algorithm makes sensor radius adjustments and assumes the mobility of the sensors in order to optimize two objectives: maximizing the network coverage area and reducing network energy consumption. It is worth mentioning that consequently also the network lifetime is extended.

Additionally, the proposed algorithm emphasizes the successful communication between sensors in a given neighborhood, with the use of the Prim–Dijkstra algorithm described by [51, 52]. If a tree is not created, the nodes distribution is discarded even when the network fulfills the rest of the restrictions regarding area, energy, and bounds on different shaped areas. Also, the Hungarian algorithm [53] is used to find the shortest distance between the initial and the final node positions. This information is important for the case

of having mobile ad hoc sensors; each node knows where to move with the minimal distance.

A summary of the employed techniques and the involved parameters is shown in Table 1.

The rest of this paper is organized as follows: Section 2 presents the methodology. Section 3 describes theoretical models to find the energy lower bound and coverage area's upper bound. Section 4 develops and explains the MODEA for wireless ad hoc sensor distribution. Section 5 provides numerical results and a discussion. Finally, Section 6 gives the conclusions and some future research directions.

## 2. Description of the Method

**2.1. Sensor Coverage Model.** The coverage area of a sensor is the region within which the sensor is able to detect or analyze the sensing parameters. There exists a special interest in the sensor distribution for which communication with neighboring nodes is more energy efficient. Also, it is assumed that the coverage radius of each sensor helps to determine its communication link distance through minimum spanning tree to any of its neighbors, and it depends on the kind of sensors considered. Additionally, the sensor distribution on different shaped areas is considered. The main objective of the sensor coverage model is to achieve a balance between the maximum effective coverage area and the minimum communication sensor energy.

According to [26], the node set on the target area  $m \times n$  grid can be defined as

$$\mathcal{N} = \{N_1, N_2, \dots, N_z\}, \quad (1)$$

where  $z$  is the cardinality of  $\mathcal{N}$ .

The coverage range of a node  $N_i$  can be expressed as a circle centered at its coordinates  $(x_i, y_i)$  with sensing radius  $r_i$ . The  $(x_i, y_i)$  coordinates express the position of sensor nodes and the subindex  $i$  represents the sensor node index. A grid point  $P(x, y)$  is covered by a sensor node  $i$  if and only if its distance to the center  $(x_i, y_i)$  of the circle (2) is not larger than the sensing radius  $r_i$ :

$$d(N_i, P) = \sqrt{(x - x_i)^2 + (y - y_i)^2}. \quad (2)$$

First of all, a binary detection model is considered. A random variable  $E_i$  is introduced to describe the event in which the sensor node  $i$  covers a given point  $P(x, y)$ . The probability of the event  $E_i$ ,  $P\{E_i\}$ , is equal to the coverage probability  $P_{\text{cov}}(x, y, N_i)$ . This probability is a binary valued function such as

$$P_{\text{cov}}(x, y, N_i) = \begin{cases} 1 & \text{if } (x - x_i)^2 + (y - y_i)^2 \leq r_i^2 \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

This model is applied in all cases in Section 5. It is clear that a point should be evaluated against all the nodes present in the network; then

$$S(x, y) = \sum_{i=1}^z P_{\text{cov}}(x, y, N_i), \quad (4)$$

TABLE 1: Summary of the references presented in Section 1.

ID	Cite	Technique	Involved variables
DE	[26]	DE/rand/1	Coverage area, energy
	[27]	DE/rand/1, DE/Best/1	Power allocation
	[27]	DE/Curr-to-Best/1, DE/rand-to-best/1	Power allocation
	[28]	DE/Best/1	Coverage area
	[29]	DE/rand/1, Markov topology control	Coverage area
	[30]	DE/rand/1, Voronoi topology control	Coverage area
	[31]	DE/rand/1	Coverage area, redundant area
	[32]	DE/rand/1 with rand $M$	Coverage area, energy, redundant area
	[33]	Classical and modified DE/rand/1, NSGA-II	Coverage area with node elimination
	[34]	DE/Best/1, jDE, JADE	Coverage area
	[35]	Modified DE/Curr-to-Best/1	Node position estimation
Others	[47]	DEA and NSGA-II	Routing
	[5]	PSO	Energy and coverage area (by clustering)
	[6]	PSO	Coverage area
	[8]	Ant Colony	Routing
	[36]	Ant Colony	Routing
	[37]	SPEA2	Routing
	[38]	DE, PSO, GA	Optimal path
	[39]	Ant Colony	Node deployment (target coverage)
	[40]	Combined GA and PSO	Node deployment (target coverage)
	[41]	MOSA and NSGA-II	Node deployment
	[42]	SPEA-2 and NSGA-II	Energy, coverage area
	[43]	NSGA-II and LA	Energy, coverage area
	[44]	MOEA/D and NSGA-II	Energy, coverage area
	[45]	NSGA-II, MOPSO, H3P	Node deployment (target coverage)
	[46]	NSGA-II	Node deployment (target coverage), routing

where  $S(x, y)$  is the number of times that the coordinate  $(x, y)$  is covered by the node set.

Nevertheless, to obtain the proper covered area by the set of nodes, the function  $R(x, y)$  is computed to verify if the coordinate is covered at least once or if it is not covered at all  $N_i$ .

$$R(x, y) = \begin{cases} 1 & S(x, y) \geq 1 \\ 0 & S(x, y) = 0, \end{cases} \quad (5)$$

and then the covered area is computed as

$$A_{\text{cov}} = \frac{\sum_{y=1}^n \sum_{x=1}^m R(x, y)}{m \times n}. \quad (6)$$

Equation (6) is implemented in our simulation by sweeping all the  $(x, y)$  points of the target area  $m \times n$ . For different area configurations, this area is replaced by the bounded area of interest.

**2.2. Energy Consumption.** There are models for efficient energy consumption in MANETs [5], WAHSNs [10], WSNs [54], and many others. The energy consumption model used in this work is based in [55], this model is also used in [26].

In order to show the optimization process in coverage area and energy consumption, the total energy for maintaining the network connectivity is considered as the total energy consumption of the network.

$$E_{\text{csd}} = \mu \sum_{i=1}^z r_i^\alpha, \quad (7)$$

where  $\mu$  is the power per unit area in milliwatts per square meter ( $\text{mW}/\text{m}^2$ ), considering that the sensors are on the floor,  $r_i$  is the sensor radius of the node  $i$  in meters, and  $\alpha = 2$  since the communication medium is the air (free space) [56]. In this paper,  $\mu$  equals  $0.005$  ( $\text{mW}/\text{m}^2$ ). The path loss exponent  $\alpha$  takes different values according the environment. For urban area cellular radio  $\alpha = 2.7$ – $3.5$ , for shadowed urban cellular radio  $\alpha = 3$ – $5$ , for in-building line-of-sight from  $1.6$  to  $1.8$ , for obstructed in-building from  $4$  to  $6$ , and for obstructed in factories from  $2$  to  $2.3$ , the power per unit area changes in a 3D environment to  $\text{mW}/\text{m}^2$ , [56].

### 3. Base Case: Ideal Distribution of 10 Nodes into a Squared Area

In this section, the methods to obtain the energy and coverage area bounds are presented. In order to find these lower and

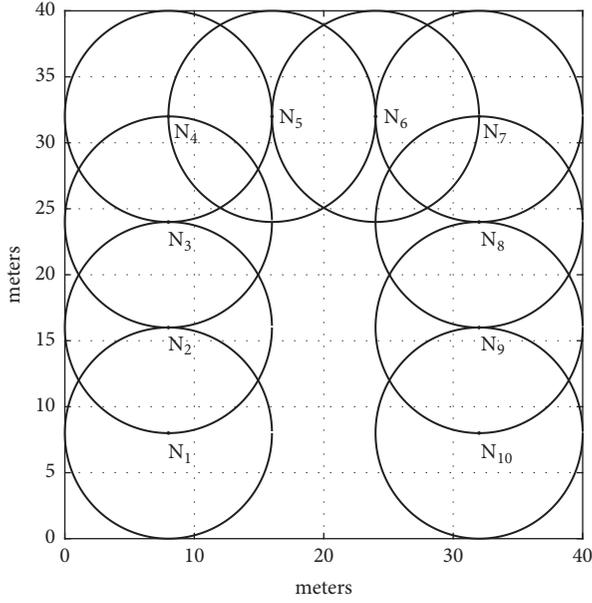


FIGURE 1: Node distribution to calculate upper bound.

upper bounds, an example of 10 nodes distributed in an area of  $40 \times 40 \text{ m}^2$ —see Figure 1—is considered. It is assumed that the sensors can have a transmitting radius between 6 and 8 meters. Details to compute these bounds are described in the following subsections.

**3.1. Energy Lower Bound.** The energy lower bound is easily calculated using (7) considering a radius of 6 meters and  $\mu = 0.005 \text{ mW/m}^2$ . For instance, assuming 10 nodes, the lowest allowed value for energy is  $E_{LB} = (0.005)(10)(6^2) = 1.8 \text{ mW}$ .

**3.2. Area Upper Bound.** The sensor distribution shown in Figure 1 is one among several that provides the nearest value to the maximum coverage area given by 10 nodes using a radius of 8 meters and including maximum three intersections of three circles. The main goal in proposing this case is their comparison to the suboptimal outputs derived of MODEA by having all the nodes' radii equal to 8 meters. If these suboptimal solutions are near to this theoretical case, then the application of MODEA will be considered feasible. Figure 1 is used as the base case model to determine its *theoretical* area upper bound.

At the top corners of Figure 1, the overlap of three circles is observed. Figure 2 shows the details of these three overlapped circles. The covered area of these three circles represents in this paper the coverage area of three overlapped wireless sensors. The set theory helps to determine the total area in a union of sets when some of the sets overlap. To obtain the covered area of the union of three overlapped circles, the inclusion/exclusion rule is used. This rule establishes that the area of the three circles is computed at the beginning. After that, all the intersection areas between two circles  $A_{N_1 \cap N_2}$ ,  $A_{N_1 \cap N_3}$ ,  $A_{N_2 \cap N_3}$  are subtracted. At the end, the intersection

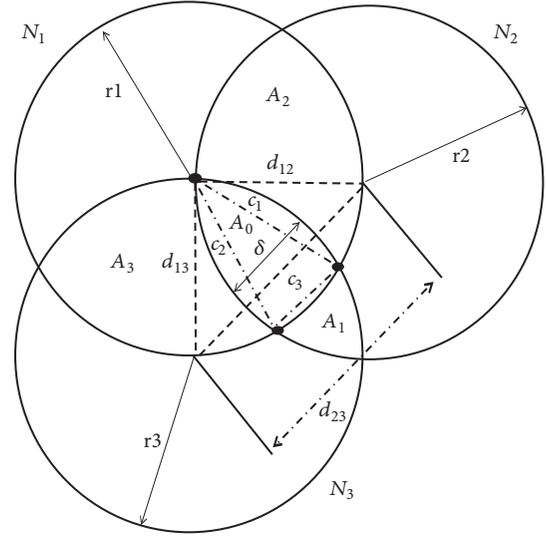


FIGURE 2: Parameters involved in the intersection of three circles.

area of the three circles  $A_{N_1 \cap N_2 \cap N_3}$  is added [57, 58]. The equation of the area of the three circles is

$$A_{3C} = 3\pi r^2 - A_{N_1 \cap N_2} - A_{N_1 \cap N_3} - A_{N_2 \cap N_3} + \dots + A_{N_1 \cap N_2 \cap N_3}, \quad (8)$$

where  $A_{N_1 \cap N_2} = A_0 + A_2$ ,  $A_{N_1 \cap N_3} = A_0 + A_3$ ,  $A_{N_2 \cap N_3} = A_0 + A_1$  and  $A_{N_1 \cap N_2 \cap N_3} = A_0$ .

This result is then used to determine the area of the base case model of 10 sensors. For this model, the area of 10 sensors is computed; then, the nine intersection areas between two circles similar to  $(A_0 + A_3)$  and the two intersection areas between two circles similar to  $(A_0 + A_1)$  are subtracted; and, finally, the two intersection areas of three circles are added.

It is important to remark that for any finite number of overlapped circles, the covered area can be seen as a set of circles with intersections and is obtained by the use of the expression [57] as

$$A_{\text{cov}} = \sum_{i=1}^z A_{N_i} - \sum_{i < j \leq z} A_{N_i \cap N_j} + \dots + \sum_{i < j < k \leq z} A_{N_i \cap N_j \cap N_k} + (-1)^{z+1} A_{\bigcap_{i=1}^z N_i}. \quad (9)$$

According to [59], it is possible to preserve two communication links with an angle of  $5\pi/6$  with the minimum power required, but it is valid for directional communication. For omnidirectional communication, it is not necessary. To obtain the maximum coverage area, it is well known that the hexagon configuration is the ideal distribution, where the angle presented on each vertex is equal to  $2\pi/3$ , but it is valid for circles with equal radius and where there is not a neighbor circle covering each node. Nevertheless, for our case, we have a set of circles with different radius and where each node is covered by at least a neighbor circle.

Now, to find the total area of Figure 1, first find the area of 10 independent circles  $A_{10C}$  given by

$$A_{10C} = 10\pi r^2, \quad (10)$$

and then use the intersection between two circles denoted as  $A_{i2C}$ , [60].

$$A_{i2C} = 2r^2 \arccos\left(\frac{d}{2r}\right) - \frac{1}{2}d\sqrt{4r^2 - d^2}, \quad (11)$$

where  $d$  is the distance between the centers of the two circles. Assume  $r_1 = r_2 = r_3 = r$ ,  $d_{12} = d_{13} = r$ ; therefore  $d_{23} = r\sqrt{2}$ . Observe that Figure 2 shows the variables of this equation where  $\delta$  is the width of the intersection  $A_{N_2 \cap N_3}$  and  $\delta = 2r - D$ . Now assuming an intersection with  $d = r$ , the intersection area  $A_{N_1 \cap N_2}$  ( $A_0$  and  $A_2$ ) or  $A_{N_1 \cap N_3}$  ( $A_0$  and  $A_3$ ) [60],

$$A_{i2C} = 2r^2 \arccos\left(\frac{1}{2}\right) - \frac{1}{2}r^2\sqrt{3}. \quad (12)$$

The area of intersection  $A_{N_2 \cap N_3}$  ( $A_0$  and  $A_1$ ), denoted as  $A_{i2C'}$  is an intersection where  $d = d_{23} = r\sqrt{2}$ , [60],

$$A_{i2C'} = 2r^2 \arccos\left(\frac{1}{\sqrt{2}}\right) - r^2. \quad (13)$$

Figure 2 shows the intersection area between three circles denoted as  $A_0$ . This area is known as a *circular triangle*, which is a triangle with arc shaped sides ( $c_1, c_2, c_3$ ). The area can be obtained using the radius and the length of the three arcs as shown in Figure 2. The area is given by the following equation [61]:

$$A_0 = \frac{1}{4}\sqrt{\kappa\lambda\nu\sigma} + \dots + \sum_{k=1}^3 \left[ r_k^2 \arcsin\left(\frac{c_k}{2r_k}\right) - \frac{c_k}{4}\sqrt{4r_k^2 - c_k^2} \right], \quad (14)$$

where  $\kappa = (c_1 + c_2 + c_3)$ ,  $\lambda = (c_2 + c_3 - c_1)$ ,  $\nu = (c_1 + c_3 - c_2)$ ,  $\sigma = (c_1 + c_2 - c_3)$ ,

$$c_1^2 = 2r^2 - \frac{d^2}{2} + \sqrt{2r^4 - \frac{3r^2d^2}{2} + \frac{d^4}{4}} - \dots - d\sqrt{r^2 - \frac{d^2}{4}} - d\sqrt{\frac{r^2}{2} - \frac{d^2}{4}}, \quad (15)$$

and  $c_1 = c_2$ .

The appendix shows how the equation for  $c_3$  is obtained:

$$c_3 = -\frac{r}{\sqrt{2}} + r\sqrt{\frac{3}{2}}. \quad (16)$$

Therefore, the coverage area upper bound ( $A_{UB}$ ) of ten wireless sensors as shown in Figure 1 is given by

$$A_{UB} = A_{10C} - 9A_{i2C} - 2A_{i2C'} + 2A_0. \quad (17)$$

With the completion of this analysis, it is possible to obtain the coverage area upper bound, 1286.7 m<sup>2</sup>. In Section 5, the values obtained with the DE algorithm are compared to the bounds.

#### 4. The Multiobjective Differential Evolution Algorithm (MODEA) for Wireless Ad Hoc Sensor Distribution

It is well-known that the DE algorithm [20] begins with an initial population of individuals and then iterates to build new populations until a good solution is found. In DE an individual is a vector of the dimension of the problem  $D$ . The individuals characterize specific solutions to the problem under study. In this paper, the vector dimension contains the coordinates of the sensor nodes ( $x, y$ ) followed by their radii values. Considering that there is not information with respect to the optimal solution, the initial population is built in random manner. DE makes repeated cycles of recombination and selection to move the population in the direction of the vicinity of a global optimum. Probability operators (crossing and mutation) are used to each individual in a population to create new individuals (children). The new individuals have some of the features of their ancestors. The ancestors are retained or removed by selection. The term generation is applied to designate the conversion of all individuals into new ones. In other words, to move from one population to another.

The DE algorithm iterates for a limited number of generations,  $G$ . It is important to mention that DE has three key control parameters: the mutation constant  $M$ , which controls the mutation strength, the recombination constant  $C_r$ , and the population size  $N_p$ . The parameters  $M$  and  $C_r$  take values in the interval  $(0, 1)$ . During the course of the execution process, the user establishes the population size  $N_p$ . At each generation, all individuals in the population are evaluated in turn; in DE's literature, the individual being evaluated is named the target vector. Three other individuals are randomly selected from the population and are mixed with each other: this operation is known as mutation and generates a mutant individual vector. The mutant individual vector is then mixed with the current target vector by an operator named recombination: the result of this recombination process is a vector named the trial vector.

Finally, the selection operator is used. If the trial vector improves the objective function then it is accepted and replaces the current target vector in the new population that is being built. Else, it is rejected and the current target vector passes on to the next generation; in this situation the trial vector is not kept.

The following is a description of the operators utilized in the DE algorithm that obtains the most promising region in the search space.

**4.1. Objective Functions.** In this research work, the DE algorithm seeks to achieve a balance between the maximal effective coverage rate  $A_{cov}$  and the minimum sensor power communication  $f_2(x) = E_{csd}$ . To translate the network coverage rate into minimal functions, the opposite effect of the coverage rate is used. Thus, the network effective coverage rate is then defined by

$$\min f_1(x) = 1 - A_{cov}, \quad (18)$$

and the minimum network energy communication is obtained by

$$\min f_2(x) = \mu \sum_{i=1}^z r_i^2. \quad (19)$$

The values of  $\min f_1(x)$  and  $\min f_2(x)$  are part of the fitness function for measuring the result. A linear combination of the objective functions transforms the original multiobjective function into a single-objective function as follows:

$$\min \sum_{k=1}^K w_k f_k(x) \quad \text{where } w_k \geq 0; \sum_{k=1}^K w_k = 1, \quad (20)$$

where  $w_k$  are weight coefficients expressing the relative importance of each objective function and  $k$  is the number of objective functions that form the total objective function  $f$ .

**4.2. The Fitness Function.** The fitness function is given by

$$f(x) = w_1 f_1(x) + w_2 n_2 f_2(x), \quad (21)$$

here  $w_1$  and  $w_2$  represent weight coefficients and  $n_2$  is the normalization coefficient of  $f_2(x)$ . Note that  $f_1(x)$  is already a normalized function (18) due to the fact that it takes only values between 0 to 1. To normalize the second term, it is necessary to use the following equation:

$$n_2 = \frac{1}{\mu z r_{\max}^2}, \quad (22)$$

where  $r_{\max}$  is the maximum sensor radius in meters (m).

Consider  $w_2 = 1 - w_1$ ; therefore if  $w_1 = 1, w_2 = 0$ , then the fitness function optimizes the network effective coverage rate. Conversely if  $w_1 = 0, w_2 = 1$ , then the fitness function optimizes the energy consumption of signal detection. The normalized terms of (21) make it possible to get the same order of magnitude. The overall value of the fitness function is in the interval  $[0, 1]$ . Obviously, the smaller the value of  $f(x)$ , the better the node location distribution and the lower the power communication consumption as well.

**4.3. The Mutation Operator.** For each target vector  $x^i, i = 1, \dots, N_p$ , the mutant individual  $m^i$  is created with the following equation [23]:

$$m^i = x^{r1} + M(x^{r2} - x^{r3}), \quad (23)$$

where  $x^{r1}, x^{r2}, x^{r3} \in \{1, \dots, N_p\}; x^{r1} \neq x^{r2} \neq x^{r3} \neq x^i$ . Here,  $x^{r1}, x^{r2}$ , and  $x^{r3}$  are three random individuals from the population, mutually different and also different from the current target vector  $x^i$ , and  $M$  is a scaling factor named the mutation constant which must be  $M > 0$ . The mutation operator is utilized to manage the magnitude of the difference between two individuals. This operator manages the tradeoff among exploitation and exploration on the search process. This operator is the one guiding the convergence of the

algorithm. This version is known as DE/rand/1 version. Aside, there is another version like the one shown in [20, 22, 62]

$$m^i = x^{\text{best}} + M(x^{r1} - x^{r1}), \quad (24)$$

where the best vector among the population is chosen instead of obtaining it randomly. Expression stated in (24) is known as DE/best/1 [20, 22, 62].

**4.4. The Recombination Operator.** The recombination operator is applied to increment the diversity in the mutation process. This operator is the final step in the creation of the trial vector. To create the trial vector, the mutant individual is combined with the current target vector. In particular, for each component  $j$ , where  $j = \{1, 2, \dots, D\}$ , of the mutant individual, a random number rand in the interval  $[0, 1]$  is selected. Then, the recombination constant  $C_r$  and rand are compared in order to check if  $\text{rand} < C_r$ . Equation (25) briefly shows that if the aforementioned condition is false, the  $j$ th element of the target vector is selected as the  $j$ th element of the trial vector. Otherwise the  $j$ th element of the mutant individual is selected as the  $j$ th element of the trial vector [23].

$$t^{i,j} = \begin{cases} m^{i,j} & \text{if } \text{rand} < C_r \quad \forall j \\ x^{i,j} & \text{otherwise.} \end{cases} \quad (25)$$

The reader is warned that a large value for  $C_r$  helps to avoid the cancelation of the mutation operation. Otherwise,  $C_r$  would become useless in its application on (25).

**4.5. Application of the Prim–Dijkstra Algorithm.** The nodes' locations are validated considering the distance communication constraint. The communication range distance equal to the sensor coverage radius is used. To satisfy this constraint, the Prim–Dijkstra algorithm is applied [51]. The input to the algorithm is a root node from the set of nodes. This root node may act as the information concentrator and is chosen arbitrarily by the network designer. An external controller starts the algorithm and the algorithm takes action based on each actual position of all nodes. The Prim algorithm [63] determines the minimum spanning tree (MST) whereas the Dijkstra algorithm [64] obtains a star topology. Prim–Dijkstra [51, 52] uses the location of the sensor nodes and obtains a topology that is a mixture of MST and a star topology. As mentioned before, information about the shortest sensor mobility is determined with the Hungarian algorithm [53] finding the minimum distance between the initial and the final node positions.

**4.6. The Selection Operator.** The last step of the algorithm is the application of the selection operator, where the fitness computed with the trial vector  $t^i$  is compared to the fitness computed with the target vector  $x^i$ . In a simplified way, (26) summarizes the aforementioned statement, where  $t^i$  is chosen if its computed fitness is lower than the fitness computed on  $x^i$  and vice versa.

$$\text{pop}_i = \begin{cases} t^i & \text{if } f(t^i) < f(x^i) \\ x^i & \text{otherwise.} \end{cases} \quad (26)$$

```

(1) procedure MODEA
(2)   Set the control parameters for MODEA;
(3)   Create initial population;
(4)   Evaluate fitness, area and energy of each member;
(5)   for  $g = 1$  to  $G$  do
(6)     for  $i = 1$  to  $N_p$  do
(7)       Select from  $\text{Pop}_i$  (see version: -R (Eq. (23)), -RM (Eq. (27)), -B (Eq. (24)), -BM (Eq. (28));
(8)       Obtain mutation  $m^i$  (see previous chosen version);
(9)       Apply recombination to obtain  $t^i$  (Eq. (25));
(10)      Validate  $t^i$  according to preliminary bounds;
(11)      Update  $(x, y)$  and radius (Eq. (30)) if needed;
(12)      while a tree is not obtained do
(13)        Validate  $t^i$ ;
(14)        Update  $(x, y)$  and radius (Eq. (30)) if needed;
(15)        Apply MST Prim-Dijkstra to array of links;
(16)        if some nodes are not connected then
(17)          Update position and radius (Eq. (30)) of unlinked nodes;
(18)        else nodes are connected
(19)          Tree has been built;
(20)        end if
(21)      end while
(22)      Apply selection operator (Eq. (26));
(23)    end for
(24)     $\text{Pop}_{\text{best}}$  solution is obtained for fitness, area and energy;
(25)  end for
(26)  Apply Hungarian algorithm to  $\text{Pop}_{\text{best}}$  regarding to  $\text{Pop}_{\text{initial}}$ ;
(27) end procedure

```

ALGORITHM 1: MODEA applied to WAHSN.

Here,  $\text{pop}_i$  is the population of the next generation that changes by accepting or rejecting new individuals. The global best individual is retained at the end of each generation, to keep track of the best solution obtained when DEA is executed.

**4.7. The Random- $M$  Mutation Parameter.** The parameter  $M$  is responsible for the exploration and exploitation of the solution spaces. To fix the problem of stagnation for small values of  $M$ , the random- $M$  parameter is added to (23). This parameter consists of a random number  $\text{rand}$  that multiplies the constant  $M$  at each evaluation, [22, 65]. Equation (27) shows how the main operator changes with this addition:

$$m^i = x^{r^1} + \text{rand}() M (x^{r^2} - x^{r^3}), \quad (27)$$

where  $\text{rand}$  is a random number in the range of ( $0 < \text{rand} < 1$ ) that changes during the evolution process. By multiplying  $M$  by  $\text{rand}$ ,  $M$  modifies its value. In this way,  $\text{rand}$  helps DEA to have a strong reduction of stagnation. Thus, it is also possible to achieve an improved convergence speed, as presented in Section 5. The same approach may be applied to (24) as shown in

$$m^i = x^{\text{best}} + \text{rand}() M (x^{r^1} - x^{r^2}). \quad (28)$$

**4.8. Pseudocode for MODEA in a WAHSN.** Employing these considerations, here the pseudocode for the MODEA is

presented. The algorithm takes the initial parameters of DE, performs the mutation and recombination, validates the coordinates  $(x, y)$  and the radius values, and, if necessary, updates the positions of some nodes. Thereafter, the links are created for each node, proceeding to form a tree, and in the case of nodes not connected to the tree, their positions and radii are updated. After a tree is obtained, the solution may be selected from the best trees generated, based on the best fitness value. The algorithm is considered and denoted as *MODEA* and also is summarized in pseudocode Algorithm 1. The details of the pseudocode are listed as follows:

- (1) MODEA starts with the following parameters:  $M, C_r, N_p, G$ .
- (2) Each individual of population starts as follows:  $\text{Pop}_i = x_{i,1}, \dots, x_{i,z}; y_{i,1}, \dots, y_{i,z}; r_{i,1}, \dots, r_{i,z}$  and  $\text{Pop}_{\text{initial}} = \text{Pop}_1$ . Each  $r_{i,j} = r_{\text{max}}$ .
- (3) Validate  $t^i$  based on coordinates  $(x, y)$  and  $r$  for each member of  $t^i$  is performed according to the boundaries,  $r_{\text{max}}$  and  $r_{\text{min}}$ . If  $r > r_{\text{max}}$  or  $r_{\text{min}} > r$ , (30) is applied to regenerate  $r$ . If  $(x, y)$  are outside the boundaries,  $(x, y)$  are regenerated.
- (4) The Hungarian algorithm receives  $\text{Pop}_{\text{best}}$  (the best result obtained from final population at the end of MODEA) and  $\text{Pop}_{\text{initial}}$  to determine the final position (e) of the sensors regarding the initial positions (i).

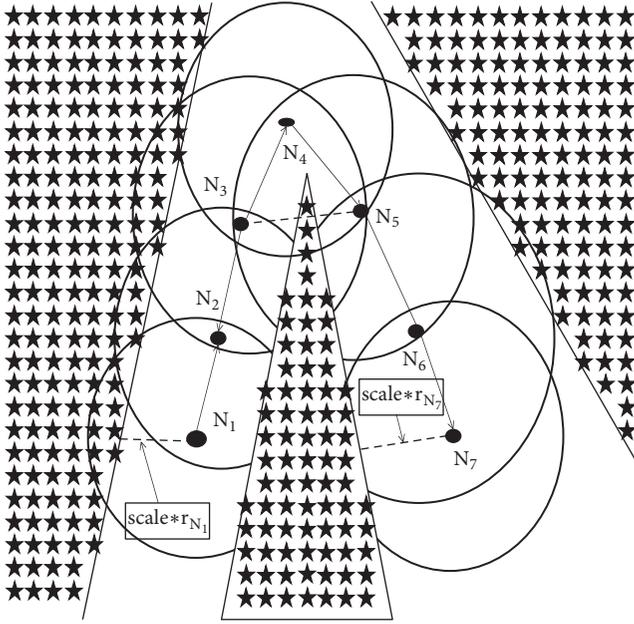


FIGURE 3: Forbidden zone.

- (5) The MST is formed by using central node (cnode = 5),  $\alpha = 0$ , and the distance to generate a link between the nodes ( $d(N_i, N_j)$ ,  $i \neq j$ ) as

$$d(N_i, N_j) = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}. \quad (29)$$

If  $d(N_i, N_j) > \min(r_i, r_j)$  or if there is a forbidden zone between the nodes, the link does not exist.

- (6) For each generation  $g \in \{1, \dots, G\}$ , all the individuals of population are updated according to selection operator.

For simplification purposes, each MODEA version will be treated as follows for the next sections of this manuscript:

MODEA rand (23)  $\rightarrow$  MODEA-R,

MODEA rand (with random- $M$ ) (27)  $\rightarrow$  MODEA-RM,

MODEA best (24)  $\rightarrow$  MODEA-B,

MODEA best (with random- $M$ ) (28)  $\rightarrow$  MODEA-BM.

Figure 3 shows the parameter *scale* which scales the radius of the current node to separate it from the boundaries of the forbidden zone (see nodes  $N_1$  and  $N_7$  with their respective radii  $r_{N_1}$  and  $r_{N_7}$ ) and the limits of the area of interest. Even when  $N_3$  and  $N_5$  are close to each other, if there is a forbidden area between them, these nodes must use  $N_4$  as the communication node. Throughout these restrictions, the nodes are separated while maintaining the MST. To separate the nodes from the main contour area ( $L \times H$ ),  $r_{\min}$  is used instead as a reference.

The MODEA (all rand and best versions and their corresponding random- $M$  versions) adjusts the radius value

according to its rules and the given restrictions. The algorithm uses (30) to generate the radius adjustment  $r_{\text{adj}}$  when validating  $t^i$ . In this algorithm, the radius of each population member starts at its maximum value; as a result, the average radius at the end of the generations is a value above the minimum radius.

$$r_{\text{adj}} = r_{\min} + \text{rand}() (r_{\max} - r_{\min}). \quad (30)$$

## 5. Numerical Results and Discussion

The methodology followed in this manuscript was based on the one presented in [26]. Nevertheless, the treatment of the scenarios and constraints here presented were stated by following the convex configurations [35]. Compared to [26] and other similar works [34], the main differences are that the center of a circle should be inside in at least one of its neighbor circles, where all the circle centers must display a connected graph. The detailed rules are described in Section 4.8. This approach was adopted before testing more elaborated bioinspired techniques for future works, in order to have a baseline and then to determine the feasibility of MODEA with stricter constraints. A previous successful approach with the use of nonconvex constraints and obstacles can be seen in [32]. In future works, such elaborated bioinspired techniques will be tested (more details are available in the Conclusion).

The algorithm was implemented in MATLAB to optimize the distribution of sensors. The effective sensor radius was assigned between  $r_{\min} = 6$  and  $r_{\max} = 8$  meters, except the *multiple-bounded* case, where  $r_{\min} = 4$  m,  $r_{\max} = 6$  m. Different shaped-variant target zones were also tested. The same starting random node distribution was implemented on each scenario, with the purpose of having fair performance comparisons and the computation of the corresponding averages, that is, a unique starting node distribution for squared areas, a unique starting node distribution for polygons, etc., where it is clear that all the starting node distributions are not equal among the analyzed scenarios.

The control parameters were set as  $M = 0.8$ ,  $NP = 35$ , and  $C_r = 0.9$ , where  $M$  and  $C_r$  were assigned taking the best reported performance for both parameters in the open literature [62]. Figure 4 shows the results of the fitness function against the number of generations of *MODEA-R*, *MODEA-RM*, *MODEA-B*, and *MODEA-BM*. These tests are the average of 50 independent tests on a squared area of  $40 \text{ m} \times 40 \text{ m}$ ,  $G = 1000$ , with the maximum possible radius; that is,  $r_{\min} = r_{\max} = 8$  m. Note that *MODEA-BM* outperforms *MODEA-R*, *MODEA-RM*, and *MODEA-B* obtaining the lowest fitness values. In this specific case, a value of  $w_1 = 1.0$  is used to obtain a result near to the area with 10 nodes computed in Section 3.

Comparing the averages obtained among the four versions of MODEA shown in Figure 4, the authors of this manuscript decided to use *MODEA-BM* because the average obtained on all the 50 tested cases is the best. The best case for *MODEA-BM* among these 50 tests was obtained in the test 23/50, where the notation is denoted as *numtest/numfulltests*. This choice was taken even when

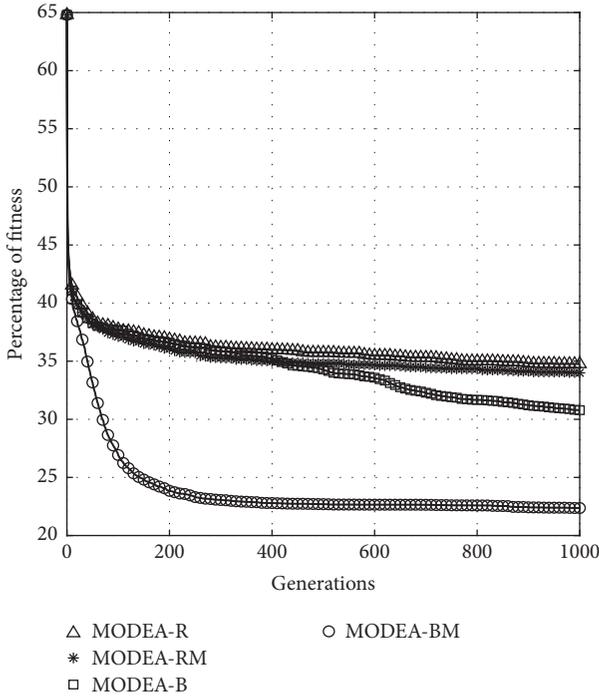


FIGURE 4: Fitness versus generations.

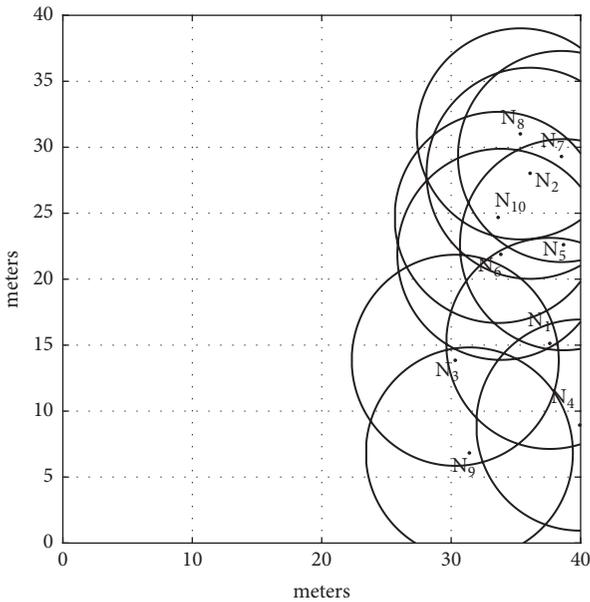


FIGURE 5: Initial random distribution of 10 sensor nodes.

there is a best result obtained with MODEDA-B. Even so, the previous mentioned results are consistent with those presented in [34], where MODEDA-B became the best possible option by considering the classical MODEDA variants presented in [20, 22].

Figure 5 shows the initial position of 10 sensors randomly distributed. The sensor nodes are identified as  $N_1, N_2, \dots, N_{10}$ , and the circles are the sensor ranges of the sensor nodes. As explained before, this starting node

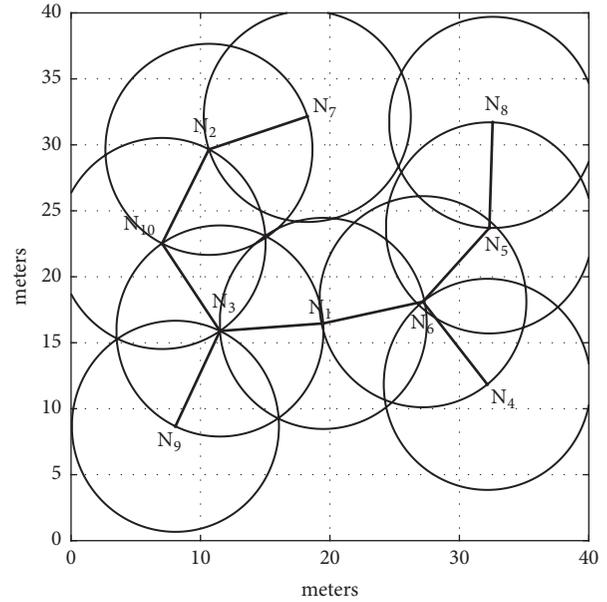


FIGURE 6: Optimized distribution of 10 sensors.

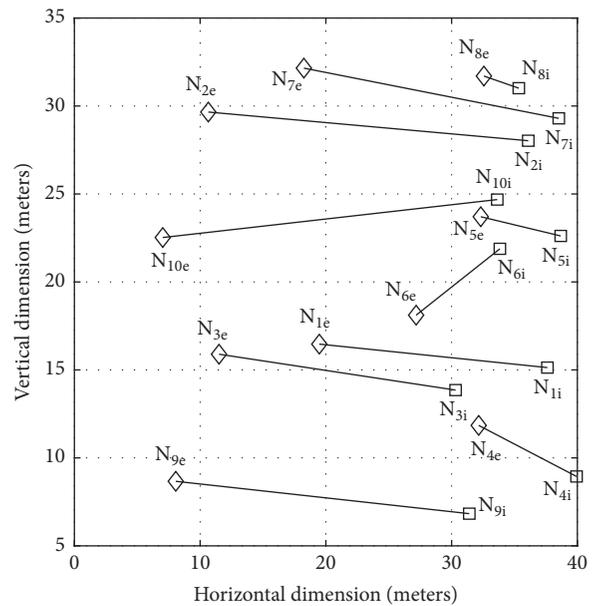


FIGURE 7: Shortest distance from initial (*i*) to final (*e*) node positions generated by the Hungarian algorithm.

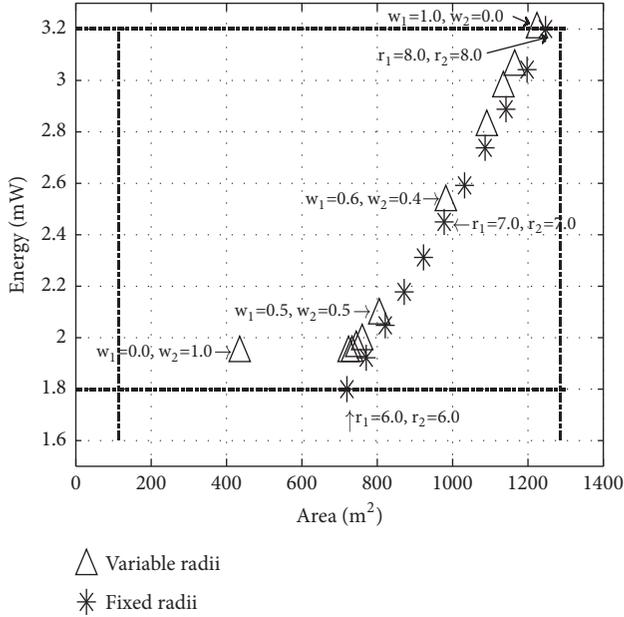
distribution was implemented for all the MODEDA versions for all the 50 tests applied on this squared scenario.

Figure 6 shows the optimized sensor distribution using the initial sensor positions of Figure 5. Also, the wireless communication links between the nodes are shown. The presence of a link between two nodes in the network depends on the relative distance between them (lower than the minimum radius between two nodes).

Figure 7 shows the initial and final position of the sensors nodes, identified with the subindexes (*i*) and (*e*), respectively. Note that the final node positions are at the shortest distance

TABLE 2: Bounds: theoretical model (TM) and MODEA-BM results.

10 Nodes Bounds	Area TM (m <sup>2</sup> )	Energy TM (mW)	Area (m <sup>2</sup> ) MODEA Best	Energy (mW) MODEA Best
Lower	113.09	1.8	-	<b>1.8</b>
Upper	1286.7	3.2	<b>1274.6</b>	3.2


 FIGURE 8: Bounds comparison of 10 nodes, varying  $w_1$  and  $w_2$ .

from the initial node positions. For mobile sensors, this information about mobility is useful to globally minimize sensor path distance. A minimum distance matching algorithm after MODEA-BM is used to obtain the minimum global traveled distance between the initial and the final node positions through the Hungarian algorithm, [53].

Table 2 shows a summary of the results obtained using the lower energy and the upper area bound equations presented in Section 3. Also, the results obtained with the MODEA-BM are shown for comparison purposes (case 23/50). The results obtained with the use of MODEA-BM are within 99.05% of the theoretical bounds ( $1274.6 \text{ m}^2/1286.7 \text{ m}^2$ ). With the use of the random- $M$  parameter, the MODEA-BM algorithm has a high convergence speed and stagnation is avoided, as shown in (27). The upper bounded area is 80.41% of the total area to be covered ( $1286.7 \text{ m}^2/1600 \text{ m}^2$ ); the computed area with MODEA-BM is  $1274.6 \text{ m}^2$  which represents 79.66% of the total area to be covered ( $1274.6 \text{ m}^2/1600 \text{ m}^2$ ).

Now, in Figure 8 an analysis of the weights  $w_1$  and  $w_2$  of (30) and how these values affect the optimization of energy and coverage area is presented for MODEA-BM. It was done by following the methodology presented in [65, 66]. Here,  $w_1$  varies from 0 to 1 in increments of 0.1; meanwhile the value of  $w_2$  varies from 1 to 0 so that  $w_2 = 1 - w_1$ . The fixed radii curve (\*) shows the bounds comparison when the values of

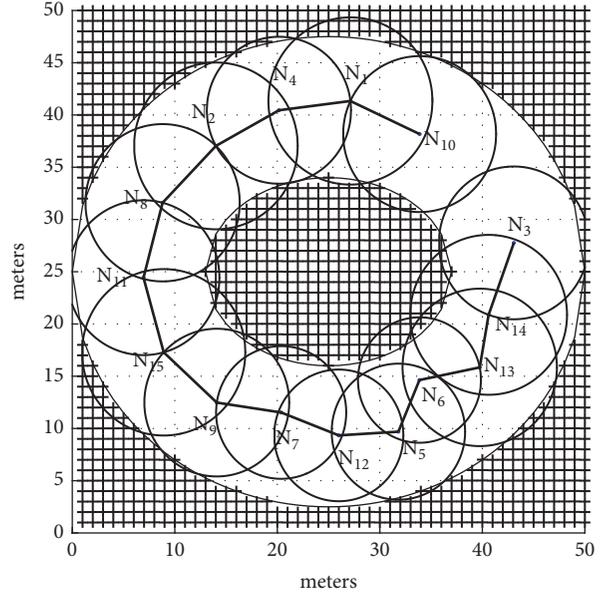


FIGURE 9: Area formed with two ellipses.

radii between  $r_{\max}$  and  $r_{\min}$  are equal. The variable radii curve ( $\Delta$ ) shows the behavior of bounds comparison when the value of radii varies between  $r_{\min} = 6 \text{ m}$  and  $r_{\max} = 8 \text{ m}$ . Here, the observed trend is to maximize the coverage area; nevertheless the restrictions help to reduce the radii of the nodes. It was observed that values of  $w_1 = 0.6$  and  $w_2 = 0.4$  are able to produce results in which the covered area is maximized and, meanwhile, the energy is minimized. This is because those values are near to the case where  $r_{\min} = r_{\max} = 7 \text{ m}$  for the fixed scenario. The results presented here are the average of 50 independent tests of  $G = 1000$  for each point at each scenario. The tests to obtain the point  $w_1 = 1.0, w_2 = 0.0$  were performed independently of those used to compute the output for Figure 6.

**5.1. Different Scenarios of Testing.** Now the results of the algorithm MODEA-BM parameter applied to the sensor distribution problem in different shaped-bounded areas are presented. The number of nodes is obtained by incrementing the number of sensors depending on the coverage area. Using  $w_1 = 0.6, w_2 = 0.4$ , and  $G = 1000$ , MODEA-BM was able to find good sensor distributions. For *multiple-bounded case*,  $G = 2500$ . Restrictions were tuned and set to obtain a suboptimal result for each of the following scenarios. Figures 11–16 were obtained by MATLAB with the command *inpolygon* as described in the corresponding subsections. The best minimum fitness was considered for all the presented cases.

**5.1.1. Scenario 1. Ellipse.** To form the ellipse area shown in Figure 9, the parameters are  $h_A = h_B = 25, k_A = k_B = 25, r_A = 15, r_B = 25, a_A = 0.8, a_B = 1.0, b_A = 0.6, b_B = 0.9$ , where  $A$  is the inner ellipse and  $B$  is the external ellipse, displayed into an area of  $50 \text{ m} \times 50 \text{ m}$ . The corresponding equations are listed as follows:

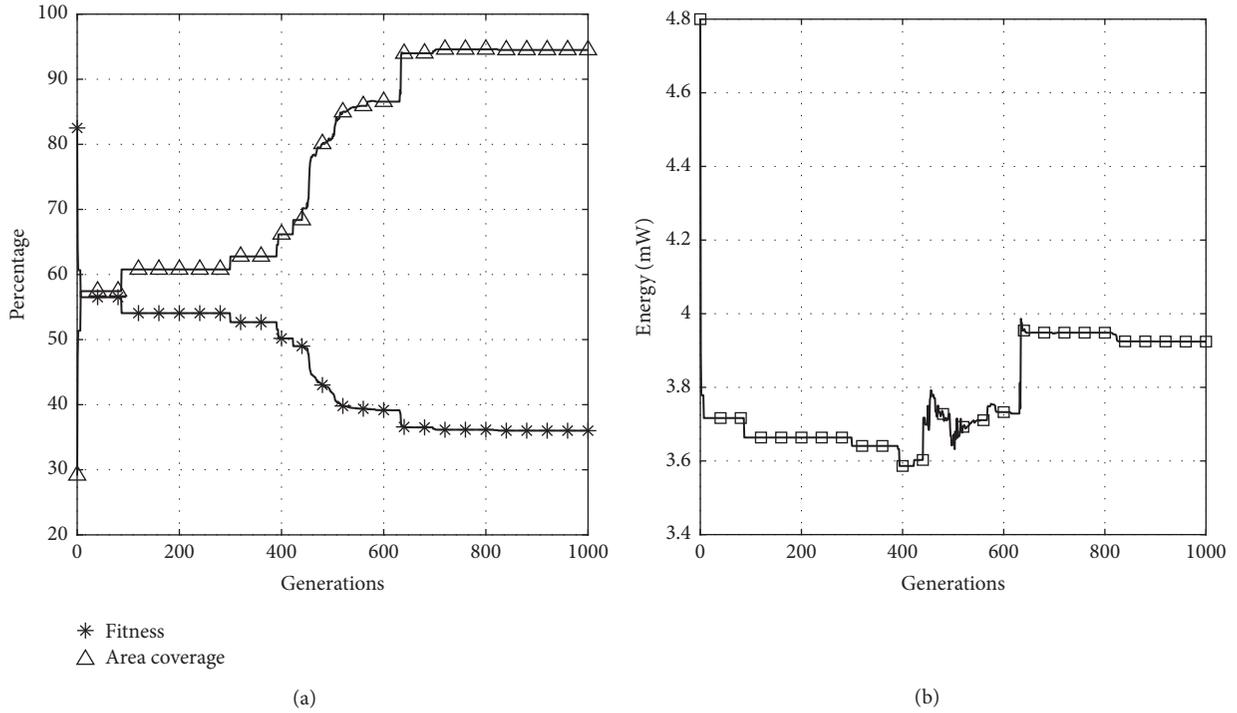


FIGURE 10: Results for (a) fitness, area, and (b) energy consumption on the scenario with two ellipses.

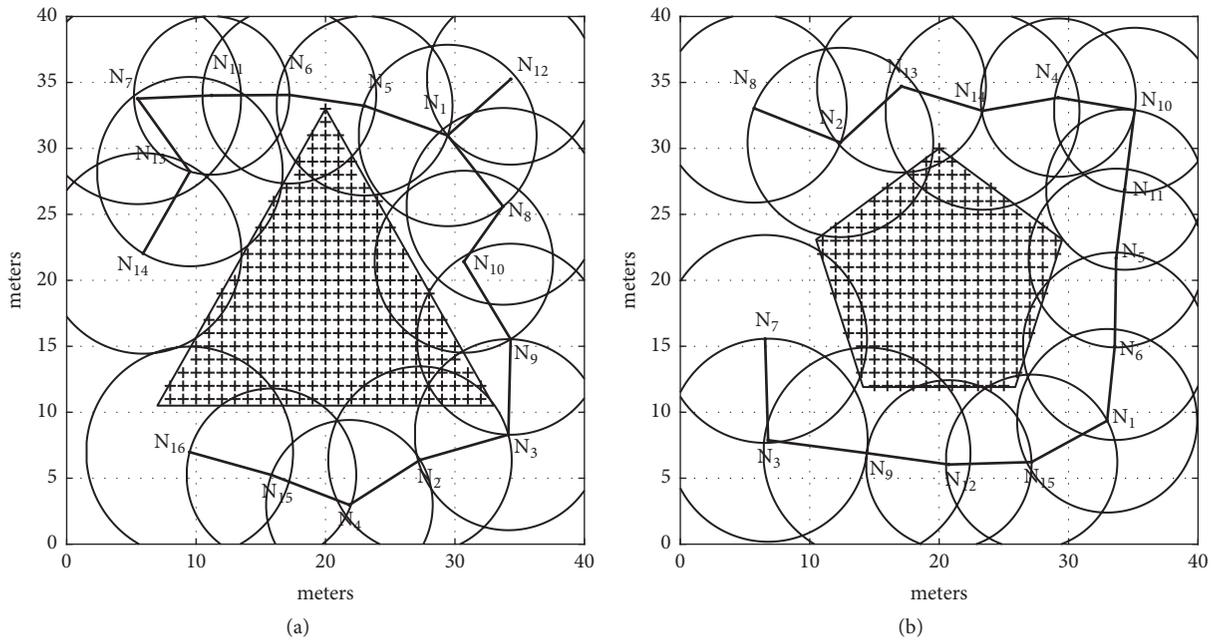


FIGURE 11: (a) Triangle and (b) pentagon scenarios, best output.

$$\frac{(x - h_A)^2}{a_A^2} + \frac{(y - k_A)^2}{b_A^2} = r_A^2, \tag{31}$$

$$\frac{(x - h_B)^2}{a_B^2} + \frac{(y - k_B)^2}{b_B^2} = r_B^2.$$

It is observed in Figure 9 that the circles follow the contour of the given ellipses, maximizing the coverage area. The best minimum fitness was obtained for the test 9/50.

Figure 10 shows the total energy consumption of the best curved-bounded scenario. The covered area is more than 90% while the energy is below 4 units. The fitness curve indicates

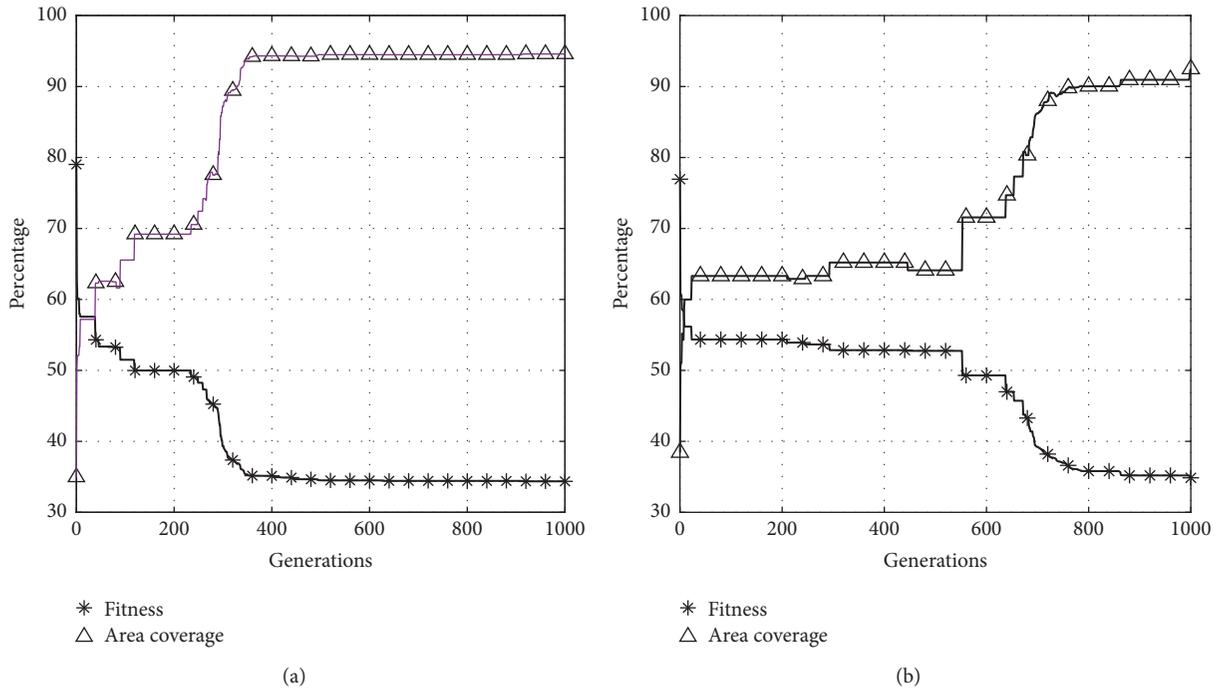


FIGURE 12: (a) Triangle and (b) pentagon scenarios, fitness and area curve.

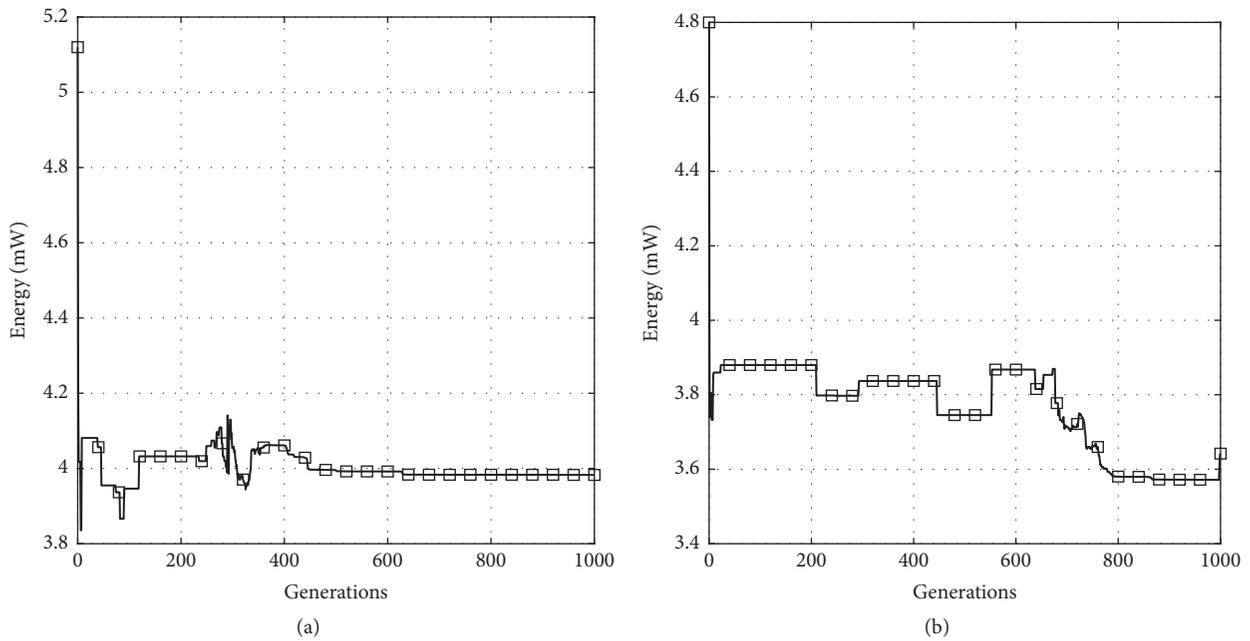


FIGURE 13: (a) Triangle and (b) pentagon scenarios, energy curve.

how both parameters are evolving to be optimized. After 700 generations, the fitness curve becomes stable, maintaining the area/energy limit values also stable.

The line resolution between two nodes was left as 0.01. The complete circle contour was considered to determine the

separation of the boundaries. That is, the angles to compute the radius circle were set as 0 : 0.01 : 360.

5.1.2. *Regular Polygons, External Area.* MODEA-BM is applied to the pentagon shown in Figure 11(b) and also to the

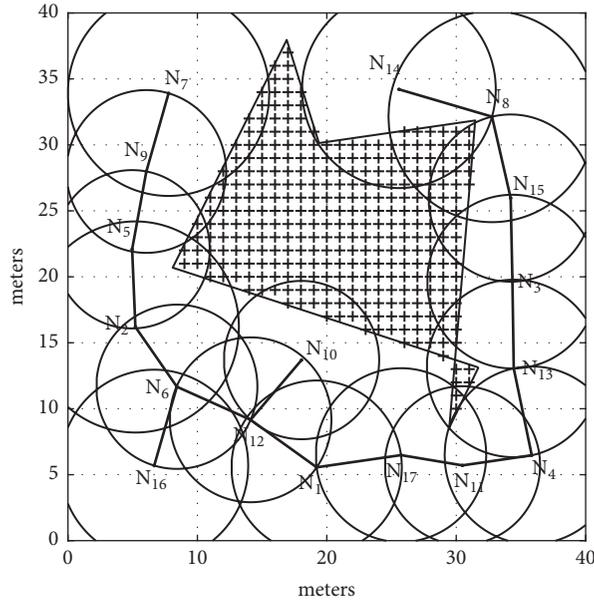
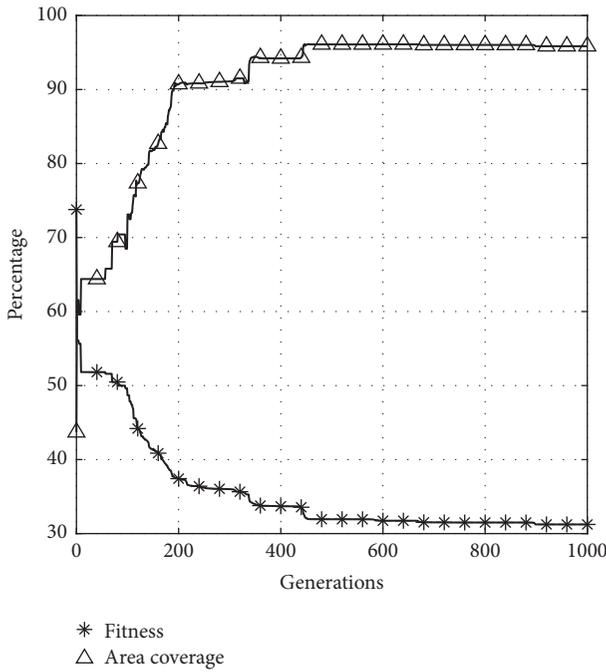
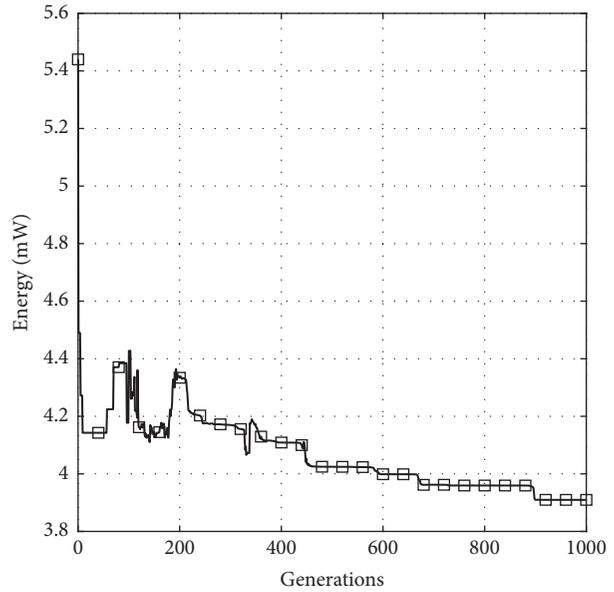


FIGURE 14: Irregular area outside an amorphous zone.



(a)



(b)

FIGURE 15: (a) Area, fitness, and (b) energy curves for amorphous figure.

triangle shown in Figure 11(a). The figures here exposed are obtained using the expression  $LT = \text{linspace}(0, 2\pi, \text{points})$  and the equations given below:

$$\begin{aligned} x &= A \left[ \cos \left( LT + \frac{\pi}{2} \right) \right] + U, \\ y &= A \left[ \sin \left( LT + \frac{\pi}{2} \right) \right] + V, \end{aligned} \quad (32)$$

where points = 4,  $A = 15$ ,  $U = 20$ ,  $V = 18$  in Figure 11(a) and points = 6,  $A = 10$ ,  $U = V = 20$  in Figure 11(b). Vectors  $x$  and  $y$  define the vertices of the polygon. Both polygons are displayed into an area of 40 m × 40 m.

Figure 12 shows the area and fitness obtained for both scenarios. While for triangle the fitness converges before  $G = 400$ , for pentagon the fitness delays to converge before  $G = 900$ . For both cases, the coverage area is greater than 90%.

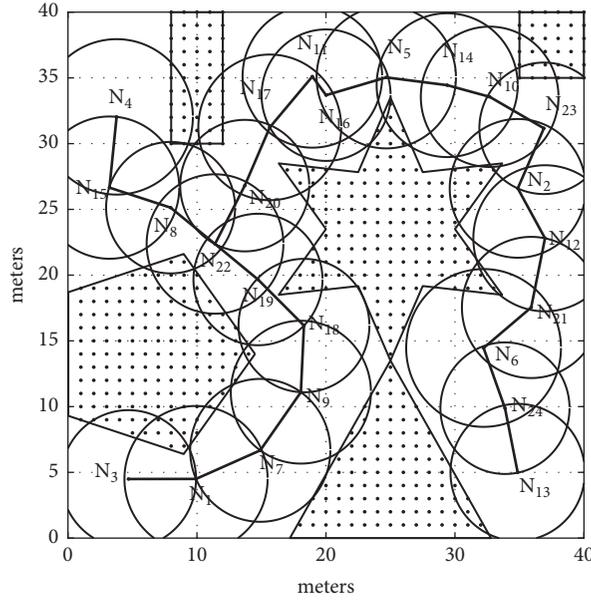


FIGURE 16: Multiple figures scenario.

Figure 13 shows the energy obtained for both scenarios. While for triangle the energy becomes stable after  $G = 400$  with 4 units, for the pentagon the energy holds the same after  $G = 800$  with 3.6 units.

The best minimum fitness was obtained as 41/50 for triangle and 45/50 for pentagon. To speed up the algorithm, each circle was sampled in angles as  $0 : 45 : 360$  to compute the radius. The line resolution between two nodes was left as 0.001 in order to detect forbidden crosses on the shaded area.

**5.1.3. Irregular External Area.** In this scenario, MODEA-BM is applied to the irregular area shown in Figure 14, which is obtained with

$$\begin{aligned}
 x &= \frac{(2H) x}{2} \\
 y &= \frac{(2H) y}{2} + 1,
 \end{aligned}
 \tag{33}$$

where the user can choose the values of vectors  $x$  and  $y$  to generate the irregular polygon of Figure 14. This scenario is presented to show how the algorithm is able to work with true irregular areas. Figure 14 is formed by two linked irregular polygons. The figure is displayed into an area of  $40 \text{ m} \times 40 \text{ m}$ .

As shown in Figure 14, the algorithm prevents that links of the MST from crossing the small polygon, also keeping a distance between the nodes and the entire polygon. With 17 nodes, it is possible to cover more than 90% of the amorphous figure (Figure 15(a)) while the energy is below 4 units (Figure 15(b)). The best minimum fitness was obtained as 10/50.

To speed up the algorithm, each circle was sampled in angles as  $0 : 90 : 360$  to compute the radius. The line resolution between two nodes was left as 0.001 to check if a link crosses forbidden areas, specially the small triangle of the scenario.

**5.1.4. Multiple-Bounded Area.** A multiple-bounded area is shown in Figure 16 with a different radii values,  $r_{\min} = 4 \text{ m}$ ,  $r_{\max} = 6 \text{ m}$ , for speeding up the computation of results. The number of tests here are equal to 20. All the polygons are displayed into an area of  $40 \text{ m} \times 40 \text{ m}$ . This case shows how the algorithm can be adapted to cover an area with multiple bounds (forbidden areas). These bounds represent a triangle, pentagon, star, and two rectangles. For instance, to get the star figure, the following parameters:  $v = 12$ ,  $t = (-1/4 : 1/v : 3/4) * 2 * \pi$ ,  $r_1 = 10$ ,  $r_2 = 5$ ,  $p = (0 : v)$ ,  $r = (r_1 + r_2)/2 + (r_1 - r_2)/2 * (-1)^p$  and (34) are used

$$\begin{aligned}
 x &= r \cos(t) + 25, \\
 y &= r \sin(t) + 23.5.
 \end{aligned}
 \tag{34}$$

Here, the algorithm is able to arrange the obtained MST as a contour to the given bounds, preventing MST links from crossing forbidden areas. This effect is seen in the case of the star, where the links of the nodes do not cross the peaks of the star. A similar effect is seen for the rectangles, where the nodes forming the circle go nearly in parallel with 24 nodes. This property is maintained even in the presence of the thinnest region that connects the star and the triangle, where there is no link that crosses it. Thus, it is possible to cover more than 90% of the area of Figure 16 (see Figure 17(a)) with less than 3.5 units of energy (see Figure 17(b)).

The best minimum fitness was obtained as 2/20. To speed up the algorithm, each circle was sampled in angles as  $0 : 30 : 360$  to compute the radius and the line between two nodes was sampled with 0.1 of resolution. When constructing the final MST, that line is sampled with 0.001 of resolution to detect if it crosses forbidden zones. The *scale* parameter guarantees that all the cases do not have issues with the line resolution when finishing MODEA-BM.

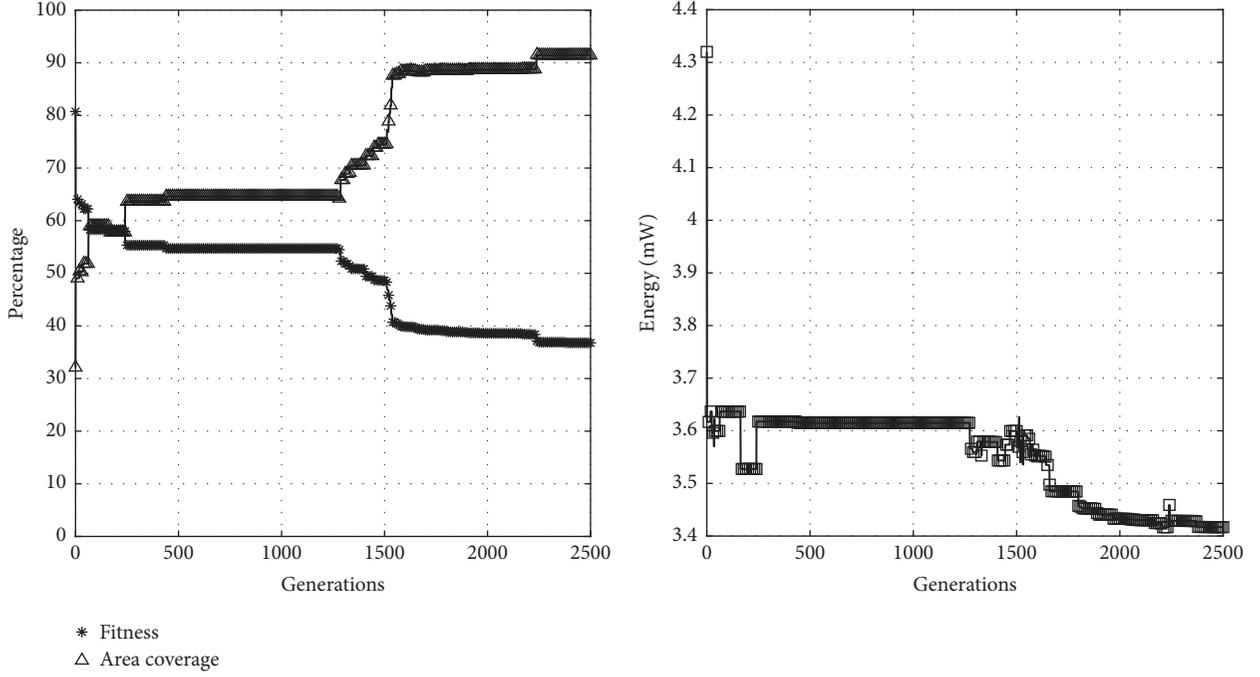


FIGURE 17: (a) Area, fitness, and (b) energy curves for multiple figures.

**5.1.5. Summary of the Results.** Table 3 shows a summary of the average area results obtained from the different figures comparing them with the maximum area to be covered. The results obtained using MODEA-BM parameter are very close to the area to be covered. On each case, 50 tests were made to obtain their mean and standard deviation, except for the multiple case, where only 20 tests were done. The energy for all the cases remains below 4 units, while the coverage area is greater than 90% for the best solutions obtained. The computation of the average and standard deviation on the area parameter shows that the *irregular external area* case remains greater than 90% with the lowest standard deviation, which is consistent with the fitness value (34.98%). In contrast, the *multiple* case shows the worst average (77.35%) and the largest standard deviation (135.10) and also presents the largest fitness value (46.17%) with the largest standard deviation (8.00). Even so, for *multiple* case, MODEA-BM is able to obtain results where the covered area is greater than 90%.

Table 4 shows the comparison of the DEA here proposed to well-known multiobjective optimization algorithms such as NSGA-II and MOEA/D. The authors modified the generic versions of both algorithms, where the corresponding source codes are available in [67]. The default configuration parameters were left unmodified except  $Np = 35$ . Here, only the base case scenario was tested. For the sake of consistency and fairness, the same 50 tests were done for the following presented cases. For MODEA, the radii-variable coordinate obtained with  $w_1 = 0.6$  and  $w_2 = 0.4$  was included for the comparison (see Figure 8). For the other multiobjective algorithms, four different scenarios were tested. Here,  $g_1(x)$

is the nonnormalized version of  $f_1(x)$  and  $g_2(x)$  is the nonnormalized energy version of  $f_2(x)$ . Additionally,  $A_{\max}$  is the area to be covered and  $E_{\max}$  is given by (22). These variants are described as follows:

- (i) Version-I (V-I): the coordinates of the algorithm are tested without normalizing them. That means

$$g(x) = [g_1(x) \quad g_2(x)]. \quad (35)$$

- (ii) Version-II (V-II): the coordinates of the algorithm are tested, but in a normalized way as

$$g(x) = \left[ \frac{g_1(x)}{A_{\max}} \quad \frac{g_2(x)}{E_{\max}} \right]. \quad (36)$$

- (iii) Version-III (V-III): each normalized coordinate is weighted by 0.5 as

$$g(x) = \left[ \frac{0.5g_1(x)}{A_{\max}} \quad \frac{0.5g_2(x)}{E_{\max}} \right]. \quad (37)$$

- (iv) Version-IV (V-IV): the normalized area ( $g_1(x)$ ) is weighted by 0.6. Meanwhile, the normalized energy ( $g_2(x)$ ) is weighted by 0.4 as

$$g(x) = \left[ \frac{0.6g_1(x)}{A_{\max}} \quad \frac{0.4g_2(x)}{E_{\max}} \right]. \quad (38)$$

TABLE 3: Area to cover and area covered by MODEA-BM, plus energy consumption. Averages, standard deviation, and best cases, where BC means best case,  $\bar{\mu}$  is the average, and  $\sigma$  is the standard deviation.

Figure Number	Data	Fitness Percent %	Fitness $\sigma$	Area (m <sup>2</sup> ) MODEA-BM	Area (m <sup>2</sup> ) to cover	Area Percentage %	Area $\sigma$	Energy (mW)	Energy $\sigma$	Scale parameter	Number of Nodes
Figure 9	Stats ( $\bar{\mu}, \sigma$ )	42.25	5.04	1151.63	1427.72	80.66	124.16	3.67	0.19	0.500	15
	BC 9/50	36.00	-	1349.12	-	94.49	-	3.92	-	-	-
Figure 11(a)	Stats ( $\bar{\mu}, \sigma$ )	37.30	1.97	1120.02	1306.40	85.73	59.75	3.67	0.13	0.500	16
	BC 41/50	34.36	-	1235.63	-	94.58	-	3.98	-	-	-
Figure 11(b)	Stats ( $\bar{\mu}, \sigma$ )	39.76	4.66	1136.88	1362.71	83.42	103.41	3.57	0.18	0.200	15
	BC 45/50	34.86	-	1260.21	-	92.47	-	3.64	-	-	-
Figure 14	Stats ( $\bar{\mu}, \sigma$ )	34.98	1.80	1159.72	1288.5	90.00	53.79	3.94	0.15	0.200	17
	BC 10/50	31.23	-	1235.06	-	95.85	-	3.90	-	-	-
Figure 16	Stats ( $\bar{\mu}, \sigma$ )	46.17	8.00	871.15	1126.21	77.35	135.10	3.51	0.13	0.400	24
	BC 2/20	36.78	-	1029.67	-	91.42	-	3.41	-	-	-

TABLE 4: Comparison of DEA algorithm with some known multi-objective algorithms. Average on 50 tests.

Algorithm	Area (m <sup>2</sup> ) mean	Energy (mW) mean
MODEA (ours) ( $w_1 = 0.6, w_2 = 0.4$ )	981.9858	2.5266
NSGA-II V-I	1028.5674	2.8120
NSGA-II V-II	667.3309	2.1155
NSGA-II V-III	721.3496	1.9494
NSGA-II V-IV	875.0040	2.3059
MOEA/D V-I	868.0113	2.9287
MOEA/D V-II	708.6827	1.9229
MOEA/D V-III	648.9909	2.0030
MOEA/D V-IV	809.0555	2.3831

According to the results shown in Table 4, MODEA presents the best average performance among all the cases, with exception of NSGA-II version V-I. NSGA-II V-I obtains a coverage area 4.53% larger than MODEA; however the energy is 10.15% larger. It is shown that MODEA is more efficient in terms of the energy consumption; meanwhile NSGA-II increases the coverage area but raises the energy consumption. The rest of the multiobjective algorithms versions do not reach a similar area to that obtained by MODEA.

## 6. Conclusion

Firstly, a base case was thoroughly studied in a theoretical way to determine the minimum and maximum limits of that base configuration. This was done to show how MODEA is able to find a suboptimal solution near to its maximum theoretical configuration with great accuracy (99.05%). At the same time on the same base case, several feasible optimization strategies were tested with the use of MODEA, denoted as MODEA-R, MODEA-RM, MODEA-B, and MODEA-BM. It can be seen that the addition of the random- $M$  parameter in Storn's algorithm helps to converge toward the optimal solution quickly, particularly for MODEA-BM. For that reason, MODEA-BM was chosen to evaluate the rest of the cases here presented, i.e., different area shapes, either one shape or multiple shapes. Results obtained with 1000 iterations and  $w_1 = 0.6, w_2 = 0.4$  showed that more than 90% of the targeted area is covered for the best outputs, except for *multiple* case. For a highly restricted scenarios like *multiple* case, the best outputs showed that the suboptimal solution is reached in 2500 iterations, even when in general the low average (77.35%) shows that some of the cases do not reach the desired output. The MST was adapted to avoid crossing forbidden areas, creating feasible communication trees while optimizing the objective function. Nevertheless, the readers are warned that an adequate restriction tuning leads to a feasible solution, especially in cases where the area shapes are irregular, multiple, or a combination of both. Thus, MODEA-BM prevents the existence of nodes in forbidden areas, near these areas' borders, or near each other with a certain tolerance, thus maximizing the covered area and reducing energy consumption. MODEA-BM is complemented with

the use of the Prim–Dijkstra algorithm to find on-the-fly the best possible MST for each population member. MODEA-BM is also complemented with the Hungarian algorithm at the end in order to find which movements should be performed to arrange the initial to the end node positions.

As further work, the following lines of research can be done: (1) The application of the MODEA algorithm should be investigated in real indoor coverage areas considering wireless propagation effects such as shadowing, interference, obstacles, and multipath. Those restrictions shown by [17, 18] will complement the results presented in this work. (2) With some further adaptations, MODEA could be applied for guiding autonomous sensorial robots in an unexplored geographical area or (3) in search and rescue operations, where it is common to have different sensor nodes with different transmission rates, ranges, and sampling rates. (4) It would be interesting to compare MODEA against other well-known bioinspired algorithms such as Ant Colony Optimization or the latest MODEA versions found in the open literature (jDE [68], JADE [69], SADE [70], and the like [62]) and obtain similar results like those shown in [34]. (5) A simulation to evaluate MODEA should be done using random variables to generate irregular coverage areas and simultaneously determine its impact in forming links in an ad hoc network, reflected in the construction of routing tables.

## Appendix

Figure 18 given in [61] shows the intersection of three circles called a circular triangle. This figure is used to obtain (14) and (16), which are used to obtain (16), which is the area of the circular triangle. The vertices of the circular triangle are labeled by  $(x_{ij}, y_{ij})$ .

The separation between the centers of circle 1 and circle 2,  $d_{12}$ , must satisfy

$$r_1 - r_2 \leq d_{12} \leq r_1 + r_2. \quad (\text{A.1})$$

If not satisfied, then there is no circular triangle. However, if satisfied, the circles intersect in two points. Each intersection point satisfies the equation of these two circles.

$$\begin{aligned} x_{12}^2 + y_{12}^2 &= r_1^2, \\ (x_{12} - d_{12})^2 + y_{12}^2 &= r_2^2, \end{aligned} \quad (\text{A.2})$$

where  $(x_{12}, y_{12})$  are the coordinate variables of the intersection points that correspond to

$$\begin{aligned} x_{12} &= \frac{r_1^2 - r_2^2 + d_{12}^2}{2d_{12}}, \\ y_{12} &= \frac{1}{2d_{12}} \sqrt{2d_{12}^2 (r_1^2 + r_2^2) - (r_1^2 - r_2^2)^2 - d_{12}^4}. \end{aligned} \quad (\text{A.3})$$

Now considering the third circle and using the coordinate system of  $(x', y')$ , the equations of the intersection point

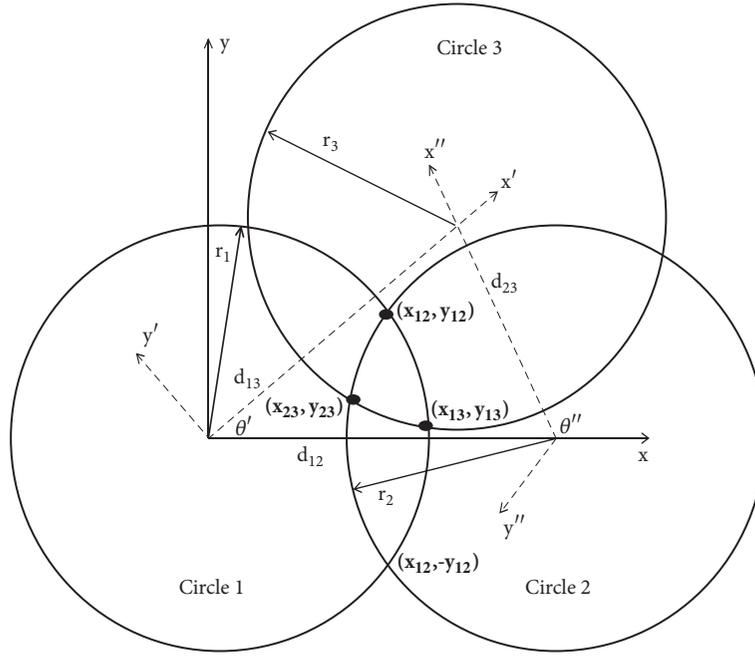


FIGURE 18: Vertices of the circular triangle.

between circle 1 and circle 3 are derived. Note that  $y'_{13}$  has a negative value.

$$\begin{aligned} x'_{13} &= \frac{r_1^2 - r_3^2 + d_{13}^2}{2d_{13}}, \\ y'_{13} &= \frac{-1}{2d_{13}} \sqrt{2d_{13}^2 (r_1^2 + r_3^2) - (r_1^2 - r_3^2)^2 - d_{13}^4}. \end{aligned} \quad (\text{A.4})$$

Assuming that  $r_1 = r_2 = r_3 = r = d$ ,  $d_{12} = r\sqrt{2}$ ,  $d_{13} = r$ , and  $d_{23} = r$ , it is found that

$$\begin{aligned} x'_{13} &= \frac{r}{2}, \\ y'_{13} &= \frac{-r\sqrt{3}}{2}. \end{aligned} \quad (\text{A.5})$$

Now, considering that

$$\begin{aligned} x_{13} &= x'_{13} \cos \theta' - y'_{13} \sin \theta', \\ y_{13} &= x'_{13} \sin \theta' + y'_{13} \cos \theta' \end{aligned} \quad (\text{A.6})$$

and that

$$\begin{aligned} \cos \theta' &= \frac{d_{12}^2 + d_{13}^2 - d_{23}^2}{2d_{12}d_{13}}, \\ \sin \theta' &= \sqrt{1 - \cos^2 \theta'} \end{aligned} \quad (\text{A.7})$$

and substituting values, the following expressions are obtained

$$\begin{aligned} \cos \theta' &= \frac{\sqrt{2}}{2}, \\ \sin \theta' &= \frac{\sqrt{2}}{2}, \\ x_{13} &= r \left( \frac{\sqrt{2}}{4} + \frac{\sqrt{6}}{4} \right), \\ y_{13} &= r \left( \frac{\sqrt{2}}{4} - \frac{\sqrt{6}}{4} \right). \end{aligned} \quad (\text{A.8})$$

Using the following equations

$$\begin{aligned} x''_{23} &= \frac{r_2^2 - r_3^2 + d_{23}^2}{2d_{23}}, \\ y''_{23} &= \frac{1}{2d_{23}} \sqrt{2d_{23}^2 (r_2^2 + r_3^2) - (r_2^2 - r_3^2)^2 - d_{23}^4}, \\ x_{23} &= x''_{23} \cos \theta'' - y''_{23} \sin \theta'' + d_{12}, \\ y_{23} &= x''_{23} \sin \theta'' + y''_{23} \cos \theta'', \\ \cos \theta'' &= \frac{-d_{12}^2 + d_{23}^2 - d_{13}^2}{2d_{12}d_{23}}, \\ \sin \theta'' &= \sqrt{1 - \cos^2 \theta''} \end{aligned} \quad (\text{A.9})$$

and substituting values, the following results are obtained

$$\begin{aligned}\cos \theta' &= \frac{\sqrt{2}}{2}, \\ \sin \theta' &= \frac{\sqrt{2}}{2}, \\ x_{13} &= r \left( \frac{\sqrt{2}}{4} + \frac{\sqrt{6}}{4} \right), \\ y_{13} &= r \left( \frac{\sqrt{2}}{4} - \frac{\sqrt{6}}{4} \right).\end{aligned}\quad (\text{A.10})$$

The use of (16) is to verify that circle 3 forms a circular triangle under the conditions

$$\begin{aligned}(x_{12} - d_{13} \cos \theta')^2 + (y_{12} - d_{13} \sin \theta')^2 &< r_3^2, \\ (x_{12} - d_{13} \cos \theta')^2 + (y_{12} + d_{13} \sin \theta')^2 &> r_3^2.\end{aligned}\quad (\text{A.11})$$

If these conditions are satisfied, then there is a circular triangle. Now, substituting values,

$$\begin{aligned}\cos \theta'' &= \frac{-\sqrt{2}}{2}, \\ \sin \theta'' &= \frac{\sqrt{2}}{2}, \\ x_{23} &= r \left( \sqrt{2} - \frac{\sqrt{2}}{4} - \frac{\sqrt{6}}{4} \right), \\ y_{23} &= r \left( \frac{\sqrt{2}}{4} - \frac{\sqrt{6}}{4} \right).\end{aligned}\quad (\text{A.12})$$

Now, the length of the chords  $c_1$ ,  $c_2$ ,  $c_3$  is found through

$$c_k^2 = (x_{ik} - x_{jk})^2 + (y_{ik} - y_{jk})^2. \quad (\text{A.13})$$

For the case of  $c_3$

$$c_3^2 = (x_{13} - x_{23})^2 + (y_{13} - y_{23})^2. \quad (\text{A.14})$$

Substituting values,  $c_3$  is given by

$$c_3 = -\frac{r}{\sqrt{2}} + r\sqrt{\frac{3}{2}}. \quad (\text{A.15})$$

This is (16) of Section 3 assuming  $d = r$  and is different from the value of  $c_3$  given in [61].

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

Armando Céspedes-Mota gives thanks to CONACyT-México, the research council of Mexico, for the scholarship granted to him. This research was partially supported by the School of Engineering and Sciences of Tecnológico de Monterrey.

## References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.
- [2] P. Santi, "Topology control in wireless ad hoc and sensor networks," *ACM Computing Surveys*, vol. 37, no. 2, pp. 164–194, 2005.
- [3] I. Mahgoub and M. Ilyas, *Sensor Network Protocols*, CRC Press, 2006.
- [4] R. V. Kulkarni and G. K. Venayagamoorthy, "Particle swarm optimization in wireless-sensor networks: a brief survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 41, no. 2, pp. 262–267, 2011.
- [5] H. Ali, W. Shahzad, and F. A. Khan, "Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization," *Applied Soft Computing*, vol. 12, no. 7, pp. 1913–1928, 2012.
- [6] A. S. Majid and E. Joelianto, "Optimal sensor deployment in non-convex region using discrete particle swarm optimization algorithm," in *Proceedings of the 1st IEEE Conference on Control, Systems and Industrial Informatics (ICCSII '12)*, pp. 109–113, IEEE, Bandung, Indonesia, September 2012.
- [7] Y. Chen, C. Chin, and D. Deng, "Efficient localization algorithm in wireless ad hoc sensor networks by utilizing radical centers," in *Proceedings of the ICC 2012 - 2012 IEEE International Conference on Communications*, pp. 27–31, Ottawa, ON, Canada, June 2012.
- [8] K. Saleem, N. Fisal, and J. Al-Muhtadi, "Empirical studies of bio-inspired self-organized secure autonomous routing protocol," *IEEE Sensors Journal*, vol. 14, no. 7, pp. 2232–2239, 2014.
- [9] W.-T. Chen and N.-F. Huang, "The strongly connecting problem on multihop packet radio networks," *IEEE Transactions on Communications*, vol. 37, no. 3, pp. 293–295, 1989.
- [10] M. X. Cheng, M. Cardei, J. Sun et al., "Topology control of ad hoc wireless networks for energy efficiency," *IEEE Transactions on Computers*, vol. 53, no. 12, pp. 1629–1635, 2004.
- [11] K. P. Eswaran and R. E. Tarjan, "Augmentation problems," *SIAM Journal on Computing*, vol. 5, no. 4, pp. 653–665, 1976.
- [12] L. M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc, "Power consumption in packet radio networks," *Theoretical Computer Science*, vol. 243, no. 1-2, pp. 289–305, 2000.
- [13] Z. Yang and Y. Liu, "Understanding node localizability of wireless ad hoc and sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 8, pp. 1249–1260, 2012.
- [14] T. Chen, Z. Yang, Y. Liu, D. Guo, and X. Luo, "Localization-oriented network adjustment in wireless ad-hoc and sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 146–155, 2014.
- [15] D. Dong, X. Liao, K. Liu, Y. Liu, and W. Xu, "Distributed coverage in wireless ad hoc and sensor networks by topological graph approaches," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 61, no. 10, pp. 1417–1428, 2012.

- [16] B. Panigrahi, S. De, B. S. Panda, and J. L. Luk, "Network lifetime maximising distributed forwarding strategies in ad hoc wireless sensor networks," *IET Communications*, vol. 6, no. 14, pp. 2138–2148, 2012.
- [17] P. Di Lorenzo and S. Barbarossa, "A bio-inspired swarming algorithm for decentralized access in cognitive radio," *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 6160–6174, 2011.
- [18] P. Di Lorenzo, S. Barbarossa, and A. H. Sayed, "Bio-inspired decentralized radio access based on swarming mechanisms over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 61, no. 12, pp. 3183–3197, 2013.
- [19] G. Kumar and M. K. Rai, "An energy efficient and optimized load balanced localization method using CDS with one-hop neighbourhood and genetic algorithm in WSNs," *Journal of Network and Computer Applications*, vol. 78, pp. 73–82, 2017.
- [20] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [21] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.
- [22] V. Feoktistov, *Differential Evolution: In Search of Solutions*, vol. 5 of *Springer Optimization and Its Applications*, New York, NY, USA, Springer, 2006.
- [23] F. Lezama, G. Castañón, and A. M. Sarmiento, "Differential evolution optimization applied to the wavelength converters placement problem in all optical networks," *Computer Networks*, vol. 56, no. 9, pp. 2262–2275, 2012.
- [24] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution—An updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.
- [25] A. W. Mohamed and A. S. Almazayad, "Differential evolution with novel mutation and adaptive crossover strategies for solving large scale global optimization problems," *Applied Computational Intelligence and Soft Computing*, vol. 2017, Article ID 7974218, 18 pages, 2017.
- [26] L. Jin, J. Jia, and D. Sun, "Node distribution optimization in mobile sensor network based on multi-objective differential evolution algorithm," in *Proceedings of the 4th International Conference on Genetic and Evolutionary Computing*, pp. 51–54, Shenzhen, China, 2010.
- [27] W. Gong and Z. Cai, "An empirical study on differential evolution for optimal power allocation in WSNs," in *Proceedings of the 2012 8th International Conference on Natural Computation (ICNC)*, pp. 635–639, Chongqing, Sichuan, China, May 2012.
- [28] Y. Xu, J. Fang, W. Zhu, and W. Cui, "Differential evolution for lifetime maximization of heterogeneous wireless sensor networks," *Mathematical Problems in Engineering*, vol. 2013, Article ID 172783, 12 pages, 2013.
- [29] S. Gundry, J. Zou, J. Kusyk, C. S. Sahin, and M. U. Uyar, "Markov chain model for differential evolution based topology control in MANETs," in *Proceedings of the 2012 35th IEEE Sarnoff Symposium*, pp. 1–5, Newark, NJ, USA, May 2012.
- [30] S. Gundry, J. Kusyk, J. Zou, C. S. Sahin, and M. U. Uyar, "Performance evaluation of differential evolution based topology control method for autonomous MANET nodes," in *Proceedings of the 2012 IEEE Symposium on Computers and Communications (ISCC)*, pp. 000228–000233, Cappadocia, Turkey, July 2012.
- [31] J.-H. Seok, J.-Y. Lee, C. Oh, J.-J. Lee, and H. J. Lee, "RFID sensor deployment using differential evolution for indoor mobile robot localization," in *Proceedings of the 23rd IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 3719–3724, Taipei, Taiwan, October 2010.
- [32] A. Céspedes-Mota, G. Castañón, A. F. Martínez-Herrera, and L. E. Cárdenas-Barrón, "Optimization of the distribution and localization of wireless sensor networks based on differential evolution approach," *Mathematical Problems in Engineering*, Article ID 7918581, 12 pages, 2016.
- [33] Y. Xu, X. Wang, and H. Zhang, "Improved differential evolution to solve the two-objective coverage problem of wireless sensor networks," in *Proceedings of the 28th Chinese Control and Decision Conference, CCDC 2016*, pp. 2379–2384, May 2016.
- [34] Y. Xu, T. Duan, W. Chen, and J. Chen, "Based on differential evolution to research the control problem of area-coverage in WSNs," in *Proceedings of the 29th Chinese Control and Decision Conference, CCDC 2017*, pp. 7298–7303, May 2017.
- [35] D. Qiao and G. Pang, "A modified differential evolution with Heuristics Algorithm for non-convex optimization on sensor network localization," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1676–1679, 2015.
- [36] B. Nancharaiiah and B. C. Mohan, "Modified ant colony optimization to enhance manet routing in adhoc on demand distance vector," in *Proceedings of the 2014 2nd International Conference on Business and Information Management, ICBIM 2014*, pp. 81–85, January 2014.
- [37] F. Hajjej, R. Ejbali, and M. Zaied, "Quality of services based routing using evolutionary algorithms for wireless sensor network," in *Proceedings of the 15th International Conference on Intelligent Systems Design and Applications, ISDA 2015*, pp. 237–242, December 2015.
- [38] X. Yulong, W. Xiaopeng, and Z. Han, "Comparative study on the optimal path problem of wireless sensor networks," in *Proceedings of the 13th IEEE International Conference on Mechatronics and Automation, IEEE ICMA 2016*, pp. 2234–2239, August 2016.
- [39] D. S. Deif and Y. Gadallah, "An ant colony optimization approach for the deployment of reliable wireless sensor networks," *IEEE Access*, vol. 5, pp. 10744–10756, 2017.
- [40] H. Feng and J. Qi, "Radio frequency identification networks planning using a new hybrid evolutionary algorithm," in *Proceedings of the 5th International Conference Advanced Communication Technology (ICACT)*, pp. 179–188, 2013.
- [41] P. Li, Y. Wang, J. Hu, and J. Zhou, "Sensors distribution optimization for impact localization using NSGA-II," *Sensor Review*, vol. 35, no. 4, pp. 409–418, 2015.
- [42] J. M. L. Gutiérrez, J. A. G. Pulido, and M. A. V. Rodríguez, "A new realistic approach for the relay node placement problem in wireless sensor networks by means of evolutionary computation," *Ad-Hoc and Sensor Wireless Networks*, vol. 26, no. 1–4, pp. 193–209, 2015.
- [43] S. M. Jamei, K. Faez, and M. Dehghan, "AMOF: adaptive multi-objective optimization framework for coverage and topology control in heterogeneous wireless sensor networks," *Telecommunication Systems*, vol. 61, no. 3, pp. 515–530, 2016.
- [44] S. Özdemir, B. A. Attea, and Ö. A. Khalil, "Multi-objective evolutionary algorithm based on decomposition for energy efficient coverage in wireless sensor networks," *Wireless Personal Communications*, vol. 71, no. 1, pp. 195–215, 2013.
- [45] M. Le Berre, M. Rebai, F. Hnaïen, and H. Snoussi, "A specific heuristic dedicated to a coverage/tracking bi-objective problem for wireless sensor deployment," *Wireless Personal Communications*, vol. 84, no. 3, pp. 2187–2213, 2015.

- [46] B. A. A. Attea, E. A. Khalil, and A. Cosar, "Multi-objective evolutionary routing protocol for efficient coverage in mobile sensor networks," *Soft Computing*, vol. 19, no. 10, pp. 2983–2995, 2015.
- [47] R. Murugeswari and S. Radhakrishnan, "Discrete multi-objective differential evolution algorithm for routing in wireless mesh network," *Soft Computing*, vol. 20, no. 9, pp. 3687–3698, 2016.
- [48] S. Singh and R. M. Sharma, "Optimization techniques in wireless sensor networks," in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies, ICTCS '16*, pp. 1–7, New York, NY, USA, March 2016.
- [49] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [50] K. Miettinen, *Nonlinear Multiobjective Optimization*, Kluwer Academic Publishers, Norwell, Mass, USA, 1999.
- [51] R. S. Cahn, *Wide Area Network Design: Concepts and Tools for Optimization*, The Morgan Kaufmann Series in Networking, Morgan Kaufmann Publishers, 1998.
- [52] A. S. Nepomniaschaya, "An associative version of the Prim-Dijkstra algorithm and its application to some graph problems," in *Perspectives of System Informatics*, vol. 1181 of *Lecture Notes in Computer Science*, pp. 203–213, Springer Berlin Heidelberg, Berlin, Germany, 1996.
- [53] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society For Industrial & Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.
- [54] U. K. Chakraborty, S. K. Das, and T. E. Abbott, "Energy-efficient routing in hierarchical wireless sensor networks using differential-evolution-based memetic algorithm," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '12)*, pp. 1–8, Brisbane, Australia, June 2012.
- [55] J. Jia, J. Chen, G. Chang, Y. Wen, and J. Song, "Multi-objective optimization for coverage control in wireless sensor network with adjustable sensing radius," *Computers & Mathematics with Applications. An International Journal*, vol. 57, no. 11-12, pp. 1767–1775, 2009.
- [56] T. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001.
- [57] A. L. Garcia, *Probability, Statistics, and Random Processes for Electrical Engineering*, Pearson/Prentice Hall, 2008.
- [58] S. Epp, *Discrete Mathematics with Applications*, Brooks/Cole-Thompson Learning, Belmont, CA, USA, 3rd edition, 2004.
- [59] L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer, "A cone-based distributed topology-control algorithm for wireless multi-hop networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 1, pp. 147–159, 2005.
- [60] E. W. Weisstein, "Circle-Circle Intersection," <http://mathworld.wolfram.com/Circle-CircleIntersection.html>, 2017.
- [61] P. M. Fewell, "Area of common overlap of three circles," Technical report, DSTO Defence Science and Technology Organization, Edinburg South Australia, 2006.
- [62] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 482–500, 2012.
- [63] R. C. Prim, "Shortest connection networks and some generalizations," *Bell Labs Technical Journal*, vol. 36, no. 6, pp. 1389–1401, 1957.
- [64] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.
- [65] F. Lezama, G. Castañón, and A. M. Sarmiento, "Routing and wavelength assignment in all optical networks using differential evolution optimization," *Photonic Network Communications*, vol. 26, no. 2-3, pp. 103–119, 2013.
- [66] F. Lezama, G. Castañón, A. M. Sarmiento, and I. B. Martins, "Differential evolution optimization applied to the routing and spectrum allocation problem in flexgrid optical networks," *Photonic Network Communications*, vol. 31, no. 1, pp. 129–146, 2016.
- [67] S. M. K. Heris, "Generic implementations of MOEA/D and NSGA-II," <http://yarpiz.com>, 2018.
- [68] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.
- [69] J. Q. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.
- [70] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

## Research Article

# A Hybrid Epigraph Directions Method for Nonsmooth and Nonconvex Constrained Optimization via Generalized Augmented Lagrangian Duality and a Genetic Algorithm

Wilhelm P. Freire,<sup>1</sup> Afonso C. C. Lemonge ,<sup>2</sup>  
Tales L. Fonseca ,<sup>3</sup> and Hernando J. R. Franco <sup>3</sup>

<sup>1</sup>Department of Mathematics, Institute of Exact Sciences, Federal University of Juiz de Fora, Brazil

<sup>2</sup>Department of Applied and Computational Mechanics, Faculty of Engineering, Federal University of Juiz de Fora, Brazil

<sup>3</sup>Postgraduate Program of Computational Modeling, Federal University of Juiz de Fora, Brazil

Correspondence should be addressed to Hernando J. R. Franco; [hernando.franco@ifsudestemg.edu.br](mailto:hernando.franco@ifsudestemg.edu.br)

Received 16 January 2018; Accepted 15 April 2018; Published 22 May 2018

Academic Editor: Guillermo Cabrera-Guerrero

Copyright © 2018 Wilhelm P. Freire et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Interior Epigraph Directions (IED) method for solving constrained nonsmooth and nonconvex optimization problem via Generalized Augmented Lagrangian Duality considers the dual problem induced by a Generalized Augmented Lagrangian Duality scheme and obtains the primal solution by generating a sequence of iterates in the interior of the epigraph of the dual function. In this approach, the value of the dual function at some point in the dual space is given by minimizing the Lagrangian. The first version of the IED method uses the Matlab routine `fminsearch` for this minimization. The second version uses NFDNA, a tailored algorithm for unconstrained, nonsmooth and nonconvex problems. However, the results obtained with `fminsearch` and NFDNA were not satisfactory. The current version of the IED method, presented in this work, employs a Genetic Algorithm, which is free of any strategy to handle the constraints, a difficult task when a metaheuristic, such as GA, is applied alone to solve constrained optimization problems. Two sets of constrained optimization problems from mathematics and mechanical engineering were solved and compared with literature. It is shown that the proposed hybrid algorithm is able to solve problems where `fminsearch` and NFDNA fail.

## 1. Introduction

We present a new version of the Interior Epigraph Directions Method published by Burachik et al. [1] for solving constrained nonsmooth and nonconvex optimization problems of the following kind:

$$\text{minimize } f(x) \text{ over all } x \in X \text{ satisfying } g(x) = 0, \quad (P)$$

where  $X \subset \mathbb{R}^n$  is compact and the functions  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  and  $g: \mathbb{R}^n \rightarrow \mathbb{R}^m$  are continuous. Inequality constraints can be incorporated into (P) by using the operator  $a^+ = \max\{a, 0\}$ ,  $a \in \mathbb{R}$ .

The IED method uses a Lagrangian duality technique, where each constraint function is appended to the objective function via a multiplier, or dual variable, to form the classical

Lagrangian. In a duality scheme, instead of problem (P), the problem solved is the one where the dual function is maximized. In this approach, the value of the dual function at a given point in the dual space is computed by minimizing the Lagrangian which is an unconstrained problem, and therefore unconstrained methods can be used.

It turns out that the optimal value of the dual problem associated with the classical Lagrangian is in general not the same as that of the primal problem (P), if (P) is nonconvex. The difference between the two optimal values is called duality gap.

Zero duality gap and saddle-point properties without convexity assumptions are known to hold for the following family of Lagrangian function [2, Remark 2.1].

$$L(x, (u, c)) = f(x) + c\sigma(g(x)) + \langle Au, g(x) \rangle, \quad (1)$$

where  $x \in \mathbb{R}^n$ ,  $u \in \mathbb{R}^m$ ,  $c \in \mathbb{R}_+ = [0, \infty)$ ,  $A \in \mathbb{R}^{m \times m}$  is a real symmetric matrix, and  $\sigma : \mathbb{R}^m \rightarrow \mathbb{R}$  is a continuous function such that  $\sigma(x) > 0$ ,  $\forall x \in \mathbb{R}^m \setminus \{0\}$  and  $\sigma(0) = 0$ .

The dual problem generated by such Lagrangians as in (1) is a (nondifferentiable) convex problem, which is usually solved by nonsmooth convex optimization techniques, such as subgradient methods and their extensions [2–8].

The IED method uses the family of Lagrangians (1). These Lagrangians induce dual variables  $(u, c) \in \mathbb{R}^m \times \mathbb{R}_+$ , where  $u$  is the classical multiplier associated with the linear term in the Lagrangian and  $c \geq 0$  is the penalty multiplier associated with the augmenting term.

A duality approach based on the Lagrangian  $L$  for solving (P) can be described as follows. Given a current dual iterate  $(u_k, c_k)$ , a primal iterate is computed through the rule

$$x_k \in \arg \min_{x \in X} L(x, (u_k, c_k)), \quad (2)$$

where  $L$  is as in (1).

The iterate  $x_k$  obtained in (2) can be used for computing the search direction which leads to the next dual iterate  $(u_{k+1}, c_{k+1})$ . Indeed,  $x_k$  is used to obtain a deflection of the subgradient direction which in turn is utilized to improve the dual values. This is the idea behind the methods studied in [2–8]. These methods are referred to as deflected subgradient (DSG) methods.

The IED method [1] combines the Nonsmooth Feasible Directions Algorithm (NFDA) for solving unconstrained nonsmooth convex problems, firstly presented in [9] and further studied in [10], with a certain DSG method.

Roughly speaking, the IED method works as follows: starting at a point in the interior of the epigraph of the dual function, a deflected subgradient direction is used to define a linear approximation to the epigraph. An auxiliary point is obtained by solving the optimality conditions of a resulting linear problem. If the auxiliary point belongs to the interior of the epigraph, the IED step is declared serious and the auxiliary point becomes the next iterate, from which the process is repeated. If the auxiliary point does not belong to the interior of the epigraph, the IED step is declared null and a suitable version of the DSG step is applied from the original point. Thanks to the properties of the DSG method, this step stays in the interior of the epigraph. The new point takes the place of the original point, from which the process is repeated, until a serious step is performed. If an infinite number of null steps occurs, the method converges to a primal solution, as a consequence of the convergence properties of the DSG.

An important aspect of the method is the minimization of the Lagrangian function. Although the solver used for this task does not affect the convergence of the IED method, as proved in [1], solving this problem effectively is fundamental for the success of the methods based on Lagrangian duality.

The first version of the IED method [1] employs the Matlab routine `fminsearch` for the Lagrangian minimization. The second version of the IED method [15] uses an algorithm called Nonsmooth Feasible Directions Nonconvex Algorithm (NFDNA) [16, 17] for solving nonsmooth and nonconvex unconstrained optimization problems which correspond to our problem (2). Although the previous versions of IED

were able to solve many problems, they failed some test problems and did not find the desired solutions as one can observe through some numerical experiments presented in this paper and found in [1, 15]. We do not know why `fminsearch` and `NFDNA` fail. For `fminsearch` is a routine of Matlab, we think that it does not make sense trying to fix or change it. The `NFDNA` performance needs deeper investigation. In this work, we have used a Genetic Algorithm (GA) for minimizing the Lagrangian function. With this modification, the method gained robustness and was able to solve all the test problems analyzed here. The hybridization of the IED method and the Genetic Algorithm is the main purpose of this work and can be considered innovative. Also, it is important to notice that the GA is free of any strategy to handle the constraints, a difficult task when a metaheuristic, such as the GA, is applied alone to solve constrained optimization problems [18].

We have solved twenty problems and compared the performance of the current version of the IED method with the performance of the first version. We also solved four real-world mechanical engineering optimization problems with the current version of the IED.

This paper is organized in 8 sections. In the next section, we state the primal and the dual problems and recall how to find a subgradient of the dual function. In Section 3, we show how the search direction used by IED is obtained. In Section 4, we present NFDA, an algorithm for convex optimization problems that inspired the IED method. The IED method is then presented in Section 5. In Section 6, we present a brief description of the main characteristics of the Genetic Algorithm and its basic features adopted in this paper. Numerical results and comparisons are given in Section 7. Finally, conclusions and suggestions for future work are given in Section 8.

## 2. The Problem in Study

The problems we are interested in are of the same type of the primal problem (P).

The augmented Lagrangian function  $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_+ \rightarrow \mathbb{R}$  associated with (P) is defined as in (1). The corresponding concave dual function  $H : \mathbb{R}^m \times \mathbb{R}_+ \rightarrow \mathbb{R}$  is defined as

$$H(u, c) = \min_{x \in X} [f(x) + c\sigma(g(x)) + \langle Au, g(x) \rangle]. \quad (3)$$

The dual problem is given by

$$\max_{(u, c) \in \mathbb{R}^m \times \mathbb{R}_+} H(u, c) \quad (D)$$

and, since the dual function  $H$  is concave, (D) is equivalent to the convex problem

$$\min_{(u, c) \in \mathbb{R}^m \times \mathbb{R}_+} -H(u, c). \quad (D^*)$$

For convenience, we introduce the set

$$X(u, c) = \arg \min_{x \in X} [f(x) + c\sigma(g(x)) + \langle Au, g(x) \rangle] \quad (4)$$

whose elements are obtained by the GA used in this work.

Figure 8 shows a flowchart that summarizes how such minimization is done.

For simplicity ( $D^*$ ) will be our dual problem and we will refer to  $q := -H$  as the dual function.

We recall now an important result that shows how, after solving problem (4), we can easily obtain a subgradient  $s \in \partial q(u, c)$ .

**Theorem 1** (see [2], Lemma 2.1(a)). *Given the dual function  $q$  and a pair  $v = (u, c) \in \mathbb{R}^m \times \mathbb{R}_+$ , take  $x \in X(v)$ . Then  $s = (-Ag(x), -\sigma(g(x))) \in \partial q(v)$ .*

### 3. The IED Search Direction

In this section we describe how the IED search direction is obtained.

The methodology employed here was firstly used by an algorithm called Feasible Directions Interior Point Algorithm (FDIPA) [19] for smooth nonlinear problems.

Let us consider the inequality constrained optimization problem

$$\begin{aligned} & \text{minimize} && f(x), \\ & \text{subject to} && g(x) \leq 0, \end{aligned} \quad (P_3)$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  and  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$  are continuously differentiable.

Let  $x^*$  be a regular point of problem ( $P_3$ ). The Karush-Kuhn-Tucker (KKT) first order necessary optimality conditions are expressed as follows: If  $x^*$  is a local minimum of ( $P_3$ ) then there exists  $\lambda^* \in \mathbb{R}^m$  such that

$$\nabla f(x^*) + \nabla g(x^*) \lambda^* = 0 \quad (5)$$

$$G(x^*) \lambda^* = 0 \quad (6)$$

$$\lambda^* \geq 0 \quad (7)$$

$$g(x^*) \leq 0, \quad (8)$$

where  $G(x)$  is a diagonal matrix with  $G_{ii}(x) \equiv g_i(x)$ .

The point  $x$  such that  $g(x) \leq 0$  is called a ‘‘Primal Feasible Point’’ and  $\lambda \geq 0$  a ‘‘Dual Feasible Point’’. Given an initial feasible pair  $(x^0, \lambda^0)$ , KKT points are found by solving iteratively the nonlinear system of equations (5) and (6) in  $(x, \lambda)$ , in such a way that all the iterates are primal and dual feasible.

A Newton-like iteration to solve the nonlinear system of equations (5) and (6) can be stated as

$$\begin{aligned} & \begin{pmatrix} B^k & \nabla g(x^k) \\ \Lambda^k \nabla g(x^k)^T & G(x^k) \end{pmatrix} \begin{pmatrix} x - x^k \\ \lambda - \lambda^k \end{pmatrix} \\ & = - \begin{pmatrix} \nabla f(x^k) + \nabla g(x^k) \lambda^k \\ G(x^k) \lambda^k \end{pmatrix}, \end{aligned} \quad (9)$$

where  $(x^k, \lambda^k)$  is the current point of the iteration  $k$ ,  $\Lambda$  is a diagonal matrix with  $\Lambda_{ii} \equiv \lambda_i$ , and  $B^k \equiv \nabla^2 f(x^k) +$

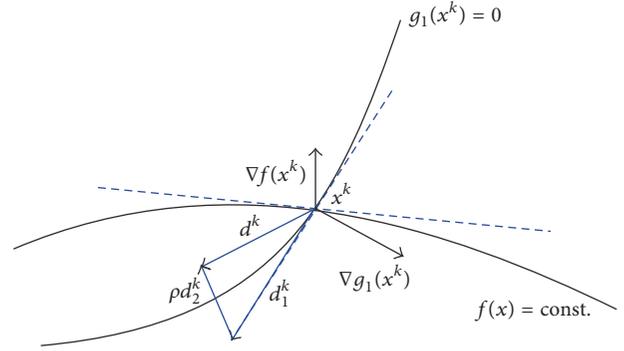


FIGURE 1: The search direction of FDIPA.

$\sum_{i=1}^m \lambda_i^k \nabla^2 g_i(x^k)$  is the Hessian of the Lagrangian function  $L(x, \lambda) = f(x) + \lambda^T g(x)$  or some quasi-Newton approximation. However,  $B^k$  must be symmetric and positive definite in order to ensure convergence.

Calling  $d = x - x^k$ , the following system can be written from (9):

$$B^k d + \nabla g(x^k) \lambda = -\nabla f(x^k) \quad (10)$$

$$\Lambda^k \nabla g(x^k)^T d + G(x^k) \lambda = 0. \quad (11)$$

The solution  $(d_1^k, \lambda_1^k)$  of the system (10)-(11) provides a descent direction  $d_1^k$  as proved in [19]. However,  $d_1^k$  can not be employed as a search direction, since it is not necessarily a feasible direction because, in the case when  $g_i(x^k) = 0$ , it follows from (11) that  $\nabla g_i(x^k)^T d_1^k = 0$ .

To avoid this drawback, a new system with unknowns  $d$  and  $\bar{\lambda}$  is defined by adding a negative matrix  $-\rho^k \Lambda^k$ , with  $\rho^k > 0$ , to the right side of (11),

$$B^k d + \nabla g(x^k) \bar{\lambda} = -\nabla f(x^k) \quad (12)$$

$$\Lambda^k \nabla g(x^k)^T d + G(x^k) \bar{\lambda} = -\rho^k \Lambda^k. \quad (13)$$

Now, (13) is equivalent to  $\lambda_i^k \nabla g_i(x^k)^T d^k + g_i(x^k) \bar{\lambda}^k = -\rho^k \lambda_i^k$ ,  $i = 1, 2, \dots, m$ . Consequently, if  $g_i(x^k) = 0$ , the latest equation is reduced to  $\nabla g_i(x^k)^T d^k = -\rho^k < 0$ , which means that  $d^k$  is a feasible direction.

The addition of a negative term  $-\rho^k \lambda^k$  produces a proportional deflexion of  $d^k$  into the feasible region, as one can see in Figure 1. The problem now is that  $d^k$  might not be a descent direction for the function  $f$ . However, we can keep such property if  $\rho^k$  is properly chosen. Notice that the direction  $d^k$  can be obtained by solving two systems with the same matrix

$$B^k d_1 + \nabla g(x^k) \lambda_1 = -\nabla f(x^k) \quad (14)$$

$$\Lambda^k \nabla g(x^k)^T d_1 + G(x^k) \lambda_1 = 0, \quad (15)$$

$$B^k d_2 + \nabla g(x^k) \lambda_2 = 0 \quad (16)$$

$$\Lambda^k \nabla g(x^k)^T d_2 + G(x^k) \lambda_2 = -\Lambda^k \quad (17)$$

and setting  $d^k = d_1^k + \rho^k d_2^k$ , which gives  $(d^k)^T \nabla f(x^k) = (d_1^k)^T \nabla f(x^k) + \rho^k (d_2^k)^T \nabla f(x^k)$ . In the case where  $(d_2^k)^T \nabla f(x^k) \leq 0$  we have  $(d^k)^T \nabla f(x^k) < 0, \forall \rho^k > 0$ .

Thus,  $\rho^k$  must be well chosen if  $(d_2^k)^T \nabla f(x^k) > 0$ . In this case, imposing  $(d^k)^T \nabla f(x^k) \leq \xi (d_1^k)^T \nabla f(x^k)$ , where  $\xi \in (0, 1)$ , implies  $(d_1^k)^T \nabla f(x^k) + \rho^k (d_2^k)^T \nabla f(x^k) \leq \xi (d_1^k)^T \nabla f(x^k)$  which leads to

$$\rho^k \leq \frac{(\xi - 1) (d_1^k)^T \nabla f(x^k)}{(d_2^k)^T \nabla f(x^k)}. \quad (18)$$

Thus, with  $\rho^k$  chosen as described above, we have that  $d^k = d_1^k + \rho^k d_2^k$  is a feasible descent direction.

The ideas just described are sufficient for us to understand how the search direction used by the IED method is computed.

For more on FDIPA, its features, and convergence, see [19].

#### 4. NFDA for Convex Problems

The Nonsmooth Feasible Directions Algorithm (NFDA), firstly presented by Freire [9], has been devised for solving unconstrained nonsmooth convex problems of the following kind:

$$\begin{aligned} \min \quad & F(x) \\ \text{subject to} \quad & x \in \mathbb{R}^n, \end{aligned} \quad (P_{4.1})$$

where  $F : \mathbb{R}^n \rightarrow \mathbb{R}$  is a convex, not necessarily differentiable function.

Problem  $(P_{4.1})$  can be replaced with the equivalent constrained problem

$$\begin{aligned} \min \quad & f(x, z) = z \\ \text{subject to} \quad & F(x) \leq z \\ & (x, z) \in \mathbb{R}^n \times \mathbb{R}. \end{aligned} \quad (P_{4.2})$$

It is assumed that one arbitrary subgradient  $s \in \partial F(x)$  can be computed at any point  $x \in \mathbb{R}^n$ .

NFDA starts at a point  $(x^0, z^0)$  in the interior of the epigraph of the function  $F$ . At a point  $(x^k, F(x^k))$ , the method determines a supporting hyperplane to the epigraph of the function  $F$

$$h^k(x) = F(x^k) + \langle s^k, (x - x^k) \rangle, \quad (19)$$

where  $s^k \in \partial F(x^k)$  and defines an auxiliary constrained linear problem employing all the supporting hyperplanes computed so far, as follows:

$$\begin{aligned} \min \quad & f(x, z) = z \\ \text{subject to} \quad & g^k(x, z) \leq 0 \\ & (x, z) \in \mathbb{R}^n \times \mathbb{R}, \end{aligned} \quad (P_{4.3})$$

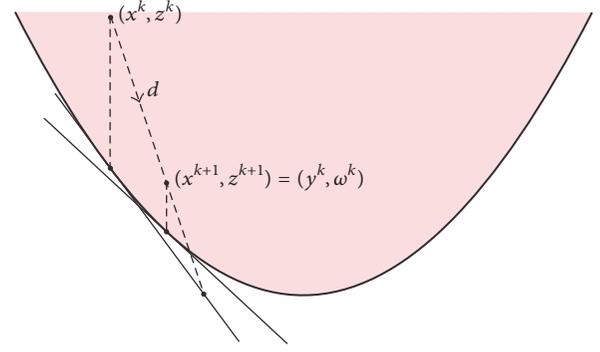


FIGURE 2: Serious step.

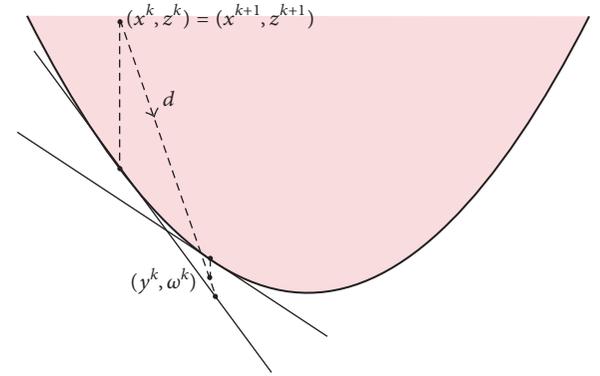


FIGURE 3: Null step.

where  $g^k = (g_1^k, \dots, g_m^k) : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^m$  is a vector function with  $g_i^k : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$  given by  $g_i^k(x, z) = h_i^k(x) - z$ .

Problem  $(P_{4.3})$  is not solved. Indeed, it might not have a solution. NFDA uses its structure, which is similar to the structure of the problem  $(P_3)$ , to obtain a search direction by solving two systems and choosing the parameter  $\rho$  in the same way FDIPA does.

At the iterate  $(x^k, z^k)$  and having the search direction  $d^k$  at hand, a step  $t^k$  is obtained by the rule

$$t^k = \min \{t_{\max}, T\}, \quad (20)$$

where  $t_{\max} = \max\{t \mid g_i^k((x^k, z^k) + td^k) \leq 0\}$  and  $T > 0$  is a predefined parameter, since  $t_{\max}$  is not always finite.

A new point  $(y^k, \omega^k) = (x^k, z^k) + \mu t^k d^k$  with  $\mu \in (0, 1)$  is computed. If  $F(y^k) < \omega^k$  then  $(x^{k+1}, z^{k+1}) = (y^k, \omega^k)$  and the step is called *serious*, as shown in Figure 2. Otherwise, if  $F(y^k) \geq \omega^k$  then  $(x^{k+1}, z^{k+1}) = (x^k, z^k)$  and the step is called *null*; see Figure 3.

In any case, a new supporting hyperplane at the point  $(y^k, F(y^k))$  is added to form a new auxiliary problem like  $(P_{4.3})$ .

These are the main ideas from NFDA; we need to understand the IED algorithm. For more on NFDA, see [9, 10].

## 5. The IED Method

In this section we describe the Interior Epigraph Directions (IED) method for solving constrained, nonsmooth, and nonconvex optimization problems.

As we have seen in Section 2, the dual problem (D) is equivalent to the problem

$$\begin{aligned} \min \quad & q(u, c) \\ & (u, c) \in \mathbb{R}^m \times \mathbb{R}_+ \end{aligned} \quad (D^{**})$$

which, in its turn, is equivalent to

$$\begin{aligned} \min \quad & \psi(u, c, z) = z \\ \text{subject to} \quad & q(u, c) \leq z \\ & (u, c, z) \in \mathbb{R}^m \times \mathbb{R}_+ \times \mathbb{R}. \end{aligned} \quad (ED^*)$$

One can notice that problems  $(P_{4.2})$  and  $(ED^*)$  have the same structure. Therefore, we can apply to  $(ED^*)$  the ideas used by NFDA.

However, NFDA can only minimize nonsmooth convex functions which have bounded level sets. The convex function  $q$ , in general, does not enjoy this property. Therefore, the IED method uses directions which are suited to the function  $q$  we have. Indeed, the IED method takes advantage of both NFDA search direction and the special features of the dual function  $q$ .

At the step  $k$  of the IED method, we have a point  $(v^k, z^k) = ((u^k, c^k), z^k) \in \text{int}(\text{epi}(q))$  and a primal iterate  $x^k \in X(v^k)$  such that  $g(x^k) \neq 0$ .

Now, having  $x^k$ , according to Theorem 1, we can easily find a subgradient

$$s^k = (-Ag(x^k), -\sigma(g(x^k))) \in \partial q(v^k). \quad (21)$$

This subgradient defines a supporting hyperplane

$$h^k(v) = q(v^k) + \langle s^k, v - v^k \rangle \quad (22)$$

to the  $\text{epi}(q)$  at the point  $(v^k, q(v^k))$ . By employing this supporting hyperplane, the following auxiliary problem is defined

$$\begin{aligned} \min \quad & \psi(u, c, z) = z \\ \text{subject to} \quad & g^k(u, c, z) \leq 0 \\ & (u, c, z) \in \mathbb{R}^m \times \mathbb{R}_+ \times \mathbb{R}, \end{aligned} \quad (AP_k)$$

where  $g^k : \mathbb{R}^m \times \mathbb{R}_+ \times \mathbb{R} \rightarrow \mathbb{R}$  is given by  $g^k(v, z) = h^k(v) - z$ .

Solving two systems, similar to (14)-(15) and (16)-(17), obtained from  $(AP_k)$ , the directions  $d_1^k$  and  $d_2^k$  are computed. It is important to highlight that there is no duplication of the computational cost since the coefficient matrices of both systems are the same.

The search direction is thus defined by  $d^k = d_1^k + \rho^k d_2^k$  where the parameter  $\rho^k > 0$  must be chosen so that  $d^k$  is a

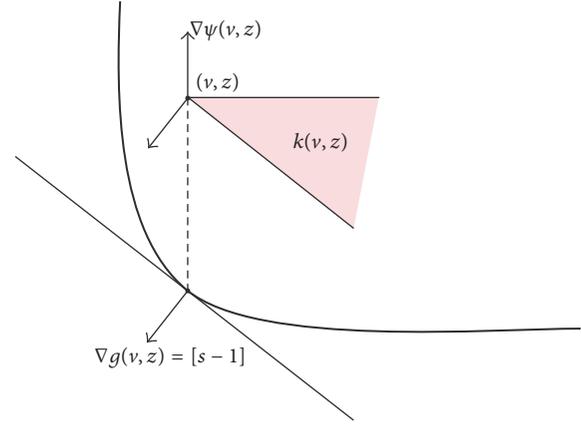


FIGURE 4: The cone  $K(v, z)$ .

descent direction for  $\psi$ , i.e.,  $(d^k)^T \nabla \psi(v^k, z^k) < 0$  and, at the same time, a local descent direction for the dual function  $q$ . This requires  $d^k$  to be a descent direction for  $g$  as well, i.e.,  $(d^k)^T \nabla g(v^k, z^k) < 0$ .

In other words, IED finds a direction  $d$  which, at each iteration, belongs to the interior of the cone

$$\begin{aligned} K(v^k, z^k) = \{d \in \mathbb{R}^{m+1} \times \mathbb{R} \mid & d^T \nabla g(v^k, z^k) \\ & < 0, d^T \nabla \psi(v^k, z^k) < 0\}, \end{aligned} \quad (23)$$

according to Figure 4.

Burachik et al. [1] proved that if  $\rho$  satisfies

$$-\frac{d_1^T \nabla g(v, z)}{d_2^T \nabla g(v, z)} < \rho < (1 - \xi) \frac{d_1^T \nabla \psi(v, z)}{d_2^T \nabla \psi(v, z)} \quad (24)$$

then  $d \in K(v, z)$ .

From the current iterate  $(v^k, z^k) \in \text{int}(\text{epi}(q))$ , using a suitable stepsize  $t^k$ , an auxiliary point

$$(\bar{u}^k, \bar{c}^k, \bar{z}^k) = (\bar{v}^k, \bar{z}^k) = (v^k, z^k) + t^k d^k \quad (25)$$

is computed.

If  $g(\bar{x}^k) = 0$  where  $\bar{x}^k \in X(\bar{v}^k)$  then IED stops.

Otherwise, i.e., if  $g(\bar{x}^k) \neq 0$  and  $q(\bar{v}^k) < \bar{z}^k$  then  $(v^{k+1}, z^{k+1}) = (\bar{v}^k, \bar{z}^k)$ .

If  $q(\bar{v}^k) \geq \bar{z}^k$  then a DSG step is performed in order to obtain the next iterate  $(v^{k+1}, z^{k+1}) \in \text{int}(\text{epi}(q))$ .

A new supporting hyperplane at the point  $(v^{k+1}, q(v^{k+1}))$  is computed, a new auxiliary problem is defined, and the algorithm goes this way until a primal iterate  $x^k$  satisfying  $g(x^k) = 0$  is found.

We now describe formally the algorithm.

### 5.1. The IED Algorithm

*Step 0.* Fix a sequence  $\{\alpha_k\} \subset \mathbb{R}_+$  and a sequence  $\{B^k\}$  of positive definite matrices. Fix  $\beta \geq \eta > 0$ ,  $T_0 \in (0, 1)$ ,  $\hat{q} \leq \bar{q}$ , where  $\bar{q}$  is the optimal dual value, a symmetric matrix  $A$ ,



```

(1) Input: Parameter  $popsiz$ ,  $gensiz$ ,  $u$ ,  $c$ ;
(2) Output: The best solution found in all generations;
(3)  $gen = 1$ ;
(4)  $Pop[gen] = InitializePopulation()$ ;
(5) Evaluate  $(L(x, (u, c)) = f(x) + c\sigma(g(x)) + \langle Au, g(x) \rangle)$ ,  $Pop[gen]$ ;
(6) while  $gen \leq gensiz$  do
(7)    $selPop[gen] = Selection(Pop[gen])$ ;
(8)    $newPop[gen] = Crossover(selPop[gen])$ ;
(9)    $Mutation(newPop[gen])$ ;
(10)  Evaluate  $(L(x, (u, c)) = f(x) + c\sigma(g(x)) + \langle Au, g(x) \rangle)$ ,  $newPop[gen]$ ;
(11)   $Pop[gen + 1] = NewGeneration(newPop[gen], Pop[gen])$ ;
(12)   $gen = gen + 1$ ;
(13) end
    
```

ALGORITHM 1: The generational Genetic Algorithm to minimize the Lagrangian function.

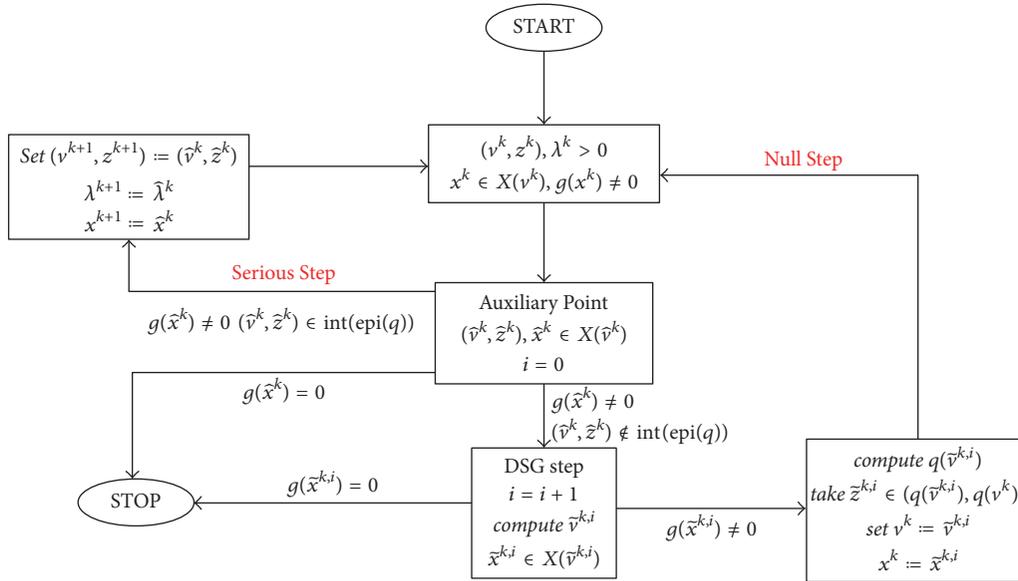


FIGURE 6: A flow chart for a step of the IED method with a Genetic Algorithms.

## 6. The Genetic Algorithm

Genetic algorithms (GAs) [20–22] were inspired by the Darwinian principles of natural selection and evolution. They are considered the most famous bioinspired metaheuristic and the basis of innumerable algorithms found in the related literature. GAs encode all the variables, using an adopted alphabet, as binary code, for example, corresponding to a candidate solution (defined as the phenotype) in a data structure (defined as the genotype). A population of genotypes evolves mimicking the evolutionary process found in the Nature. The candidate solutions are submitted to the evaluation of their quality and they are selected by a stochastic process favoring better solutions. The genetic materials of these are recombined and mutated by means of genetic operators generating a new population.

Basically, a GA presents the following five steps [21]: (i) a genetic coding of solutions to the problem; (ii) a procedure

for creating an initial population of solutions; (iii) an evaluation function that returns the quality of each individual (fitness function); (iv) genetic operators manipulating genetic material of parents during the reproduction process giving rise to new individuals; and (v) parameters to be used in the algorithm during the recombination and mutation operations.

The GAs are free-derivative algorithms and massively used to solve unconstrained and constrained optimization problems.

The link between IED and Genetic Algorithm recognized through the notation  $x^k \in X(v^k)$  can be seen in Figures 6 and 7.

The pseudocode depicted in Algorithm 1 describes the main steps of a generational Genetic Algorithm for minimizing the Lagrangian.

In this paper, a Genetic Algorithm with a real code is adopted and the Simulated Binary Crossover (SBX) [23] and

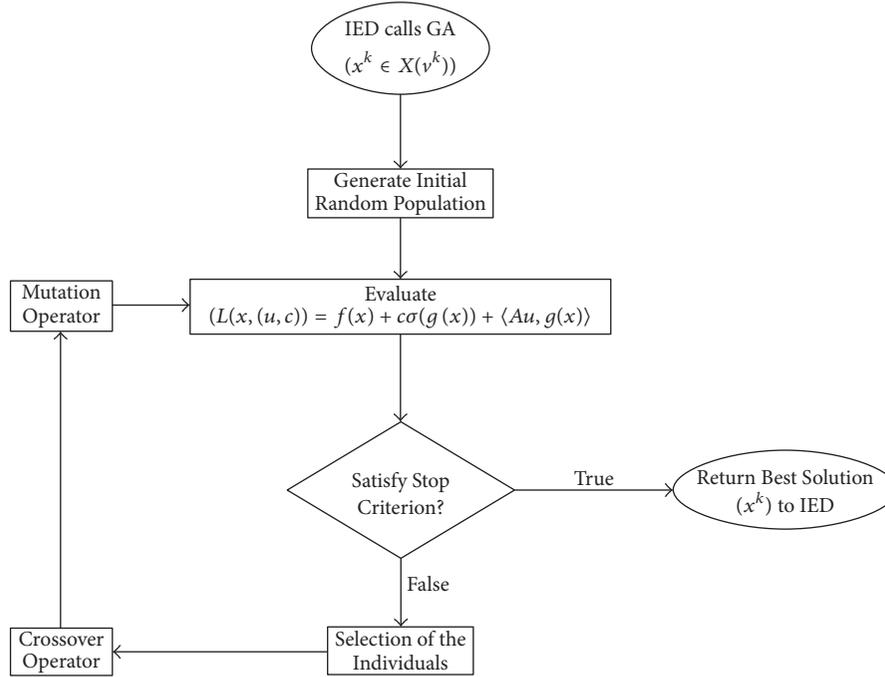


FIGURE 7: A flow chart of the Genetic Algorithm used inside the IED.

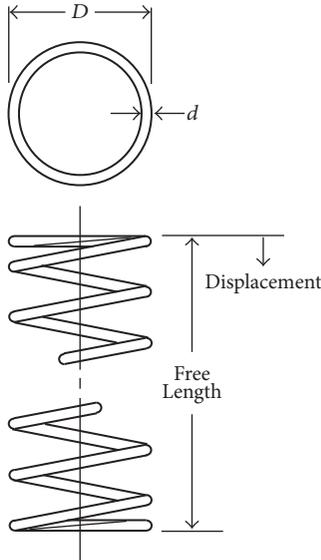


FIGURE 8: The Tension/Compression String.

the polynomial mutation [24] have been set as the genetic operators. The next subsections describe these operators. The baseline real-coded generational GA uses a rank-based selection and elitism (the best element is always copied into the next generation avoiding the loss of good solutions).

**6.1. SBX Crossover.** SBX is a crossover based on the properties presented by the one-point crossover in binary-coded. These properties are as follows:

- (i) Average: the average of the parents is equal to the average of the children after the crossover operation;
- (ii) Spread factor  $\beta$ : most of the crossover events result in a spread factor of  $\beta \approx 1$ , that is, children tend to be close to their parents.

To maintain the average property, the offspring values are calculated as

$$\begin{aligned} ch_1 &= \bar{q} - \frac{1}{2}\beta(p_2 - p_1) \\ ch_2 &= \bar{q} + \frac{1}{2}\beta(p_2 - p_1), \end{aligned} \quad (30)$$

where  $\bar{q} = (1/2)(p_1 + p_2)$ ,  $p_2 > p_1$ ,  $p_1$  and  $p_2$  are the parents and,  $ch_1$  and  $ch_2$  are the offspring. It can be observed that  $\overline{ch} = \bar{q}$ . The parameter  $\beta$  is a random variable obtained from the following probabilistic distribution function:

$$f(\beta) = \begin{cases} 0.5(\eta + 1)\beta^\eta, & \text{if } \beta \leq 1, \\ 0.5(\eta + 1)\frac{1}{\beta^{\eta+2}} & \text{otherwise,} \end{cases} \quad (31)$$

where small values for  $\eta$  generate offspring which are more similar to their parents and, on the other hand, great values generate offspring which are quite different from their parents. In this work, we used  $\eta = 2$ , a value that works well according to the original reference [23].

**6.2. Polynomial Mutation.** Polynomial mutation is an operator that uses polynomial probability distribution to perturb a solution in a parent's proximity. Given a parent solution

$p \in [a, b]$ , the polynomial mutation solution  $p'$  is created by using a random number  $r \in [0, 1]$ , for a given variable, as follows:

$$p'_i = \begin{cases} p_i + \bar{\delta}_L (p_i - x_i^{(L)}), & \text{for } r \leq 0.5, \\ p_i + \bar{\delta}_R (x_i^{(R)} - p_i) & \text{for } r > 0.5, \end{cases} \quad (32)$$

where  $\bar{\delta}_L$  and  $\bar{\delta}_R$  can be calculated as follows:

$$\begin{aligned} \bar{\delta}_L &= (2r)^{(1/1+\eta_m)} - 1, & \text{for } r \leq 0.5 \\ \bar{\delta}_R &= 1 - (2(1-r))^{(1/1+\eta_m)}, & \text{for } r > 0.5. \end{aligned} \quad (33)$$

It is important to note that no value outside the range  $[a, b]$  is created by the polynomial mutation. In this work, we used  $\eta_m = 100$ , a value that works well according to the original reference [24].

## 7. Numerical Experiments

In this section, we present the results obtained by the current version of the IED method which uses genetic algorithms for minimizing the Lagrangian.

Regarding the function  $\sigma$  (see (1)), we have used seven norms in our first set of numerical experiments, namely,  $\|\cdot\|_1$ ,  $\|\cdot\|_2$ ,  $\|\cdot\|_\infty$ ,  $e^{\|\cdot\|_2} - 1$ ,  $e^{\|\cdot\|_1} - 1$ ,  $\sqrt{m} (e^{\|\cdot\|_\infty} - 1)$ , and  $\max(\|\cdot\|_2, \|\cdot\|_2^2)$ . Usually, those norms are used in order to check the efficiency of algorithm [1, 2, 25]. Therefore, we have kept the norms to compare the results obtained by our algorithm.

With respect to the matrix  $A$ , we have considered  $A = I$  which corresponds to the augmented Lagrangian case.

Inequality constraints  $g(x) \leq 0$  have been replaced with their nonsmooth equivalent  $\max\{g(x), 0\} = 0$  so that the problems have been converted into equality constrained optimization problems.

As the stopping criterion for IED we have used  $\|g(x)\| \leq 10^{-6}$ . A total of 10 independent runs have been executed by the IED + GA, extracting the best solution, the worst solution, the mean of the solutions, and the standard deviation (sd), for the first set of experiments. For the second set, we have executed 30 independent runs.

The algorithms have been implemented in Matlab - (R2012b) - in a microcomputer core (TM) i7 of 2.60 GHz with 8.00 GB of RAM.

This section is divided into two sets of test problems.

**7.1. The First Set of Numerical Experiments.** This set of numerical experiments corresponds to the analysis of 20 constrained optimization problems extracted from the Hock and Schittkowski collection [26] and all of them refer to minimization problems. They are described below were the objective  $f(x)$  and the constraints are presented.

*GLR-PI-1:*

$$\begin{aligned} f(x) = & -32,174 \left( 255 \right. \\ & \cdot \ln \left( \frac{(x_1 + x_2 + x_3 + 0,03)}{(0,09x_1 + x_2 + x_3 + 0,03)} \right) + 280 \\ & \cdot \ln \left( \frac{(x_2 + x_3 + 0,03)}{(0,07x_2 + x_3 + 0,03)} \right) + 290 \\ & \left. \cdot \ln \left( \frac{(x_3 + 0,03)}{(0,13x_3 + 0,03)} \right) \right) \end{aligned} \quad (34)$$

subject to

$$\begin{aligned} x_1 + x_2 + x_3 &= 0 \\ 0 \leq x_i &\leq 1, \quad i = 1, 2, 3 \end{aligned} \quad (35)$$

*GQR-PI-1:*

$$\begin{aligned} f(x) = & -75,196 + 3,8112x_1 + 0,0020567x_1^3 \\ & - 1,0345 (e - 5) x_1^4 + 6,8306x_2 \\ & - 0,030234x_1x_2 + 1,28134 (e - 3) x_2x_1^2 \\ & + 2,266 (e - 7) x_1^4x_2 - 0,25645x_2^2 \\ & + 0,0034604x_2^3 - 1,3514 (e - 5) x_2^4 \\ & + \frac{28,106}{(x_2 + 1)} + 5,2375e - 6x_1^2x_2^2 \\ & + 6,3 (e - 8) x_1^3x_2^2 - 7 (e - 10) x_1^3x_2^3 \\ & - 3,405 (e - 4) x_1x_2^2 + 1,6638 (e - 6) x_1x_2^3 \\ & + 2,8673 \exp(0,0005x_1x_2) \\ & - 3,5256 (e - 5) x_1^3x_2 \end{aligned} \quad (36)$$

subject to

$$\begin{aligned} x_1x_2 - 700 &\geq 0 \\ x_2 - \frac{x_1^2}{125} &\geq 0 \\ (x_2 - 50)^2 - 5(x_1 - 55) &\geq 0 \\ 0 \leq x_1 &\leq 75 \\ 0 \leq x_2 &\leq 65 \end{aligned} \quad (37)$$

*PPR-PI-2:*

$$\begin{aligned} f(x) = & 5x_1 + \frac{50000}{x_1} + 20x_2 + \frac{72000}{x_2} + 10x_3 \\ & + \frac{144000}{x_3} \end{aligned} \quad (38)$$

subject to

$$1 - \frac{4}{x_1} - \frac{32}{x_2} - \frac{120}{x_3} \quad (39)$$

$$1 \cdot 10^{-5} \leq x_i, \quad i = 1, 2, 3$$

*PQR-TI-7:*

$$f(x) = -x_1 x_2 x_3 \quad (40)$$

subject to

$$-x_1^2 - 2x_2^2 - 4x_3^2 + 48 \geq 0 \quad (41)$$

*SQR-PI-1:*

$$f(x) = \sum_{i=1}^{44} f_i(x)^2, \quad (42)$$

$$f_i(x) = b_i - x_1 - (0,49 - x_1) \exp(-x_2(a_i - 8)), \\ i = 1, \dots, 44$$

subject to

$$0,49x_2 - x_1 x_2 - 0,09 \geq 0 \\ 0,4 \leq x_1 \\ -4 \leq x_2 \quad (43)$$

*LGR-PI-1:*

$$f(x) = 0,2x_3 - 0,8x_1 \quad (44)$$

subject to

$$x_2 - \exp(x_1) \geq 0 \\ x_3 - \exp(x_2) \geq 0 \\ 0 \leq x_1 \leq 100 \\ 0 \leq x_2 \leq 100 \\ 0 \leq x_3 \leq 10 \quad (45)$$

*QBR-TI-1:*

$$f(x) = x_2 + 10^{-5}(x_2 - x_1)^2 \quad (46)$$

subject to

$$0 \leq x_2 \quad (47)$$

*PBR-TI-1:*

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \quad (48)$$

subject to

$$-x_2 - 1,5 \leq 0 \quad (49)$$

*PBR-TI-2:*

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \quad (50)$$

subject to

$$1,5 \leq x_2 \quad (51)$$

*QQR-PI-3:*

$$f(x) = (x_1 - x_2)^2 + \frac{(x_1 + x_2 - 10)^2}{9} + (x_3 - 5)^2 \quad (52)$$

subject to

$$48 - x_1^2 - x_2^2 - x_3^2 \geq 0 \\ -4,5 \leq x_i \leq 4,5, \quad i = 1, 2 \\ -5 \leq x_3 \leq 5 \quad (53)$$

*QQR-TI-6:*

$$f(x) = (x_1 - 2)^2 + (x_2 - 1)^2 \quad (54)$$

subject to

$$x_1^2 - x_2 \leq 0 \\ x_1 + x_2 - 2 \leq 0 \quad (55)$$

*PLR-TI-1:*

$$f(x) = \frac{1}{(27\sqrt{3})((x_1 - 3)^2 - 9)x_2^3} \quad (56)$$

subject to

$$\frac{x_1}{\sqrt{3}} - x_2 \geq 0 \\ x_1 + \sqrt{3}x_2 \geq 0 \\ -x_1 + \sqrt{3}x_2 - 6 \geq 0 \\ 0 \leq x_1 \\ 0 \leq x_2 \quad (57)$$

*PBR-TI-3:*

$$f(x) = \frac{1}{3(x_1 + 1)^3} + x_2 \quad (58)$$

subject to

$$1 \leq x_1 \\ 0 \leq x_2 \quad (59)$$

QLR-TI-1:

$$f(x) = 0,01x_1^2 + x_2^2 - 100 \quad (60)$$

subject to

$$\begin{aligned} 10x_1 - x_2 - 10 &\geq 0 \\ 2 &\leq x_1 \leq 50 \\ -50 &\leq x_2 \leq 50 \end{aligned} \quad (61)$$

PQR-TI-1:

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \quad (62)$$

subject to

$$\begin{aligned} x_1x_2 - 1 &\geq 0 \\ x_1 + x_2^2 &\geq 0 \\ x_1 &\leq 0,5 \end{aligned} \quad (63)$$

QQR-TI-3:

$$f(x) = 0,5x_1^2 + x_2^2 - x_1x_2 - 7x_1 - 7x_2 \quad (64)$$

subject to

$$-25 + 4x_1^2 + x_2^2 \leq 0 \quad (65)$$

GBR-TI-1:

$$f(x) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - 1,5x_1 + 2,5x_2 + 1 \quad (66)$$

subject to

$$\begin{aligned} -1,5 &\leq x_1 \leq 4 \\ -3 &\leq x_2 \leq 3 \end{aligned} \quad (67)$$

PQR-TI-4:

$$f(x) = (x_1 - 10)^3 + (x_2 - 20)^3 \quad (68)$$

subject to

$$\begin{aligned} -100 + (x_1 - 5)^2 + (x_2 - 5)^2 &\geq 0 \\ -(x_2 - 5)^2 - (x_1 - 6)^2 + 82,81 &\geq 0 \\ 13 &\leq x_1 \leq 100 \\ 0 &\leq x_2 \leq 100 \end{aligned} \quad (69)$$

QQR-TI-2:

$$f(x) = (x_1 - 5)^2 + x_2^2 - 25 \quad (70)$$

subject to

$$-x_1^2 + x_2^2 \geq 0 \quad (71)$$

QPR-TI-1:

$$f(x) = (x_1 - 2)^2 + x_2^2 \quad (72)$$

subject to

$$\begin{aligned} (1 - x_1)^3 - x_2 &\geq 0 \\ 0 &\leq x_1 \\ 0 &\leq x_2 \end{aligned} \quad (73)$$

7.1.1. *Comparison of Results of the First Set of Numerical Experiments.* Table 1 provides the initial guesses  $c_0$  and  $u_0$  used by IED + fmin and IED + GA, the size of the population, the number of generations, and bound constraints used by the Genetic Algorithm. The table also shows the optimal values.

For solving the first six problems, we have kept the initial guesses used by the IED in its previous version as described in [1]. For the rest of the problems,  $c_0$  and  $u_0$  were randomly chosen.

One can notice from Tables 2–21 that IED + GA has found the solution for all the problems regardless of the norms used whereas IED + fmin failed some of them.

It is interesting to highlight that the standard deviation of all the problems is small. This shows robustness of the version of the IED method proposed in this work.

Tables 2–21 show the results of the problems obtained by both versions of the algorithm. For the IED + GA algorithm, we present the best and the worst results, the mean and the standard deviation. Cases where a solution has been not found are indicated by a line struck through the entry of the table.

7.2. *The Second Set of Numerical Experiments.* In this section the algorithm proposed is applied to solve 4 real-world mechanical engineering optimization problems, largely used as a test-bed in the literature.

*The Tension/Compression String Design.* In this problem, the objective is to minimize the volume  $V$  of a coil spring, depicted in the Figure 8, under a constant tension/compression load. There are three design variables to be considered in this problem: The number  $x_1 = N$  of active coils of the spring, the winding diameter  $x_2 = D$  and the wire diameter  $x_3 = d$ . The volume of the coil to be minimized is written as [27]

$$V(x) = (x_1 + 2)x_2x_3^2 \quad (74)$$

and is subject to the constraints

$$g_1(x) = 1 - \frac{x_2^3x_1}{71785x_3^4} \leq 0$$

$$g_2(x) = \frac{4x_2^2 - x_3x_2}{12566(x_2x_3^3 - x_3^4)} + \frac{1}{5108x_3^2} - 1 \leq 0$$

TABLE 1: Parameters of the IED method.

Test-problem	$c_0$	$u_0$	Population Size	Generations	Lower Bound	Upper Bound	Optimal Value
GLR-PI-1	50	[0 0 0 0]	50	300	[0.01, 0.01, 0.01]	[1, 1, 1]	$-2.6273 \times 10^4$
GQR-PI-1	1	[1 1 1 1 1 1 1]	50	100	[10, 10]	[60, 60]	-7.8028
PPR-PI-2	0.1	[1 1 1 1]	50	200	[0, 0, 0]	[300, 300, 300]	$6.2998 \times 10^3$
PQR-TI-7	1.5	[1 1 1 1 1 1 1]	50	100	[-5, -5, -5]	[5, 5, 5]	-22.6274
SQR-PI-1	0.2	[0 0 0]	50	100	[-10, -10]	[10, 10]	0.0285
LGR-PI-1	0.1	[0 0 0 0 0 0 0 0]	50	100	[0, 0, 0]	[5, 5, 5]	0.5181
QBR-TI-1	1	[1 1 1 1 1 1 1 1]	50	100	[0, 0, 0]	[5, 5, 5]	0
PBR-TI-1	1	[1]	50	100	[-2, -2]	[2, 2]	0
PBR-TI-2	1	[1]	50	100	[-2, -2]	[2, 2]	0.0504
QQR-PI-3	1	[1 1 1 1 1 1 1]	50	100	[0, 0, 0]	[5, 5, 5]	0.9535
QQR-TI-6	1	[1 1]	50	100	[-2, -2]	[2, 2]	1
PLR-TI-1	1	[1 1 1 1 1]	50	100	[1, 1]	[4, 4]	-1
PBR-TI-3	1	[1 1]	50	100	[-1, -1]	[2, 2]	2.6666
QLR-TI-1	1	[1 1 1 1 1]	50	100	[-1, -1]	[3, 3]	-99.96
PQR-TI-1	5	[1 1 1]	50	100	[0, 0]	[3, 3]	306.5
QQR-TI-3	0.1	[1]	20	50	[1, 1]	[4, 4]	-30
GBR-TI-1	1	[1 1 1 1]	50	100	[-2, -2]	[0, 0]	-1.9132
PQR-TI-4	1	[1 1 1 1 1 1]	50	100	[0, 0]	[15, 15]	-6961.8138
QQR-TI-2	1	[1]	20	50	[0, 0]	[2, 2]	-8.4984
QPR-TI-1	0.1	[1 1 1]	50	100	[-1, -1]	[2, 2]	1

TABLE 2: Comparison of the results obtained by both versions of the IED for problem GLR-PI-1.

$\sigma(\cdot)$	GLR-PI-1				
	IED + fmin result	best	worst	IED + GA mean	sd
$\ \cdot\ _1$	$-2.6273 \times 10^4$	$-2.6272 \times 10^4$	$-2.6267 \times 10^4$	$-2.6271 \times 10^4$	2.0661
$\ \cdot\ _2$	$-2.6273 \times 10^4$	$-2.6272 \times 10^4$	$-2.6267 \times 10^4$	$-2.6271 \times 10^4$	2.0661
$\sqrt{m} \ \cdot\ _\infty$	—	$-2.6272 \times 10^4$	$-2.6267 \times 10^4$	$-2.6271 \times 10^4$	2.0661
$e^{\ \cdot\ _2} - 1$	$-2.6273 \times 10^4$	$-2.6273 \times 10^4$	$-2.6270 \times 10^4$	$-2.6272 \times 10^4$	0.8555
$e^{\ \cdot\ _1} - 1$	$-2.6273 \times 10^4$	$-2.6273 \times 10^4$	$-2.6270 \times 10^4$	$-2.6272 \times 10^4$	0.8555
$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	—	$-2.6272 \times 10^4$	$-2.6225 \times 10^4$	$-2.6261 \times 10^4$	14.5763
$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	—	$-2.6273 \times 10^4$	$-2.4727 \times 10^4$	$-2.6049 \times 10^4$	477.8463

TABLE 3: Comparison of the results obtained by both versions of the IED for problem GQR-PI-1.

$\sigma(\cdot)$	GQR-PI-1				
	IED + fmin result	best	worst	IED + GA mean	sd
$\ \cdot\ _1$	—	-7.7978	-6.7495	-7.5575	0.4268
$\ \cdot\ _2$	—	-7.7978	-6.7495	-7.5572	0.4276
$\sqrt{m} \ \cdot\ _\infty$	-7.8028	-7.7984	-6.7495	-7.0554	0.4927
$e^{\ \cdot\ _2} - 1$	-7.8028	-7.7925	-6.7495	-7.4544	0.4872
$e^{\ \cdot\ _1} - 1$	-7.8028	-7.8012	-7.7443	-7.7801	0.0171
$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	-7.8028	-7.7985	-6.7495	-7.4601	0.4914
$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	-7.8028	-7.8024	-6.7495	-7.3560	0.5228

$$g_3(x) = 1 - \frac{140.45x_3}{x_2^2x_1} \leq 0$$

$$g_4(x) = \frac{x_2 + x_3}{1.5} - 1 \leq 0,$$

(75)

where

$$2 \leq x_1 \leq 15$$

$$0.25 \leq x_2 \leq 1.3$$

$$0.05 \leq x_3 \leq 2.$$

(76)

TABLE 4: Comparison of the results obtained by both versions of the IED for problem PPR-P1-2.

$\sigma(\cdot)$	PPR-P1-2				
	IED + fmin	IED + GA			
	result	best	worst	mean	sd
$\ \cdot\ _1$	$6.2998 \times 10^3$	$6.3002 \times 10^3$	$6.3018 \times 10^3$	$6.3008 \times 10^3$	0.4452
$\ \cdot\ _2$	$6.2998 \times 10^3$	$6.3002 \times 10^3$	$6.3018 \times 10^3$	$6.3008 \times 10^3$	0.4452
$\sqrt{m} \ \cdot\ _\infty$	$6.2998 \times 10^3$	$6.3001 \times 10^3$	$6.3035 \times 10^3$	$6.3015 \times 10^3$	1.0803
$e^{\ \cdot\ _2} - 1$	—	$6.3000 \times 10^3$	$6.3023 \times 10^3$	$6.3007 \times 10^3$	0.7107
$e^{\ \cdot\ _1} - 1$	—	$6.3000 \times 10^3$	$6.3023 \times 10^3$	$6.3007 \times 10^3$	0.7107
$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	—	$6.3000 \times 10^3$	$6.3053 \times 10^3$	$6.3012 \times 10^3$	1.6442
$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	$6.2999 \times 10^3$	$6.2999 \times 10^3$	$6.3045 \times 10^3$	$6.3013 \times 10^3$	1.5406

TABLE 5: Comparison of the results obtained by both versions of the IED for problem PQR-T1-7.

$\sigma(\cdot)$	PQR-T1-7				
	IED + fmin	IED + GA			
	result	best	worst	mean	sd
$\ \cdot\ _1$	—	-22.5467	-22.1224	-22.4160	0.1450
$\ \cdot\ _2$	—	-22.6033	-22.1466	-22.4274	0.1383
$\sqrt{m} \ \cdot\ _\infty$	-22.6274	-22.6233	-22.3691	-22.5153	0.1151
$e^{\ \cdot\ _2} - 1$	-22.6262	-22.5424	-22.0668	-22.3474	0.1526
$e^{\ \cdot\ _1} - 1$	-22.6270	-22.5468	-21.9238	-22.2471	0.1891
$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	-22.6274	-22.5215	21.6514	-17.8111	13.8676
$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	-22.6156	-22.5704	-22.0331	-22.3381	0.1683

TABLE 6: Comparison of the results obtained by both versions of the IED for problem SQR-P1-1.

$\sigma(\cdot)$	SQR-P1-1				
	IED + fmin	IED + GA			
	result	best	worst	mean	sd
$\ \cdot\ _1$	0.0285	0.0285	0.0287	0.0286	$8.4993 \times 10^{-5}$
$\ \cdot\ _2$	0.0285	0.0285	0.0287	0.0286	$8.2863 \times 10^{-5}$
$\sqrt{m} \ \cdot\ _\infty$	0.0285	0.0285	0.0293	0.0287	$2.3577 \times 10^{-4}$
$e^{\ \cdot\ _2} - 1$	0.0285	0.0285	0.0293	0.0287	$5.7703 \times 10^{-4}$
$e^{\ \cdot\ _1} - 1$	0.0285	0.0285	0.0306	0.0289	$6.5958 \times 10^{-4}$
$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	0.0285	0.0285	0.0290	0.0286	$1.8364 \times 10^{-4}$
$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	0.0285	0.0285	0.0289	0.0286	$1.2819 \times 10^{-4}$

TABLE 7: Comparison of the results obtained by both versions of the IED for problem LGR-P1-1.

$\sigma(\cdot)$	LGR-P1-1				
	IED + fmin	IED + GA			
	result	best	worst	mean	sd
$\ \cdot\ _1$	0.5187	0.5234	0.5407	0.5314	0.0061
$\ \cdot\ _2$	—	0.5216	0.5427	0.5301	0.0083
$\sqrt{m} \ \cdot\ _\infty$	0.5228	0.5213	0.5726	0.5364	0.0157
$e^{\ \cdot\ _2} - 1$	0.5199	0.5213	0.5726	0.5364	0.0157
$e^{\ \cdot\ _1} - 1$	0.5195	0.5232	0.5622	0.5384	0.0125
$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	0.5183	0.5239	0.5677	0.5390	0.0161
$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	0.5182	0.5201	0.5488	0.5313	0.0090

TABLE 8: Comparison of the results obtained by both versions of the IED for problem QBR-T1-1.

$\sigma(\cdot)$	QBR-T1-1				
	IED + fmin	IED + GA			
	result	best	worst	mean	sd
$\ \cdot\ _1$	—	$5.4134 \times 10^{-7}$	$1.5704 \times 10^{-4}$	$6.7404 \times 10^{-5}$	$5.3498 \times 10^{-5}$
$\ \cdot\ _2$	—	$5.4134 \times 10^{-7}$	$1.5704 \times 10^{-4}$	$6.7404 \times 10^{-5}$	$5.3498 \times 10^{-5}$
$\sqrt{m} \ \cdot\ _\infty$	—	$5.4134 \times 10^{-7}$	$1.5704 \times 10^{-4}$	$6.7404 \times 10^{-5}$	$5.3498 \times 10^{-5}$
$e^{\ \cdot\ _2} - 1$	$3.668 \times 10^{-17}$	$7.2718 \times 10^{-6}$	$2.0872 \times 10^{-4}$	$9.4788 \times 10^{-5}$	$8.2756 \times 10^{-5}$
$e^{\ \cdot\ _1} - 1$	$3.668 \times 10^{-17}$	$7.2718 \times 10^{-6}$	$2.0872 \times 10^{-4}$	$9.4788 \times 10^{-5}$	$8.2756 \times 10^{-5}$
$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	$3.668 \times 10^{-17}$	$7.2718 \times 10^{-6}$	$2.0872 \times 10^{-4}$	$9.4788 \times 10^{-5}$	$8.2756 \times 10^{-5}$
$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	$2.4563 \times 10^{-14}$	$5.4134 \times 10^{-7}$	$1.5704 \times 10^{-4}$	$6.7404 \times 10^{-5}$	$5.3498 \times 10^{-5}$

TABLE 9: Comparison of the results obtained by both versions of the IED for problem PBR-T1-1.

$\sigma(\cdot)$	PBR-T1-1				
	IED + fmin	IED + GA			
	result	best	worst	mean	sd
$\ \cdot\ _1$	$-9.0273x10^{-14}$	0	0.0134	0.0027	0.0052
$\ \cdot\ _2$	$-9.0273x10^{-14}$	0	0.0134	0.0027	0.0052
$\sqrt{m} \ \cdot\ _\infty$	$-9.0273x10^{-14}$	0	0.0134	0.0027	0.0052
$e^{\ \cdot\ _2} - 1$	$-9.0273x10^{-14}$	0	0.0134	0.0027	0.0052
$e^{\ \cdot\ _1} - 1$	$-9.0273x10^{-14}$	0	0.0134	0.0027	0.0052
$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	$-9.0273x10^{-14}$	0	0.0134	0.0027	0.0052
$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	$-9.0273x10^{-14}$	0	0.0134	0.0027	0.0052

TABLE 10: Comparison of the results obtained by both versions of the IED for problem PBR-T1-2.

$\sigma(\cdot)$	PBR-T1-2				
	IED + fmin	IED + GA			
	result	best	worst	mean	sd
$\ \cdot\ _1$	0.0504	0.0506	0.0677	0.0536	0.0052
$\ \cdot\ _2$	0.0504	0.0506	0.0677	0.0536	0.0052
$\sqrt{m} \ \cdot\ _\infty$	0.0504	0.0506	0.0677	0.0536	0.0052
$e^{\ \cdot\ _2} - 1$	0.0504	0.0508	0.0839	0.0593	0.0123
$e^{\ \cdot\ _1} - 1$	0.0504	0.0508	0.0839	0.0593	0.0123
$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	0.0504	0.0508	0.0839	0.0593	0.0123
$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	0.0504	0.0505	0.0790	0.0545	0.0088

TABLE 11: Comparison of the results obtained by both versions of the IED for problem QQR-P1-3.

$\sigma(\cdot)$	QQR-P1-3				
	IED + fmin	IED + GA			
	result	best	worst	mean	sd
$\ \cdot\ _1$	0.9535	0.9576	0.9837	0.9676	0.0099
$\ \cdot\ _2$	0.9535	0.9565	0.9853	0.9637	0.0890
$\sqrt{m} \ \cdot\ _\infty$	0.9535	0.9550	0.9881	0.9665	0.0115
$e^{\ \cdot\ _2} - 1$	0.9535	0.9550	0.9684	0.9603	0.0039
$e^{\ \cdot\ _1} - 1$	0.9535	0.9541	0.9860	0.9652	0.0096
$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	-25.0694	0.9558	0.9683	0.9628	0.0040
$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	0.9535	0.9568	1.0124	0.9708	0.0185

TABLE 12: Comparison of the results obtained by both versions of the IED for problem QQR-T1-6.

$\sigma(\cdot)$	QQR-T1-6				
	IED + fmin		IED + GA		
	result	best	worst	mean	sd
$\ \cdot\ _1$	1	1.0002	1.0185	1.0052	0.0057
$\ \cdot\ _2$	1	1.0023	1.0179	1.0101	0.0044
$\sqrt{m} \ \cdot\ _\infty$	1	1.0003	1.0404	1.0112	0.0125
$e^{\ \cdot\ _2} - 1$	1	1.0005	1.0287	1.0126	0.0093
$e^{\ \cdot\ _1} - 1$	1	1.0005	1.0400	1.0122	0.0137
$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	1	1.0002	1.0366	1.0133	0.0106
$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	1	1.0002	1.0097	1.0046	0.0037

TABLE 13: Comparison of the results obtained by both versions of the IED for problem PLR-T1-1.

$\sigma(\cdot)$	PLR-T1-1				
	IED + fmin		IED + GA		
	result	best	worst	mean	sd
$\ \cdot\ _1$	—	-0.9976	-0.9729	-0.9915	0.0071
$\ \cdot\ _2$	—	-0.9966	-0.9817	-0.9892	0.0052
$\sqrt{m} \ \cdot\ _\infty$	—	-0.9984	-0.9663	-0.9899	0.0102
$e^{\ \cdot\ _2} - 1$	-1	-0.9974	-0.9806	-0.9916	0.0057
$e^{\ \cdot\ _1} - 1$	-1	-0.9966	-0.9805	-0.9925	0.0048
$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	-1	-0.9993	-0.9793	-0.9926	0.0061
$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	-1	-0.9996	-0.9758	-0.9920	0.0073

TABLE 14: Comparison of the results obtained by both versions of the IED for problem PBR-T1-3.

$\sigma(\cdot)$	PBR-T1-3				
	IED + fmin		IED + GA		
	result	best	worst	mean	sd
$\ \cdot\ _1$	—	2.6686	2.7135	2.6827	0.0134
$\ \cdot\ _2$	—	2.6698	2.6855	2.6743	0.0050
$\sqrt{m} \ \cdot\ _\infty$	—	2.6697	2.6909	2.6784	0.0079
$e^{\ \cdot\ _2} - 1$	2.6667	2.6692	2.6947	2.6784	0.0082
$e^{\ \cdot\ _1} - 1$	2.6667	2.6685	2.7022	2.6809	0.0110
$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	2.6667	2.6687	2.7059	2.6909	0.0125
$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	2.6667	2.6684	2.7125	2.6847	0.0150

TABLE 15: Comparison of the results obtained by both versions of the IED for problem QLR-T1-1.

$\sigma(\cdot)$	QLR-T1-1				
	IED + fmin		IED + GA		
	result	best	worst	mean	sd
$\ \cdot\ _1$	-99.9600	-99.9600	-99.9597	-99.9599	$8.9350 \times 10^{-5}$
$\ \cdot\ _2$	-99.9600	-99.9600	-99.9595	-99.9598	$1.4311 \times 10^{-4}$
$\sqrt{m} \ \cdot\ _\infty$	-99.9600	-99.9600	-99.9599	-99.9599	$5.1418 \times 10^{-5}$
$e^{\ \cdot\ _2} - 1$	-99.9600	-99.9600	-99.9594	-99.9599	$1.5289 \times 10^{-4}$
$e^{\ \cdot\ _1} - 1$	-99.9600	-99.9600	-99.9595	-99.9599	$1.3253 \times 10^{-4}$
$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	-99.9600	-99.9600	-99.9594	-99.9598	$1.8280 \times 10^{-4}$
$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	-99.9600	-99.9599	-99.9598	-99.9599	$3.7864 \times 10^{-5}$

TABLE 16: Comparison of the results obtained by both versions of the IED for problem PQR-T1-1.

$\sigma(\cdot)$	PQR-T1-1				
	IED + fmin	IED + GA			
	result	best	worst	mean	sd
$\ \cdot\ _1$	306.5	306.5146	326.6933	310.9483	6.2614
$\ \cdot\ _2$	306.5	308.7375	329.3517	312.3063	6.1269
$\sqrt{m} \ \cdot\ _\infty$	306.5	307.5067	327.2424	311.6510	5.8228
$e^{\ \cdot\ _2} - 1$	306.5	308.9435	317.7634	312.9006	2.7421
$e^{\ \cdot\ _1} - 1$	306.5	306.5408	315.6229	310.7872	3.0505
$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	306.5	307.8027	316.7068	312.1473	2.7962
$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	306.5	307.9935	315.4329	311.4265	2.4128

TABLE 17: Comparison of the results obtained by both versions of the IED for problem QQR-T1-3.

$\sigma(\cdot)$	QQR-T1-3				
	IED + fmin	IED + GA			
	result	best	worst	mean	sd
$\ \cdot\ _1$	—	-29.9998	-29.8286	-29.9714	0.0532
$\ \cdot\ _2$	—	-29.9998	-29.8286	-29.9714	0.532
$\sqrt{m} \ \cdot\ _\infty$	—	-29.9998	-29.8286	-29.9714	0.0532
$e^{\ \cdot\ _2} - 1$	-30	-29.9972	-29.3533	-29.8566	0.2162
$e^{\ \cdot\ _1} - 1$	-30	-29.9972	-29.3533	-29.8566	0.2162
$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	-30	-29.9972	-29.3533	-29.8566	0.2162
$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	-30	-29.9960	-29.2340	-29.8780	0.2316

TABLE 18: Comparison of the results obtained by both versions of the IED for problem GBR-T1-1.

$\sigma(\cdot)$	GBR-T1-1				
	IED + fmin	IED + GA			
	result	best	worst	mean	sd
$\ \cdot\ _1$	-1.9132	-1.9132	-1.9132	-1.9132	$5.1999 \times 10^{-7}$
$\ \cdot\ _2$	-1.9132	-1.9132	-1.9132	-1.9132	$5.1999 \times 10^{-7}$
$\sqrt{m} \ \cdot\ _\infty$	-1.9132	-1.9132	-1.9132	-1.9132	$1.4981 \times 10^{-6}$
$e^{\ \cdot\ _2} - 1$	-1.9132	-1.9132	-1.9132	-1.9132	$9.3127 \times 10^{-7}$
$e^{\ \cdot\ _1} - 1$	-1.9132	-1.9132	-1.9132	-1.9132	$9.3127 \times 10^{-7}$
$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	-1.9132	-1.9132	-1.9132	-1.9132	$1.2166 \times 10^{-6}$
$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	-1.9132	-1.9132	-1.9132	-1.9132	$5.1999 \times 10^{-7}$

TABLE 19: Comparison of the results obtained by both versions of the IED for problem PQR-T1-4.

$\sigma(\cdot)$	PQR-T1-4				
	IED + fmin	IED + GA			
	result	best	worst	mean	sd
$\ \cdot\ _1$	—	-6791.5	173850	1237.6	$5.6747 \times 10^4$
$\ \cdot\ _2$	—	-6936.9	1792.1	-5153.5	$2.8982 \times 10^3$
$\sqrt{m} \ \cdot\ _\infty$	—	-6800.2	-2548.9	-5962.7	$1.3694 \times 10^3$
$e^{\ \cdot\ _2} - 1$	-6961.8	-6874.3	-6258.0	-6737.2	188.5621
$e^{\ \cdot\ _1} - 1$	-6961.8	-6915.8	-1642.2	-6165.1	1606.1
$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	-6961.8	-6777.9	-6159.4	-6590.4	222.8402
$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	-6961.8	-6958.0	-1735.3	-6274.8	1609.0

TABLE 20: Comparison of the results obtained by both versions of the IED for problem QQR-TI-2.

$\sigma(\cdot)$	QQR-TI-2				
	IED + fmin result	best	worst	IED + GA mean	sd
$\ \cdot\ _1$	-8.4985	-8.4958	-8.3391	-8.4729	0.0481
$\ \cdot\ _2$	-8.4985	-8.4958	-8.3391	-8.4729	0.0481
$\sqrt{m} \ \cdot\ _\infty$	-8.4985	-8.4958	-8.3391	-8.4729	0.0481
$e^{\ \cdot\ _2} - 1$	-8.4985	-8.4958	-7.5694	-8.2675	0.3201
$e^{\ \cdot\ _1} - 1$	-8.4985	-8.4958	-7.5694	-8.2675	0.3201
$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	-8.4985	-8.4958	-7.5694	-8.2675	0.3201
$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	-8.4985	-8.4980	-8.0434	-8.3737	0.1564

TABLE 21: Comparison of the results obtained by both versions of the IED for problem QPR-TI-1.

$\sigma(\cdot)$	QPR-TI-1				
	IED + fmin result	best	worst	IED + GA mean	sd
$\ \cdot\ _1$	—	1.1349	1.4578	1.2706	0.0846
$\ \cdot\ _2$	—	1.0632	1.4943	1.2132	0.1356
$\sqrt{m} \ \cdot\ _\infty$	—	1.1317	1.6091	1.2670	0.1421
$e^{\ \cdot\ _2} - 1$	0.9978	1.1316	1.4839	1.2715	0.1220
$e^{\ \cdot\ _1} - 1$	—	1.0729	1.5440	1.2488	0.1375
$\sqrt{m}(e^{\ \cdot\ _\infty} - 1)$	0.9979	1.0818	1.4799	1.2300	0.1555
$\max(\ \cdot\ _2, \ \cdot\ _2^2)$	0.9987	1.0967	1.6843	1.2762	0.1853

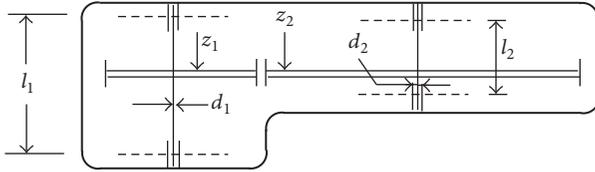


FIGURE 9: The Speed Reducer design.

*The Speed Reducer Design.* The objective of this problem is to minimize the weight  $W$  of the speed reducer [27] shown in the Figure 9. The design variables are the face width ( $x_1 = b$ ), the module of teeth ( $x_2 = m$ ), the number of teeth on pinion ( $x_3 = n$ ), the length of the shaft 1 between the bearings ( $x_4 = l_1$ ), the length of the shaft 2 between the bearings ( $x_5 = l_2$ ), the diameter of the shaft 1 ( $x_6 = d_1$ ), and, finally, the diameter of the shaft 2 ( $x_7 = d_2$ ). The third variable is integer and all the others are continuous. The constraints include limitations on the bending and surface stress of the gear teeth, transverse deflections of the shafts 1 and 2 generated by the transmitted force, and, finally, the stress in the shafts 1 and 2. The weight of the speed reducer, to be minimized, is given by

$$\begin{aligned}
 W(x) &= 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) \\
 &\quad - 1.508x_1(x_6^2 + x_7^2) + 7.477(x_6^3 + x_7^3) \\
 &\quad + 0.7854(x_4x_6^2 + x_5x_7^2)
 \end{aligned} \tag{77}$$

subject to

$$\begin{aligned}
 g_1(x) &= 27x_1^{-1}x_2^{-2}x_3^{-1} \leq 1 \\
 g_2(x) &= 397.5x_1^{-1}x_2^{-2}x_3^{-2} \leq 1 \\
 g_3(x) &= 1.93x_2^{-1}x_3^{-1}x_4^3x_6^{-4} \leq 1 \\
 g_4(x) &= 1.93x_2^{-1}x_3^{-1}x_5^3x_7^{-4} \leq 1 \\
 g_5(x) &= \frac{[(745x_4/x_2x_3)^2 + 16.9 \times 10^6]^{0.5}}{0.1x_6^3} \leq 1100 \\
 g_6(x) &= \frac{[(745x_5/x_2x_3)^2 + 157.5 \times 10^6]^{0.5}}{0.1x_7^3} \leq 850 \\
 g_7(x) &= x_2x_3 \leq 40 \\
 g_8(x) &= \frac{x_1}{x_2} \geq 5 \\
 g_9(x) &= \frac{x_1}{x_2} \leq 12 \\
 g_{10}(x) &= (1.5x_6 + 1.9)x_4^{-1} \leq 1 \\
 g_{11}(x) &= (1.1x_7 + 1.9)x_5^{-1} \leq 1 \\
 2.6 &\leq x_1 \leq 3.6 \\
 0.7 &\leq x_2 \leq 0.8
 \end{aligned}$$

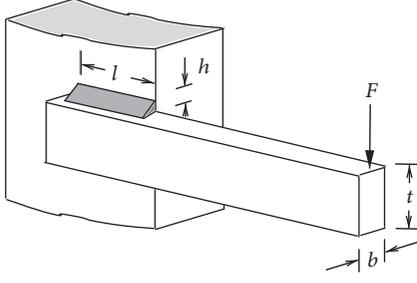


FIGURE 10: The welded beam.

$$17 \leq x_3 \leq 28$$

$$7.3 \leq x_4 \leq 8.3$$

$$7.8 \leq x_5 \leq 8.3$$

$$2.9 \leq x_6 \leq 3.9$$

$$5.0 \leq x_6 \leq 5.5.$$

(78)

*The Welded Beam.* This problem corresponds to the design of the welded beam [27] depicted in the Figure 10. The design variables are  $\{h, l, t, b\}$ , with bounds  $0.125 \leq h \leq 10$ , and  $0.1 \leq l, t, b \leq 10$ . The objective function to be minimized is the cost of the beam given as

$$C(h, l, t, b) = 1.10471h^2l + 0.04811tb(14.0 + l) \quad (79)$$

subject to

$$g_1(\tau) = 1 - \frac{\tau}{13600} \geq 0$$

$$g_2(\sigma) = 1 - \frac{\sigma}{30000} \geq 0$$

$$g_3(b, h) = b - h \geq 0 \quad (80)$$

$$g_4(P_c) = \frac{P_c}{6000} - 1 \geq 0$$

$$g_5(\delta) = 0.25 - \delta \geq 0.$$

The expressions for  $\tau$ ,  $\sigma$ ,  $P_c$ , and  $\delta$  involve the design variables and are given by

$$\tau = \sqrt{\tau'^2 + \tau''^2 + \frac{l\tau'\tau''}{\alpha}}$$

$$\alpha = \sqrt{0.25(l^2 + (h+t)^2)},$$

$$\sigma = \frac{504000}{t^2b}$$

$$P_c = 64746.022(1 - 0.0282346t)tb^3$$

$$\delta = \frac{2.1952}{t^3b},$$

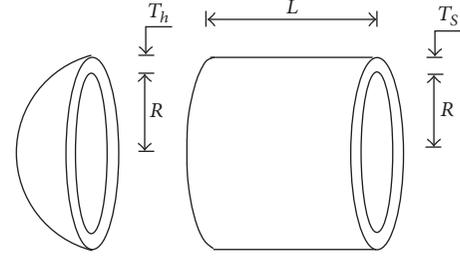


FIGURE 11: The pressure vessel.

$$\tau' = \frac{6000}{\sqrt{2hl}}$$

$$\tau'' = \frac{6000(14 + 0.5l)\alpha}{2(0.707hl(l^2/12 + 0.25(h+t)^2))}.$$

(81)

*The Pressure Vessel.* This problem corresponds to the weight minimization of a cylindrical pressure vessel with two spherical heads [27] as shown in Figure 11. The objective function involves four variables: the thickness of the pressure vessel ( $T_s$ ), the thickness of the head ( $T_h$ ), the inner radius of the vessel ( $R$ ), and the length of the cylindrical component ( $L$ ). Since there are two discrete variables ( $T_s$  and  $T_h$ ) and two continuous variables ( $R$  and  $L$ ), one has a nonlinearly constrained mixed discrete-continuous optimization problem.

The bounds of the design variables are  $0.0625 \leq T_s, T_h \leq 5$  (in constant steps of 0.0625) and  $10 \leq R, L \leq 200$ . The design variables are given in inches and the weight is written as

$$w(T_s, T_h, R, L) = 0.6224T_sT_hRL + 1.7781T_hR^2 + 3.1661T_s^2L + 19.84T_s^2R \quad (82)$$

to be minimized subject to the constraints

$$g_1(T_s, R) = T_s - 0.0193R \geq 0$$

$$g_2(T_h, R) = T_h - 0.00954R \geq 0$$

$$g_3(R, L) = \frac{\pi R^2L + 4/3\pi R^3}{1296000} - 1 \geq 0 \quad (83)$$

$$g_4(L) = -L + 240 \geq 0.$$

The first two constraints establish a lower bound to the ratios  $T_s/R$  and  $T_h/R$ , respectively. The third constraint corresponds to a lower bound for the volume of the vessel and the last one to an upper bound to length of the cylindrical component.

*7.2.1. Comparison of Results of the Second Set of Numerical Experiments.* Table 22 provides the initial guesses  $c_0$  and  $u_0$  used by IED + GA, the size of the population, and the number of generations used by the Genetic Algorithm.

Table 23 presents the best and the worst results, the mean, and the standard deviation. Tables 24–27 show the results found by the algorithm.

TABLE 22: Parameters of the IED method.

Problems	$c_0$	$u_0$	Population Size	Generations
The Tension/Compression String	10	[1 1 1 1]	100	300
The Speed Reducer	1	[1 1 1 1 1 1 1 1 1 1]	100	300
The Welded Beam	10	[1 1 1 1 1]	100	300
The Pressure Vessel	1	[1 1 1 1]	100	300

TABLE 23: Statistical Analysis obtained by IED + GA in the engineering problems.

Engineering problems	IED + GA			
	best	worst	mean	sd
The Tension/Compression String	0.0128	0.0164	0.0144	0.0009
The Speed Reducer	2996.7	3039.7	3015.0	13.7485
The Welded Beam	2.4151	5.4177	3.1995	0.7005
The Pressure Vessel	6477.2	8787.4	7454.1	636.1933

TABLE 24: Comparison for The Tension/Compression String problem.

Var.	Alg-1 [11]	Alg-2 [11]	IED + GA
$x_1$	11.852177	11.329555	11.184056
$x_2$	0.34747463	0.35603234	0.36004951
$x_3$	0.051301897	0.051660806	0.051889349
$g_1$	-0.00000012	-0.000006437	-0.003088921
$g_2$	-0.00000047	-0.000013709	-0.003358194
$g_3$	-4.03513200	-4.052324300	-4.026621039
$g_4$	-0.73414900	-0.728204600	-0.725374090
V	0.012668	0.012666	0.012781

TABLE 25: Comparison for the speed reducer problem.

Var.	Ref. [12]	Ref. [13]	Alg-1 [11]	Alg-2 [11]	IED + GA
$x_1$	3.506163	3.500000	3.500001	3.5	3.499684
$x_2$	0.700831	0.700000	0.700000	0.7	0.700011
$x_3$	17.0	17.0	17.0	17.0	17.290709
$x_4$	7.460181	7.300008	7.300017	7.3000035	7.302923
$x_5$	7.962143	7.715322	7.715326	7.7153225	7.806459
$x_6$	3.362900	3.350215	3.350216	3.3502147	3.350185
$x_7$	5.308949	5.286655	5.286654	5.2866545	5.286355
$g_1$	-0.077734	-0.07391524	-0.07391554	-0.07391524	-0.07386154
$g_2$	-0.201305	-0.19799852	-0.19799876	-0.19799852	-0.19795199
$g_3$	-0.474119	-0.49917084	-0.49916983	-0.49917156	-0.49856110
$g_4$	-0.897068	-0.90464383	-0.90464365	-0.90464383	-0.90120377
$g_5$	-0.011021	-0.00000023	-0.00000155	-0.000000119	0.000030538
$g_6$	-0.012500	-0.00000029	0.00000000	0.00000000	0.000187373
$g_7$	-0.702147	-0.70250000	-0.70250000	-0.702500000	-0.702495228
$g_8$	-0.000573	0.00000000	-0.00000029	0.00000000	0.00010613
$g_9$	-0.583095	-0.58333333	-0.58333320	-0.583333300	-0.583377553
$g_{10}$	-0.069144	-0.051326692	-0.05132753	-0.051326156	-0.051711423
$g_{11}$	-0.027920	-0.00000018	-0.00000077	-0.000000357	-0.011717075
V	3025.0051	2994.4717	2994.4720	2994.4712	2996.7085

TABLE 26: Comparison for the welded beam problem.

Var.	Ref. [14]	Alg-1 [11]	Alg-2 Ref. [11]	IED + GA
$x_1$	0.2442949	0.24432427	0.24438575	0.240091515
$x_2$	6.2116738	6.2201996	6.2183037	6.390631032
$x_3$	8.3015486	8.291464	8.291165	8.314142911
$x_4$	0.2443003	0.24436942	0.24438748	0.246214266
$g_1$	0.0004447	0.000000000	0.001953125	0.004494538
$g_2$	64.378068	0.001953125	0.056640625	0.012900125
$g_3$	0.0000054	0.000045150	0.000001728	0.006122751
$g_4$	0.0002553	0.029785156	1.210937500	0.024765021
$g_5$	0.2342937	0.234240830	0.234240280	0.234486548
$V$	2.38159	2.381246	2.3812175	2.4151069

TABLE 27: Comparison for the pressure vessel problem.

Var.	Ref. [13]	Ref. [14]	Alg-1 [11]	Alg-2 [11]	IED + GA
$T_s$	0.8125	0.8125	0.8125	0.8125	0.90625
$T_h$	0.4375	0.4375	0.4375	0.4375	0.46250
$R$	42.086994	42.0946558	42.093082	42.094967	46.189759
$L$	176.779128	176.684062	176.70308	176.67972	138.552311
$g_1$	0.000221	0.000073	0.0001035	0.000007	0.014787
$g_2$	0.035990	0.035917	0.0359320	0.035914	0.021849
$g_3$	3.219817	2.929000	0.1562500	0.0625	0.035064
$g_4$	63.220872	63.315938	63.296920	63.320282	101.447688
$V$	6061.1229	6060.187934	6060.3677	6060.138	6477.182

## 8. Conclusions

We have proposed a version of the IED method for nonsmooth and nonconvex optimization problems that employs a Genetic Algorithm for minimization the Lagrangian function.

We have solved twenty test problems for comparison between the first version and the current version to verify the effectiveness of the new approach proposed.

We also solved four classical engineering problems to show that the method can be applied to more complex problems.

We have evidence that shows that the more effectively we minimize the Lagrangian the better the IED method will be. Therefore, more powerful methods must be sought for this task. We also believe that the IED method can be bettered by establishing rules to update the matrices  $B^k$  and employing different types of augmented Lagrangians with different matrices  $A$  and different functions  $\sigma$ .

Finally, the hybridization presented in this work combines two algorithms, namely, the IED and a Genetic Algorithm. This approach frees the Genetic Algorithm of handling with constraints which is troublesome for these type of methods and, at the same time, provides IED with a good technique for minimizing the Lagrangian.

## Data Availability

Data are available at [https://www.dropbox.com/sh/5nahzdoy-u37ug7z/AABYgt71fc\\_UB1EsP38xD08fa?dl=0](https://www.dropbox.com/sh/5nahzdoy-u37ug7z/AABYgt71fc_UB1EsP38xD08fa?dl=0) or from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

The authors thank CNPq (Grant 306186/2017-9), FAPEMIG, and FADEPE (Grant 01/17 STRICTO SENSU).

## References

- [1] R. S. Burachik, W. P. Freire, and C. Y. Kaya, "Interior Epigraph Directions method for nonsmooth and nonconvex optimization via generalized augmented Lagrangian duality," *Journal of Global Optimization*, vol. 60, no. 3, pp. 501–529, 2014.
- [2] R. S. Burachik and C. Y. Kaya, "A deflected subgradient method using a general augmented Lagrangian duality with implications on penalty methods," *Springer Optimization and Its Applications*, vol. 47, pp. 109–132, 2010.
- [3] R. S. Burachik and C. Y. Kaya, "An update rule and a convergence result for a penalty function method," *Journal of Industrial and Management Optimization*, vol. 3, no. 2, pp. 381–398, 2007.
- [4] R. S. Burachik, R. N. Gasimov, N. A. Ismayilova, and C. Y. Kaya, "On a modified subgradient algorithm for dual problems via sharp augmented Lagrangian\*," *Journal of Global Optimization*, vol. 34, no. 1, pp. 55–78, 2006.
- [5] R. S. Burachik, A. N. Iusem, and J. G. Melo, "A primal dual modified subgradient algorithm with sharp Lagrangian," *Journal of Global Optimization*, vol. 46, no. 3, pp. 347–361, 2010.

- [6] R. S. Burachik, A. N. Iusem, and J. G. Melo, "Duality and Exact Penalization for General Augmented Lagrangians," *Journal of Optimization Theory and Applications*, vol. 147, no. 1, pp. 125–140, 2010.
- [7] R. S. Burachik, C. Y. Kaya, and M. Mammadov, "An inexact modified subgradient algorithm for nonconvex optimization," *Computational Optimization and Applications*, vol. 45, no. 1, pp. 1–24, 2010.
- [8] R. N. Gasimov, "Augmented Lagrangian duality and nondifferentiable optimization methods in nonconvex programming," *Journal of Global Optimization*, vol. 24, no. 2, pp. 187–203, 2002.
- [9] P. W. Freire, *A Feasible Directions Algorithm for Convex Nondifferentiable Optimization [Ph.D. thesis]*, 2005.
- [10] J. Herskovits, W. P. Freire, M. Tanaka Fo, and A. Canelas, "A feasible directions method for nonsmooth convex optimization," *Structural and Multidisciplinary Optimization*, vol. 44, no. 3, pp. 363–377, 2011.
- [11] S. H. Bernardino, J. C. H. Barbosa, and C. C. A. Lemonge, "Constraint handling in genetic algorithms via artificial immune systems," in *In Late-breaking paper at Genetic and Evolutionary Computation Conference (GECCO 2006)*, 2006.
- [12] E. Mezura-Montes, C. A. C. Coello, and R. Landa-Becerra, "Engineering Optimization Using a Simple Evolutionary Algorithm," in *Proceedings of the 15th IEEE International Conference on Tools with artificial Intelligence*, pp. 149–156, November 2003.
- [13] C. A. C. Coello and N. C. Cortés, "Hybridizing a genetic algorithm with an artificial immune system for global optimization," *Engineering Optimization*, vol. 36, no. 5, pp. 607–634, 2004.
- [14] A. C. C. Lemonge and H. J. C. Barbosa, "An adaptive penalty scheme for genetic algorithms in structural optimization," *International Journal for Numerical Methods in Engineering*, vol. 59, no. 5, pp. 703–736, 2004.
- [15] P. B. Luis, F. F. Camila, P. F. Wilhelm, and J. R. Hernando, "An improvement on the ied method for solving constrained nonsmooth and nonconvex optimization problems," in *Proceedings of the In EngOpt 2016 - 5th International Conference on Engineering Optimization - Proceedings*, pp. 434–443, 2017.
- [16] M. T. Filho, *Algorithms of feasible directions to nonsmooth optimization [Ph.D. thesis]*, 2011, COPPE/UFRJ.
- [17] N. Karmitsa, M. Tanaka Filho, and J. Herskovits, "Globally Convergent Cutting Plane Method for Nonconvex Nonsmooth Minimization," *Journal of Optimization Theory and Applications*, vol. 148, no. 3, pp. 528–549, 2011.
- [18] R. Datta and K. Deb, *Evolutionary Constrained Optimization*, Springer, 2015.
- [19] J. Herskovits, "Feasible direction interior-point technique for nonlinear optimization," *Journal of Optimization Theory and Applications*, vol. 99, no. 1, pp. 121–146, 1998.
- [20] C. Darwin, *On the Origin of Species by Means of Natural Selection*, Murray, London, 1859.
- [21] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, vol. 412, Addison-Wesley Reading, Menlo Park, Calif, USA, 1989.
- [22] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, 1975.
- [23] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 2, pp. 115–148, 1995.
- [24] K. Deb and M. Goyal, "A combined genetic adaptive search (geneas) for engineering design," *Computer Science and Informatics*, vol. 26, pp. 30–45, 1996.
- [25] R. S. Burachik, A. N. Iusem, and J. G. Melo, "An Inexact Modified Subgradient Algorithm for Primal-Dual Problems via Augmented Lagrangians," *Journal of Optimization Theory and Applications*, vol. 157, no. 1, pp. 108–131, 2013.
- [26] W. Hock and K. Schittkowski, *Test examples for nonlinear programming codes*, Lecture Notes in Economics and Mathematical Systems, Springer, New York, NY, USA, 1981.
- [27] V. S. Aragón, S. C. Esquivel, and A. C. C. Carlos, "A modified version of a t-cell algorithm for constrained optimization problems," *International Journal for Numerical Methods in Engineering*, vol. 84, no. 3, pp. 351–378, 2010.

## Research Article

# A Modified Priority-Based Encoding for Design of a Closed-Loop Supply Chain Network Using a Discrete League Championship Algorithm

Javid Ghahremani Nahr,<sup>1</sup> Ramez Kian ,<sup>2,3</sup> and Hassan Rezazadeh<sup>3</sup>

<sup>1</sup>Academic Center for Education, Culture and Research (ACECR), Tabriz, Iran

<sup>2</sup>Nottingham Business School, Nottingham Trent University, UK

<sup>3</sup>Department of Industrial Engineering, University of Tabriz, Tabriz, Iran

Correspondence should be addressed to Ramez Kian; [ramezk@bilkent.edu.tr](mailto:ramezk@bilkent.edu.tr)

Received 6 January 2018; Accepted 11 April 2018; Published 17 May 2018

Academic Editor: Guillermo Cabrera-Guerrero

Copyright © 2018 Javid Ghahremani Nahr et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In a closed-loop supply chain network, the aim is to ensure a smooth flow of materials and attaining the maximum value from returning and end-of-life goods. This paper presents a single-objective deterministic mixed integer linear programming (MILP) model for the closed-loop supply chain (CLSC) network design problem consisting of plants, collection centers, disposal centers, and customer zones. Our model minimizes the total costs comprising fixed opening cost of plants, collection, disposal centers, and transportation costs of products among the nodes. As supply chain network design problems belong to the class of NP-hard problems, a novel league championship algorithm (LCA) with a modified priority-based encoding is applied to find a near-optimal solution. New operators are defined for the LCA to search the discrete space. Numerical comparison of our proposed encoding with the existing approaches in the literature is indicative of the high quality performance of the proposed encoding.

## 1. Introduction

Nowadays, supply chain management (SCM) has received attention in several organizations. SCM is described as the design, production, organization, execution, control, and testing regarding supply chain activities with the goal of creating net value, minimizing the logistics cost, creating a competitive infrastructure, synchronizing demand with supply, leveraging worldwide supply chain, and measuring effectiveness globally. Generally, SCM includes the coordination and integration of key business activities including activities from purchase of raw materials to distribution of the finished products to customers. An efficient and effective supply chain can be regarded as a competitive advantage for companies and plants and helps them to cope with the global market pressure. There are two kinds of supply chain, i.e., forward supply chains and reverse supply chains. The forward supply chain is defined as a set of activities converting raw materials to products as well as storing and distributing

products to the customers, while the reverse supply chain consists of a series of activities such as collection, inspection, repair, recovery, and disposal of used products. Integration of reverse and forward supply chains creates a CLSC. In other words, both forward and reverse supply chain networks are present in the CLSC networks. Network design is one of the most significant strategic decisions in SCM. In general, supply chain network design decisions include determining the number and location of facilities and the quantity of flow between them. In recent years, a few studies have focused on integrated forward and reverse network designs, while this type of integration can prevent the suboptimality and increase the level of network performance and coordination between forward and reverse processes. The present paper proposes a MILP model for design of a CLSC network consisting of plants, collection centers, disposal centers, and customer zones. Furthermore, it addresses two different problems, i.e., the facility location problem and the quantity of flow between facility optimization. A discrete LCA with a

new modified priority-based encoding is applied to solve the proposed model to minimize the total network costs.

## 2. Literature Review

Over recent years, considering the rising importance of reverse and CLSC network designs, numerous articles have been published in this regard. Govindan et al. [1] present a more comprehensive literature review regarding the closed-loop and reverse supply chains. They classify 382 papers published from 2007 to 2013 and propose a more detailed classification based on 10 factors, e.g., the year of release, approaches, objectives, and functions. They assert that almost 50% of the total surveys are linked to the CLSC network design, and almost 40% of them are connected to the reverse supply chain network design (RSCND). Furthermore, the mentioned study reveals that 12% and 88% of the published papers are related to the single-objective and multiobjective models, respectively. Most of the logistics network (RL) design problems (both forward and reverse logistics) include different MILP-based facility location models. These models include vast varieties from simple ones, such as locating facilities with unlimited capacity and lightweight single-piece model, and single product (e.g., [2]) to more complex ones, such as models for limited-capacity multilevel or multiproduct and multiperiod models (e.g., [3]). Jayaraman et al. [4] develop a MILP model for RL network design under a pull system based on customer demand for recycled products aiming to minimize the total cost. Krikke et al. [5] present a MILP for a two-stage RL network of a copier production plant. In their model, both processing returns and inventory costs are considered in the objective function. The uncertainty of the return number and quality of products is an important factor in the design of the RL network. Accordingly, Listes and Dekker [6] present a MILP model for a sand recycling network with the aim of maximizing the total profit. They extend their model to the different conditions under several scenarios. Aras et al. [7] present a nonlinear model to determine the location of collection centers in a simple RL network. The most important thing to note in their paper is the ability of the proposed model in determining optimal purchase price of the used products with the profit-maximizing objective function. In their solution approach, they develop a heuristic method based on Tabu Search (TS) algorithm. Üster et al. [8] develop a semi-integrated network in which there is a forward logistics network where only collection and recycling centers should be located. It optimizes forward and reverse flows simultaneously. They propose an exact approach based on Benders decomposition technique to solve the model. Lu and Bostel [9] consider a two-level location problem with three types of facilities that should be in a special reverse logistics called Reconstruction Location Network. They present a mixed integer binary programming model in which the forward and reverse flow and the interaction between them are considered at the same time. They also develop an algorithm based on Lagrangian heuristic algorithms to solve the model. Wojanowski et al. [10] study the interactions between industries and government agencies in relation to a series of products used by families.

In order to design a CLSC for third-party logistics providers Du and Evans [3] propose an advanced biobjective MILP model. The objective functions of their model are minimizing the lateness and the total cost. They develop a hybrid scatter search method to solve their model. Pishvae et al. [11] propose a linear model for the location of collection and inspection centers in a reverse logistics network and develop a simulated annealing (SA) algorithm to solve the model in large-scale sizes. Bing et al. [12] consider the plastic recycling in the Netherlands. They propose a MILP model and aim to minimize transportation costs and environmental impacts under different scenarios to find the optimal separation strategy. Gomes et al. [13] propose a MILP model to find the optimal location for the collection and sorting centers. Selection is carried out simultaneously under what called "Tactical Network Planning." Their work is inspired by the European Union directive for electrical and electronic waste. Alumur et al. [14] propose a general MILP model with great flexibility that can be used for different recycled products and can be expanded further to include more settings. Their model determine optimal locations and capacities of inspection centers and reproduction plants in the design of RL network. Also, a case study with a RL network design concept for washing machines and dryers (large appliances) was studied in Germany. Their important assumption is that all the components of a product can be used again. Kannan et al. [15] propose a multistage, multiperiod, multiproduct integrated forward/reverse logistics network model for returned products, which is based on GA-based heuristic algorithm. Listes and Dekker [6] present a MILP model and certain dynamic aspects such as due dates and inventory status which results in a complicated model and for this reason they consider a single level single product network and solved it by genetic algorithms. Giri and Sharma [16] solve the problem with developed algorithms for sequential and global optimization. Pati et al. [17] perform a different formulation by presenting a MILP model for a better management of recycled paper logistics system. The authors examine the relationship between multiple targets of recycled paper distribution network such as RL costs, improving product quality and benefits of recycling waste-paper. In addition, their model determines strategic decisions such as locating facilities, as well as tactical decisions, such as recyclable product flows and routing under multi-item, multilevel, and multifacility. Godichaud and Amodeo [18] perform a multiobjective optimization in order to increase control policies with regard to returned products. Fonseca et al. [19] consider uncertainty with the cost of transportation and the waste of production. They present a comprehensive model for reverse logistics planning that consider levels of multiple facilities, multiple products, and selection of different technologies apart from being random. They present a two-stage biobjective mixed integer stochastic programming model for providing strategic and tactical decisions, respectively, in the first and second stages to minimize costs and also the negative impact. Özceylan et al. [20] develop a mixed integer nonlinear programming (MINLP) model, which optimizes the tactical decisions on balancing the decomposition lines in the reverse supply chain and the strategic decisions related to the quantity of

TABLE 1: Papers published in the field of SCND using metaheuristics algorithms.

Author-Date	Model	Network	Encoding	Algorithm
Gen et al. [25]	MILP	SC	Priority-based encoding	Genetic Algorithm
Pishvaei et al. [11]	MILP	RL	Priority-based encoding	Simulated Annealing
Costa et al. [26]	MILP	SC	Matrix encoding	Genetic Algorithm
Jamshidi et al. [27]	MILP	SC	Priority-based encoding	Hybrid Memetic Algorithm
Subramanian et al. [28]	MILP	CLSC	Priority-based encoding	Simulated Annealing
Zohal and Soleimani [29]	MILP	CLSC	Matrix encoding	Ant Colony Algorithm
Zandieh and Chensebli [30]	MILP	RL	Priority-based encoding	Water Flow-like Algorithm
Singh et al. [31]	MILP	RL	Matrix encoding	Genetic Algorithm
Pasandideh and Asadi [32]	MILP	SC	Priority-based encoding	Imperialist Competitive Algorithm
<i>This Paper</i>	<i>MILP</i>	<i>CLSC</i>	<i>A new Priority-based encoding</i>	<i>Discrete League Championship Algorithm</i>

products flowing on the CLSC. Soleimani et al. [21] propose a stochastic integer linear programming model for designing a multiproduct CLSC in the case of uncertainties regarding the demand, purchase price, and rate of return. Some of the most crucial studies addressing the supply chain are based on a metaheuristic algorithm (Table 1). However, recently other researchers have investigated this problem meticulously (e.g., [22, 23]). Table 1 summarizes similar works and highlights our contribution in this study.

### 3. Contribution Highlights

We contribute to the literature in three dimensions: (i) to the best of our knowledge, this work is the first League Champion Algorithm in which continuous encoded solutions are converted to discrete encoded solutions based on the new operators defined in this paper. (ii) We propose an efficient encoding methodology in our algorithm which keeps the generated solutions in the feasible region which causes speeding up of the algorithm and rising the chance of finding optimal solution. (iii) The numerical experiments demonstrate that our algorithm outperform similar existing ones in the literature and capable of solving large-scale problems in reasonable execution time which are not solvable via the commercial solver in reasonable time.

### 4. Problem Definition and Modeling

In the current study, a general CLSC network is considered. The forward network includes plants and customer zones, and the reverse network includes collection and disposal centers. According to Figure 1, the plants could manufacture new products and remanufacture the returned ones. The plants send products to the customer zones. Then, the returned products from customers are collected by the collection centers, and after inspection of products, the repairable products are sent to the plants and the remaining ones are sent to the disposal center. To determine the scope of the study, the following assumptions are made for the proposed model.

- (i) The model is designed for a single period.
- (ii) All facilities have limited and identified capacities.

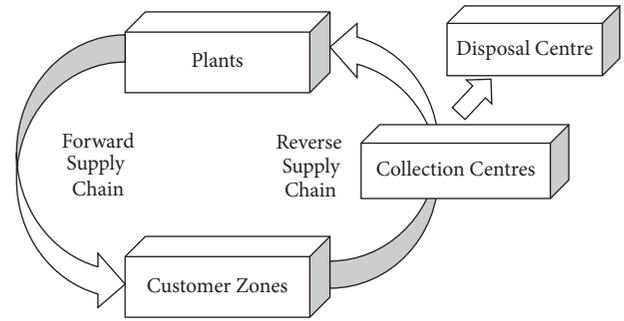


FIGURE 1: The closed-loop supply chain network Amin and Zhang [24].

- (iii) The locations of all centers are potential and unknown.
- (iv) All customers demands must be satisfied and all the returned products from customers must be collected.
- (v) Customers locations are fixed and predefined.

Given the above assumptions, the most important issue mentioned in this paper is locating the plants, collection centers, and disposal centers as well as determining the optimal amount of flow between centers. The network can be formulated as a MILP model. Sets, parameters, and decision variables are defined as follows:

#### Sets

- $I$ : set of potential plants, whose elements are addressed by index  $i \in I$
- $J$ : set of fixed locations of customer zones, whose elements are addressed by index  $j \in J$
- $K$ : set of potential collection centers, whose elements are addressed by index  $k \in K$
- $L$ : set of potential disposal centers, whose elements are addressed by index  $l \in L$

#### Parameters

- $TPS_{ij}$ : unit transportation cost between plant  $i$  and customer zone  $j$

$TSC_{jk}$ : unit transportation cost between customer zone  $j$  and collection center  $k$

$TCP_{ki}$ : unit transportation cost between collection center  $k$  and plant  $i$

$TCD_{kl}$ : unit transportation cost between collection center  $k$  and disposal center  $l$

$E_i$ : fixed cost for opening plant  $i$

$F_k$ : fixed cost for opening collection center  $k$

$G_l$ : fixed cost for opening disposal center  $l$

$CapP_i$ : capacity of plant  $i$

$CapC_k$ : capacity of collection center  $k$

$CapD_l$ : capacity of disposal center  $l$

$Dem_j$ : demand of customer  $j$

$r_j$ : return of customer  $j$

$\beta$ : minimum disposal fraction

#### Decision Variables

$X_{ij}$ : quantity of new products shipped from plant  $i$  to customer zone  $j$

$Y_{jk}$ : quantity of returned products from customer  $j$  to collection center  $k$

$S_{ki}$ : quantity of returned products from collection center  $k$  to plant  $i$

$T_{kl}$ : quantity of returned products from collection center  $k$  to disposal center  $l$

$Z_i$ : 1, if a plant is located and set up at potential site  $i$ , and 0, otherwise

$W_k$ : 1, if a collection center is located and set up at potential site  $k$ , and 0, otherwise

$H_l$ : 1, if a disposal center is located and set up at potential site  $l$ , and 0, otherwise.

**4.1. Model Formulation.** The mathematical model of the problem can be presented as follows:

$$\begin{aligned} \text{Minimize: } & \sum_i E_i Z_i + \sum_k F_k W_k + \sum_l G_l H_l \\ & + \sum_i \sum_j TPS_{ij} X_{ij} + \sum_j \sum_k TSC_{jk} Y_{jk} \\ & + \sum_k \sum_i TCP_{ki} S_{ki} + \sum_k \sum_l TCD_{kl} T_{kl}, \end{aligned} \quad (1)$$

$$\text{s.t. } \sum_i X_{ij} \geq Dem_j, \quad \forall j \in J, \quad (2)$$

$$\sum_i X_{ij} \geq \sum_k Y_{jk}, \quad \forall j \in J, \quad (3)$$

$$\sum_k Y_{jk} = r_j, \quad \forall j \in J, \quad (4)$$

$$\beta \sum_j Y_{jk} = \sum_l T_{kl}, \quad \forall k \in K, \quad (5)$$

$$\sum_j Y_{jk} = \sum_i S_{ki} + \sum_l T_{kl}, \quad \forall k \in K, \quad (6)$$

$$\sum_j X_{ij} + \sum_k S_{ki} \leq Z_i CapP_i, \quad \forall i \in I, \quad (7)$$

$$\sum_j Y_{jk} \leq W_k CapC_k, \quad \forall k \in K, \quad (8)$$

$$\sum_k T_{kl} \leq H_l CapD_l, \quad \forall l \in L, \quad (9)$$

$$Z_i, W_k, H_l \in \{0, 1\}, \quad \forall i \in I, k \in K, l \in L, \quad (10)$$

$$X_{ij}, Y_{jk}, S_{ki}, T_{kl} \geq 0, \quad \forall i \in I, j \in J, k \in K, l \in L. \quad (11)$$

The first term in the objective function (1) represents the fixed costs of locating the plants. The second and third terms indicate the fixed costs of locating the collection centers and disposal centers, respectively. The fourth term corresponds to the production and transportation costs of new products. The fifth term represents the collection processing and transportation costs of returned products from customers. The sixth term calculates the recovery processing and transportation costs of returned products from collection centers to plants. The seventh term calculates the disposal processing and transportation costs of returned products from collection centers to disposal centers. Constraint (2) ensures that all customers demands are satisfied. Constraint (3) guarantees that forward flow is greater than reverse one. Constraint (4) computes the returned products from each customer. Constraint (5) enforces a minimum disposal fraction for each product. Constraint (6) indicates that the quantity of returned products from customer zones to collection centers is equal to the quantity of returned products from collection centers to plants and quantity of returned products from collection centers to disposal centers. Constraint (7) is a capacity constraint of plants. Constraint (8) is a capacity constraint for collection centers. Constraint (9) is a capacity constraint of disposal centers. Constraint (10) and (11) enforce the nonnegativity and binary restrictions on decision variables. The NP-hardness of supply chain network design problem has been proved by many research studies (e.g., [4]). The considered model in this paper consists of two different problems, i.e., facility location problem and quantity of flow between facility optimization; therefore, the developed model is reducible

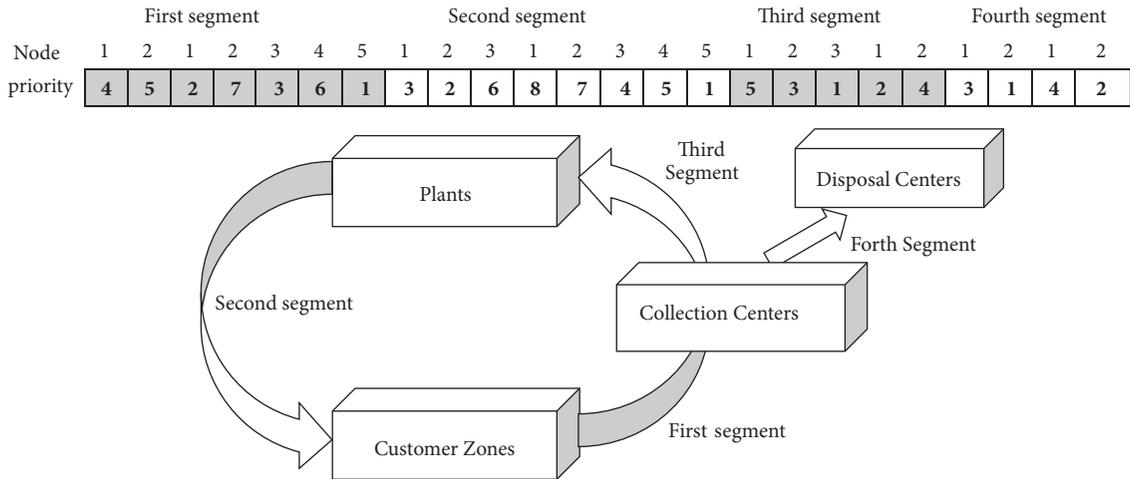


FIGURE 2: An illustration of the CLSC network and its representation.

to facility location problem. Davis and Ray [33] concluded that facility location problem is NP-complete. Hence, the discussed CLSC network design problem is considered as NP-Hard in this paper. Solving this problem by exact solutions is time-consuming and sometimes impractical in large scales. Therefore, several metaheuristic algorithms with different approaches have been developed to get near optimal solutions, though all are not efficient. In the this paper, a discrete LCA solution approach is proposed and applied based on modified priority-based encoding.

### 5. Solution Approach

In this section, we first discuss about the encoded solution format which is referred to as “representation” in the rest of this manuscript and describe the discrete LCA used for solving the CLSC network design problem will be explained.

*5.1. Representation.* Representation is one of the most essential issues for encoding and decoding, which affects the performance of algorithms. Tree-based representation is one way for representing network problems. There are several ways to encode the tree; for example, Gen and Cheng [34] introduce three ways of encoding the tree, i.e., edge-based encoding, vertex-based encoding, and edge-vertex encoding. Michalewicz et al. [35] use matrix-encoding representation, which belongs to edge-based encoding for solving the transportation tree. They present solutions through a  $|K| \times |J|$  matrix in their approach, where  $|K|$  and  $|J|$  are the number of sources and depots, respectively. This method requires a large memory space on the computer. Another representation of transportation tree belonging to the vertex-based encoding is the Prüfer number, which is developed by Gen and Cheng [34] wherein the solution is presented through  $|K| + |J| - 2$  digits. Their method needs some repair mechanisms to obtain a feasible solution. Gen and Cheng [34] develop a priority-based encoding for solving the transportation tree, which does not need any excessive repair mechanisms and the solution is presented through  $|K| + |J|$  digits. Moreover,

this representation is applied to the shortest path problem and projects the scheduling problem. In the present paper, a modified priority-based encoding representation of Gen and Cheng [34] is proposed. In contrast to the other developed representations, presented in the literature, the suggested representation can solve the facility location and quantity flow optimization problems together. The representation includes four segments of sizes  $|K| + |J|$ ,  $|I| + |J|$ ,  $|I| + |K|$ , and  $|K| + |L|$  digits, respectively, wherein the position of each digit represents the sources and depots within the supply chain network.

The first segment is devoted to the customers and collection centers, the second segment is devoted to the plants and customers, the third segment is dedicated to the collection centers and plants, and finally the fourth segment is related to the collection and disposal centers. For example, the CLSC network depicted in Figure 2 includes five customers, two collection centers, three plants, and two disposal centers with the corresponding representations.

To demonstrate the effectiveness of the proposed representation in solving the supply chain problem, Figure 3 depicts the difference between the priority-based encoding developed by Gen et al. [25], priority-based encoding modified by Jamshidi et al. [27], and priority-based encoding proposed in this paper, which are denoted by (PB1), (PB2), and (PB3), respectively. This figure illustrates a two-level supply chain with 3 sources and 4 depots, their corresponding capacities, depots demand, the opening cost of sources, transportation cost between the nodes, and the priority-based encoding. In the proposed representation, the solution is presented through  $|K| + |J|$  digits, where the position of each digit represents the sources and depots within the supply chain network. Furthermore, the value in digits denotes the priorities. After the encoding operation, in order to create a connection between the representation and the supply chain network, the decoding should be performed in a specific manner in which the output denotes the facilities opened and transportation amount between opened centers, as well. To decode the representation, first the potential facilities is

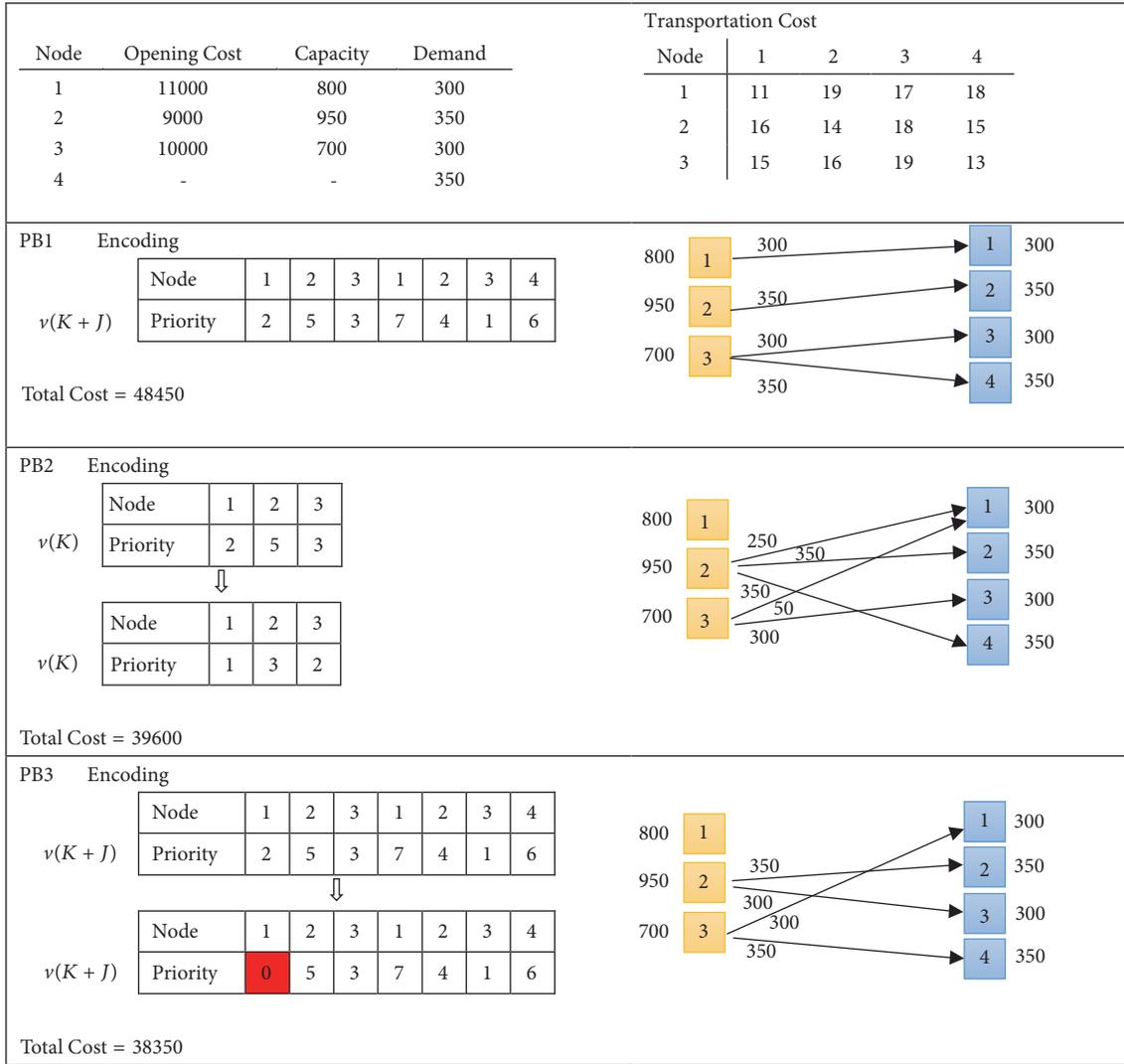


FIGURE 3: Samples of two-level supply chain network and its encoding.

located; then, the optimal shipment size among the located centers is determined. For example, to decode a two-level supply chain network given in Figure 3, the following steps are taken into account:

Let  $Tr_{ij}$  denote the transportation cost from source  $i$  to depot  $j$  and let  $v(\cdot)$  denote the priority.

In the first step of the first section of decoding procedure:

- (i) The source with the highest priority (source 2) is selected; then, the capacity of this source is compared with the total demand of depots. If the capacity of the selected source is less than total demand ( $950 < 1300$ ), then the source with the next highest priority (source 3) is selected. Continue this procedure until the total capacity of sources is less than the total demand of depots. Then, reduce the priorities of the nodes (sources) which are not selected to zero; i.e.,  $v(1) = 0$ , and set the transportation cost from them to infinity,  $Tr_{1j} = \infty, \forall j \in J$ .

In the next step,

- (i) the depot with the highest priority (depot 1) is assigned to the selected sources (source 3) as they have the lowest transportation cost among other pairs;
- (ii) among the selected nodes, determine the shipment size; here, it is equal to  $X_{31} = \min(700, 300) = 300$ ;
- (iii) Update the capacity and the demand of the selected source and depot as  $Cap_3 = 700 - 400 = 300$  and  $Dem_1 = 300 - 300 = 0$ ;
- (iv) As the demand of depot (1) is zero, its priority must be reduced to zero; i.e.,  $v(1) = 0$ .

In the next step,

- (i) depot (4) with the highest priority after updating is connected to source (3), and shipment size between them is determined. At the end, capacity values of

TABLE 2: Trace table of decoding procedure Iteration.

Iteration	$v(k)$	Cap	Cap $T$	$\sum_{j=1}^I \text{Dem}_j$	$k$	$v(k^*)$
0	[2 5 3]	(800, 950, 700)	950	1300	2	[0 5 0]
1	[2 0 3]	(800, 0, 700)	1650	1300	3	[0 5 3]
Iteration	$v(k^* + j)$	Cap	Dem	$K^*$	$j$	$X_{k^*j}$
0	[0 5 3   7 4 1 6]	(950, 700)	(300, 350, 300, 350)	3	1	300
1	[0 5 3   0 4 1 6]	(950, 400)	(0, 350, 300, 350)	3	4	350
2	[0 5 3   0 4 1 0]	(950, 50)	(0, 350, 300, 0)	2	2	350
3	[0 0 3   0 4 1 0]	(600, 50)	(0, 0, 300, 0)	2	2	-
4	[0 0 3   0 0 1 0]	(600, 50)	(0, 0, 300, 0)	3	2	-
5	[0 0 0   0 0 1 0]	(600, 50)	(0, 0, 300, 0)	2	3	300
6	[0 0 0   0 0 0 0]	(300, 50)	(0, 0, 0, 0)			

source (3) and the demand of depot (4) are updated. This sequence of operations is repeated until all demand of depots is satisfied and all priorities are reduce to zero. Table 2 presents the trace table for the example two-level supply chain network given in Figure 3 to show how its modified priority-based encoding is obtained. In this table, column  $v(k)$  denotes the priority of the source nodes while  $v(k^*)$  denotes the priority values of the sources which will serve and set the others to zero. The capacity of the sources, the total capacity of the selected nodes, and demand vectors are given in columns Cap, Cap $T$ , and Dem, respectively. Column  $v(k^* + j)$  gives the representation and column  $K^*$  nominate the selected source whereas column  $X_{k^*j}$  shows the flow amount from the source  $k^*$  to the demand node  $j$ .

The decoding algorithm for representation of a two-level supply chain network is indicated in Algorithm 1. The representation should be divided to four segments to apply the above-presented decoding algorithm to the discussed CLSC network design.

Decoding of the second segment of the representation is impossible until the first segment is decoded because customers demands consist of new products and recovered products, and the amount of recovered products is achieved only by decoding the first segment. Then, decoding of the third and fourth segments starts after determining the number and location of collection centers and the amount of the returned products. Decoding algorithm of representation for multilevel CLSC network is provided in Algorithm 2.

**5.2. Discrete League Championship Algorithm.** The LCA, first presented by Kashan [36], is a population-based algorithm for global search in a continuous space, which is inspired by the championship process of sports leagues in the real world. One of the common characteristics shared by all population-based algorithms such as LCA is their attempt to move a population of possible solutions to a number of promising areas in search of a desirable solution. Similar to many population-based algorithms, a set of  $\ell$  solutions from the search space is randomly selected, which constitutes the initial population in the algorithm. Each solution obtained

from the population is related to one of  $\ell$  teams ( $\ell$  is an even number) that shows the current formation of the team. Hence, team  $i$  represents  $i$ th member of the population. Each solution obtained from the population has its own fitness value. In this algorithm, different solutions can be provided for a problem, which are compared on the basis of their fitness values (their objective function values). Each of the solutions is improved and ultimately near-optimal solution is selected. A number of teams (examined random initial solutions) compete together in a pairwise form as a league within a few weeks (the number of evaluation steps in an iterative algorithm). Winner and loser teams are determined (draw is not allowed) based on their power of play (metaphor of the fitness or objective value of the solution) resulting from the team formation. Every week, each team forms a new team formation by its coach with an artificial process analyzing the last week's matches and obtains the best recognized formation up to that time (iterations number of the algorithm), and this process will continue. A schematic overview of our algorithm is given in Figure 4.

Parameters  $S$  (examined number of seasons) and  $\ell$  (the number of teams) and constant coefficients used to scale the contribution of the strength and weakness of components (denoted by  $\psi_1$  and  $\psi_2$ ), are adjustable parameters whose changes have a direct impact on the final answer of the algorithm. The solution space in the CLSC problem is discrete, which means that components of each individual solution within the population cannot get an arbitrary amount, and they are allowed only to get natural values. Hence, a discrete LCA is dealt with to solve the CLSC problem. In addition, the numbers in representations should not be repeated. In the conducted studies dealing with discrete league championship algorithm, the algorithm searches a continuous space and finally presents a discrete solution by an innovative method. However, in the procedure proposed in the present study, all stages of the algorithm are made in the discrete space with no changes in the presentation of solution.

Hence, according to the discrete structure of the problem and the definition of team formation, discrete version of the operators (including addition, multiplication, subtraction, and arranging) is defined to address the strategies used in LCA in an appropriate structure to solve the CLSC problem.

**Require:** Gets  $\bar{K}$ : Set of sources,  $\bar{J}$ : Set of depots,  $Dem_j$ : demand on depot  $j$ ,  $Cap_k$ : capacity of source  $k$   
**Ensure:** gives  $\bar{U}_{kj}$ : Quantity of shipment between source  $k$  and depot  $j$

- (1)  $CapT := 0, K' = K, K'' = \emptyset$
- (2) **while**  $CapT < \sum_{j=1}^J Dem_j$  **do**
- (3)   select a node on  $n = \arg \max\{v(k), k \in K'\}$
- (4)    $K' = K' \setminus \{n\}$
- (5)    $K'' = K'' \cup \{n\}$
- (6)    $CapT = \sum_{n \in K'} Cap_n$
- (7) **end while**
- (8)  $v(n) = 0, \forall n \in K'$
- (9)  $Tr_{nj} = \infty, \forall j \in \bar{J}, n \in K'$
- (10)  $X_{kj} = 0, \forall j \in \bar{J}, k \in K$
- (11) **while**  $v(n) \neq 0, \forall n \in J \cup \bar{K}$  **do**
- (12)   Select a node based on  $n = \operatorname{argmax}\{v(t), t \in \bar{K} + \bar{J}\}$
- (13)   **if**  $l \in k$  a source is selected  $k^* = n$  **then**
- (14)      $j^* = \arg \min\{Tr_{kj}\}, j \in \bar{J}$  select a depot with minimum transportation cost
- (15)   **elseif**  $n \in J$  a depot is selected  $j^* = n$
- (16)      $k^* = \arg \min\{Tr_{kj} \mid k \in K''\}$  select a source with minimum transportation cost
- (17)   **end if**
- (18)    $\bar{U}_{k^*j^*} = \min(Cap_{k^*}, Dem_{j^*})$
- (19) Update demand and capacities:
- (20)    $Cap_{k^*} = Cap_{k^*} - \bar{U}_{k^*j^*}$
- (21)    $Dem_{j^*} = Dem_{j^*} - \bar{U}_{k^*j^*}$
- (22)   **If**  $Cap_{k^*} = 0$  **then**  $v(k^*) = 0$
- (23)   **If**  $Dem_{j^*} = 0$  **then**  $v(j^*) = 0$
- (24) **end while**

ALGORITHM 1: The decoding algorithm of each section of representation for two-level supply chain network.

**Requires:** problem parameters  
**Ensure:** Calculating the objective function in the multi-level CLSC

- (1)  $Z_i = 0, W_k = 0; H_l = 0, X_{ij} = 0, Y_{jk} = 0, S_{ki} = 0, T_{kl} = 0, \forall i \in I, j \in J, k \in K, l \in L$
- (2) Calculate  $Y_{jk} := \bar{U}_{jk}, \forall j \in J, k \in K$  by Algorithm 1
- (3)   **If**  $\sum_j Y_{jk} > 0$  **then**  $W_k = 1$
- (4) Calculate  $T_{kl} := \bar{U}_{kl}, \forall k \in K, l \in L$  by Algorithm 1
- (5)   **If**  $\sum_k T_{kl} > 0$  **then**  $H_l = 1$
- (6) Calculate  $X_{ij} := \bar{U}_{ij}, \forall i \in I, j \in J$  by Algorithm 1
- (7)   **If**  $\sum_j X_{ij} > 0$  **then**  $Z_i = 1$
- (8) Calculate  $S_{ki}, \forall k \in K, i \in I$  by Algorithm 1
- (9) Calculate the total cost:
- (10) obj =  $\sum_{i=1}^I E_i Z_i + \sum_{k=1}^K f_k W_k + \sum_{l=1}^L G_l H_l + \sum_{i=1}^I \sum_{j=1}^J TPS_{ij} X_{ij}$
- (11)   +  $\sum_{j=1}^J \sum_{k=1}^K TSC_{jk} X_{jk} + \sum_{k=1}^K \sum_{i=1}^I TCP_{ki} X_{ki} + \sum_{k=1}^K \sum_{l=1}^L TCD_{kl} X_{kl}$

ALGORITHM 2: The decoding algorithm of representation for multilevel CLSC network.

The mentioned operators and their corresponding arguments are summarized below:

- (1) Arrange (formation, strategy)  $\xrightarrow{\text{Plus}}$  Formation
- (2) Subtraction (formation, formation)  $\xrightarrow{\text{Minus}}$  Strategy
- (3) Addition (strategy, strategy)  $\xrightarrow{\text{Plus}}$  Strategy
- (4) Multiplication (real\_number, strategy)  $\xrightarrow{\text{Times}}$  Strategy

The strategy vector and the above mentioned operators will be mathematically defined.

**5.2.1. Strategy Vector and Organization Operator (Arrange).**  
The strategy vector, denoted by  $S$ , should be defined in such a way that by applying to a team formation vector at any time step, another team formation is achieved. Thus, the strategy vector  $S$  for each team is defined as a sequence of “displacements” of team formation components as follows:

$$S = ((i_1, j_1), (i_2, j_2), \dots, (i_k, j_k), \dots, (i_{\|S\|}, j_{\|S\|})), \quad (12)$$

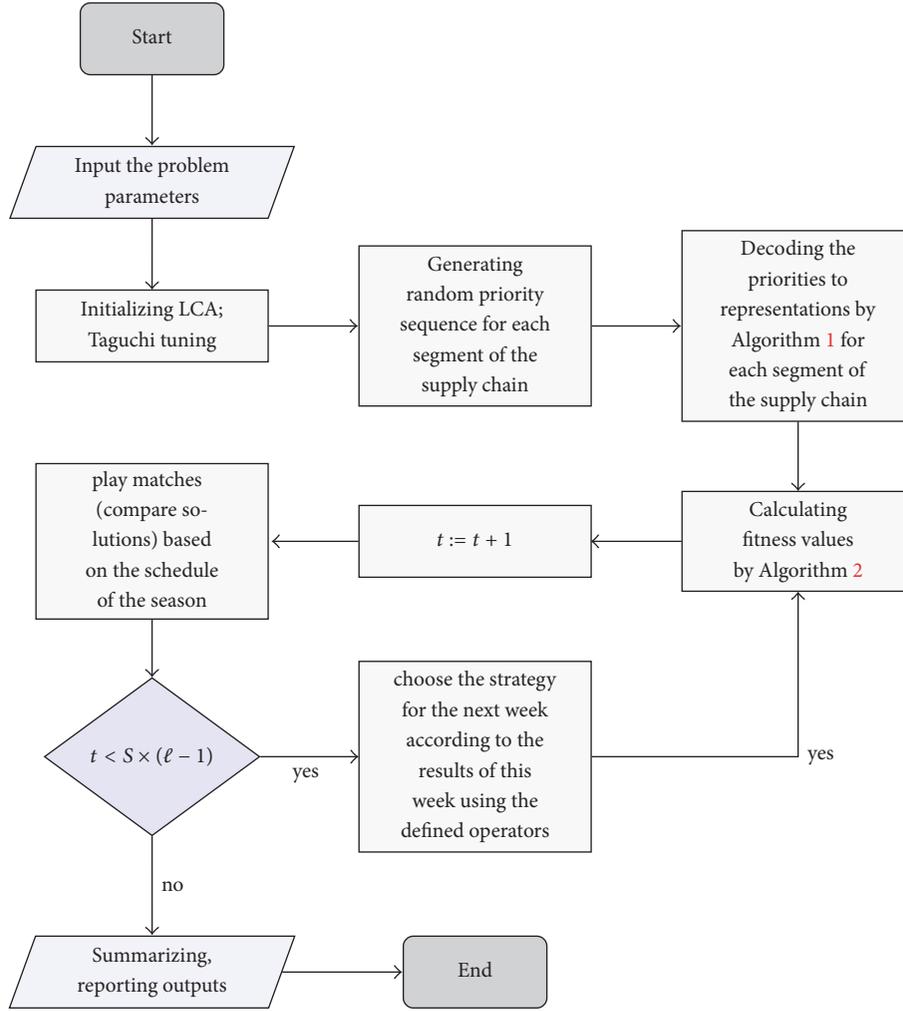


FIGURE 4: The overview of the LCA algorithm implementation.

where  $\|S\|$  specifies the length of sequence. Applying strategy  $S$  to a vector of team formation means that the components of  $i_1$  and  $j_1$  in the vector of team formation are “displaced” together and then their components  $i_2$  and  $j_2$  and  $i_3$  and  $j_3$  are “displaced” together up to  $i_{\|S\|}$  and  $j_{\|S\|}$ . Hence, organization operator, i.e., the sum of team formation with a strategy, is defined as a set of displacements specified by the strategy vector in the vector of team formation. For example, if  $F = (1, 2, 3, 4, 5)$  is the vector of team formation and  $S = ((1, 2), (2, 3))$  is the vector of strategy, then the following situations are achieved, respectively, by applying  $S$  to  $F$ :

Displacement (1) with (2): (2, 1, 3, 4, 5)

Displacement (2) with (3): (3, 1, 2, 4, 5)

Hence, arrange  $(F, S) = F + S = (3, 1, 2, 4, 5)$

Symmetry of a strategy vector  $S$  is displayed as  $-S$ , which means that the sequence of displacements in  $S$  is considered reversely.

**5.2.2. Subtraction Operator.** If  $x_1$  and  $x_2$  are two vectors of a team formation, the difference between  $x_1$  and  $x_2$  is defined

in a way that  $x_2 - x_1$  is equal to the vector of strategy,  $S$ . If it is applied to  $x_1$ ,  $x_2$  is achieved. The algorithm which calculates the difference between  $x_1$  and  $x_2$  should be chosen with consideration of

$$\begin{aligned} x_2 - x_1 &= -(x_2 - x_1) \\ x_1 = x_2 &\longrightarrow x_2 - x_1 = \emptyset \end{aligned} \tag{13}$$

Value of zero is displayed with  $\emptyset$  and is equal to a null sequence.

**5.2.3. Addition Operator.** If  $S_1$  and  $S_2$  are two strategy vectors, sum of them is displayed as  $S_1 \oplus S_2$  and equals to the strategy vector which is achieved by connecting the displacement sequence of  $S_2$  to the end of sequence in  $S_1$ . As some displacements cancel each others effects, the obtained sequence from connecting  $S_2$  to  $S_1$  can be smaller; for example, if  $S_1 = ((1, 2), (2, 5))$  and  $S_2 = (2, 5)$ , then we have  $S_1 \oplus S_2 = ((1, 2), (2, 5), (2, 5))$ .

However, since two consecutive displacements (2, 5) cancel each others effects, they can be deleted; hence,  $S_1 \oplus S_2 =$

(1, 2). Therefore, in general, we have  $\|S_1 \oplus S_2\| \leq \|S_1\| + \|S_2\|$ . In addition, for each strategy  $S$ , we will have  $S \oplus -S = \emptyset$ .

**5.2.4. Multiplication Operator.** If  $c$  is a real number and  $S$  is a strategy vector, their multiplication,  $cS$ , is defined as follows depending on the value of  $c$ .

(1) Case of  $0 \leq c < 1$ : in this case, it is assumed that  $\|cS\|$  is equal to the integral part of the number  $c\|S\|$ ; vector  $S$  is truncated so that its length is equal to  $\|cS\|$ :

$$cS = ((i_k, j_k), k = (1, 2, \dots, \|cS\|)). \quad (14)$$

In the special case of  $c = 0$ , we have  $cS = \emptyset$ .

(2) Case of  $c \geq 1$ : in this case, we have  $c = k + c'$ , where  $k = \lfloor c \rfloor$  is a natural number and  $c' = c - \lfloor c \rfloor \in [0, 1]$ . Therefore,  $cS$  is defined as follows:

$$cS = \underbrace{S \oplus S \dots S}_{k \text{ times}} \oplus c'S. \quad (15)$$

(3) Case of  $c < 0$ : in this case, the multiplication is converted to one of the above-presented modes by writing  $cS$  as  $(-c)(-S)$ .

We have four strategies whose notations are inspired from SWOT (Strength, Weakness, Opportunity, Threats) analysis in strategic planning literature.  $S/O$  corresponds to the aggressive strategy in which the team is strong and the opponent is weak while  $W/T$ , in contrast, corresponds to a defensive strategy in which the team is in a weak mode and the opponent is strong (threat). Two other intermediate strategies are denoted by  $S/T$  and  $W/O$  corresponding to *both strong* and *both weak* modes, respectively. Let  $X_i^{t+1}$  denote the formation of team  $i$  in week  $t$  and  $B_i^t$  be the best formation for team  $i$  in week  $t$ . The constant coefficients  $\psi_1$  and  $\psi_2$  are used to scale the strength and weakness of components, respectively.  $r_1$  and  $r_2$  are uniform random numbers from the interval  $[0, 1]$ .

With the application of the defined operators, equations related to strategies in discrete LCA algorithm will be written as follows.

- (i) If both teams  $i$  and  $l$  in week  $t$  win their games against teams  $j$  and  $k$ , then formation of team  $i$  in week  $t + 1$  with strategy  $S/T$  will be as follows:

$$X_i^{t+1} = B_i^t + (\psi_1 r_1 (B_i^t - B_k^t) \oplus \psi_1 r_2 (B_i^t - B_j^t)). \quad (16)$$

- (ii) If team  $i$  wins team  $j$  and team  $l$  loses against team  $k$  in week  $t$ , team formation of  $i$  in week  $t + 1$  with strategy  $S/O$  will be as follows:

$$X_i^{t+1} = B_i^t + (\psi_2 r_1 (B_k^t - B_i^t) \oplus \psi_1 r_2 (B_i^t - B_j^t)). \quad (17)$$

- (iii) If both teams  $i$  and  $l$  lose their games against teams  $j$  and  $k$  in week  $t$ , team formation of  $i$  in week  $t + 1$  with strategy  $W/T$  will be as follows:

$$X_i^{t+1} = B_i^t + (\psi_1 r_1 (B_k^t - B_i^t) \oplus \psi_2 r_2 (B_i^t - B_j^t)). \quad (18)$$

- (iv) If both teams  $i$  and  $l$  lose their games against teams  $j$  and  $k$  in week  $t$ , team formation  $i$  in week  $t + 1$  with strategy  $W/O$  will be as follows:

$$X_i^{t+1} = B_i^t + (\psi_2 r_1 (B_k^t - B_i^t) \oplus \psi_2 r_2 (B_j^t - B_i^t)). \quad (19)$$

## 6. Numerical Results

**6.1. Problem Instances.** In order to assess the performance of the proposed discrete LCA in terms of the objective-function value and CPU time, several numerical experiments with different problem sizes were implemented, and the obtained results are reported in this section. To this end, 21 sample problems with 3 different levels, which are small, medium, and large-scale problems, were generated by different combinations of the parameter values. The levels of the generated sample problems are shown in Table 3, and the ranges of the parameters are presented in Table 4. Furthermore, all parameters of the sample problems were randomly generated based on uniform distributions in prespecified intervals.

As the acquired results obtained from the proposed algorithm are sensitive to their initial parameters, any small changes can affect the accuracy of the best obtained solution. Therefore, the Taguchi tuning method is used for the parameters to find the best solutions. In this method, first, the appropriate factors (initial parameters) are determined, the level of each factor is selected, and then the design of experiments for this control factor is specified. After specifying the experimental design, the proposed algorithm is used to find the best combination of factors. In our study, 4 levels are considered for each factor. In Table 5, the number of teams ( $\ell$ ), number of seasons ( $S$ ), and scaling constants of strength and weakness components ( $\psi_1$  and  $\psi_2$ ) are given.

**6.2. Objective Function Average.** With respect to the proposed algorithm, the experimental design is employed according to the number of factors and number of levels. To this aim, each problem instance was solved five times to form the replications. Average results are reported as the final value. The best values of the proposed parameters for discrete LCA according to the mean normalized objective values are 150, 20, 6, and 2 for the number of teams, number of seasons, and constant coefficients to scale the contribution of the strength and weakness components, respectively. Moreover, Figure 5 presents the normalized average of means and the  $S/N$  (Signal-to-noise) ratio graph for the experimental design of the algorithm. Based on the average of the means given in the graph, the algorithms result in a more efficient response if the parameter is placed in the lower level. Furthermore, in the  $S/N$  ratio graph, the algorithms result in a more efficient response if the parameter is placed in the higher level. According to Table 5 and Figure 5 the best performance of our algorithm is observed when the number of teams ( $\ell$ ) and the weakness scaling coefficient ( $\psi_2$ ) are in level 2; the number of seasons ( $S$ ) is placed at level 1, and the strength scaling coefficient ( $\psi_1$ ) is placed at level 4.

**6.3. Solutions Structure and Quality.** After tuning the parameters of the proposed algorithm, PB3 results are compared

TABLE 3: Size and scale of sample problems.

Scale of the problem	Problem no.	# the potential locations for plants	# the potential locations for collection centers	# the potential locations for disposal centers	# the potential locations for customer zones
Small	1	3	2	2	10
	2	4	3	3	15
	3	5	4	3	20
	4	7	5	4	25
	5	8	6	5	30
	6	10	8	7	35
	7	12	8	7	40
Medium	8	15	9	8	45
	9	18	10	9	50
	10	21	12	10	55
	11	25	15	12	60
	12	30	20	15	70
	13	35	25	18	80
	14	40	32	20	90
Large	15	45	37	24	100
	16	50	42	29	120
	17	60	48	33	140
	18	70	53	38	160
	19	80	57	42	180
	20	90	60	45	200
	21	100	65	50	220

TABLE 4: Pre-specified intervals to generate parameters intervals based on uniform distributions.

Parameter	Range	Parameter	Range
$E_i$	$U(1000000, 1200000)$	$TSC_{jk}$	$U(20, 30)$
$F_k$	$U(1000000, 1200000)$	$TCP_{ki}$	$U(20, 30)$
$G_l$	$U(1000000, 1200000)$	$TCD_{kl}$	$U(20, 30)$
$Dem_j$	$U(100, 150)$	$CapP_i$	$U(800, 1200)$
$r_j$	$U(10, 50)$	$CapC_k$	$U(200, 400)$
$\beta$	$U(0.6, 0.8)$	$CapD_l$	$U(100, 300)$
		$TPS_{ij}$	$U(20, 30)$

TABLE 5: Proposed parameter levels.

Parameter	Level 1	Level 2	Level 3	Level 4
$\ell$	100	150	200	25
$S$	20	26	32	40
$\psi_1$	0	2	4	6
$\psi_2$	0	2	4	6

with those of the CPLEX outputs which were run on GAMS software (exact solution), PB2, and PB1 methods. For this purpose, five problem instances with random data given in Table 4 for each problem size given in Table 3 were generated, and they were solved by each method. The average of five problems were selected as the benchmark for each sample problem. In Table 6, the average results of each sample

problem instance is provided and compared with the solver's (exact) solution.

According to the results shown in Table 6, it can be observed that the results of PB3 method are very close to the exact solution with a small deviation. The average deviations from the optimality for these algorithms are depicted in Figure 6, the solution obtained from PB3 algorithm for samples (1) to (4) are the same as the exact solution, as their deviations are zero. Furthermore, in comparison with the PB2 and PB3 methods, PB3 method resulted in smaller deviation values as the size of the problem increases. The optimality deviation in problem instance #14 is 760.9 (0.0024%) for PB3 algorithm while it is 1856631.6 (5.773%) and 1354665.6 (4.213%) units for PB1 and PB2 methods, respectively.

To demonstrate the difference between the output of the mentioned algorithms and the exact solution obtained from CPLEX, the solution of the problem instance (1) is reported in Table 7. According to this table, it is observed that all the three methods present the same number and locations for facilities. However, the difference is related to the quantity flow between facilities. The obtained quantity flow among facilities in PB3 methods and the exact solution is the same.

6.4. *Convergence and Speed of Algorithms.* To show the convergence rate of the algorithms and their quality, one of the problem instances, namely, instance #12, is used to demonstrate the performance of the proposed algorithms in Figure 7. The optimality gap is shown for each of the iterations

TABLE 6: The average of the objective values for each sample problem instance.

Scale of the problem	Problem instance	CPLEX	PB1	PB2	PB3
Small	1	5216131.55	5216258.95	5216878.01	5216131.55
	2	6537367.23	6537605.34	6538131.72	6537367.23
	3	7709177.89	7709634.54	7710523.47	7709177.89
	4	9378103.61	9378752.72	9379025.74	9378103.61
	5	11611170.7	11612119.3	11612637.7	11611172.3
	6	12217294.9	12219232.9	12218828.6	12217296.1
	7	14546224.7	14547451.7	14547694.5	14546227.6
Medium	8	15642700.4	15645397.6	15644410.2	15642705.3
	9	16394986.2	16411886.8	16408650.5	16395010.8
	10	18192348.5	18209196.2	18199012.0	18192397.3
	11	19613086.9	19800951.6	19628863.7	19613153.8
	12	22984000.0	23693969.4	23485221.2	22984540.7
	13	25164851.6	26212668.5	26139382.5	25165442.0
	14	32156898.7	34013530.3	33511564.3	32157659.6

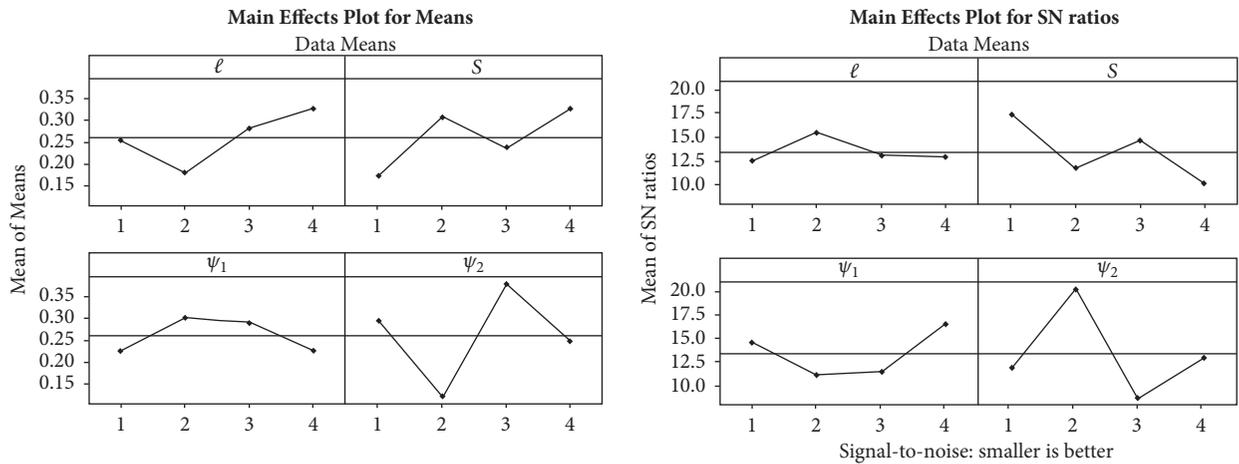


FIGURE 5: Means of means and the S/N ratio plot for discrete LCA.

and it is clearly observable that PB3 method is able to further search the solution space, which raise the chance of obtaining the optimal solution.

The computational time of the algorithms is another interesting factor to point out specially in NP-hard problems. The average computational time of our algorithms for each problem instance is given in Figure 8. According to this figure it is observed that the computational time of the exact solution increases exponentially as the size of the problem increases. The computational time of the exact solution has become more than that of PB1, PB2, and PB3 methods from the problem instance #3 onwards. This time increase shows that the exact solution is inefficient in solving large-scale problems and even it may not achieve the optimal solution. However, the computational times of PB1, PB2, and PB3 algorithms are close to each other and they are less than the solver's except for some very small size problem instances. Generally, the study revealed that the proposed solution method has obtained results very close to those of the exact solution than the other methods. However, its computational time at large sizes is less than that of the exact solution.

6.5. Algorithms Performance for Larger Scale Problems. To evaluate the performance of the proposed solution method for large-scale instances, 7 sample problems were solved using PB1, PB2, and PB3 algorithms each of them five times. As the solver could not find a feasible solution, the performance comparison on the solution quality is made using RPD measure which is defined below:

$$\% \text{ RPD} = \frac{\text{Sol}_{\text{method}} - \text{BestSol}}{\text{BestSol}} \times 100, \quad (20)$$

where  $\text{Sol}_{\text{method}}$  is the solution obtained by each algorithm and  $\text{BestSol}$  is the best solution obtained among all three solution methods. Table 8 presents the computational results in large-scale problems.

Figures 9-10 present a schematic comparison of the average CPU time and the average RPD value of three solution methods, respectively.

6.6. Statistical Analysis. We have used Duncan's multiple range test as a statistical technique to compare the average results of each problem instance to conclude about the

TABLE 7: The location and quantity flow between facilities in problem instance #1.

Variables	PB1	PB2	PB3	CPLEX
$X_{1,1}$	133.96	133.96	133.96	133.96
$X_{3,2}$	126.2	126.2	126.2	126.2
$X_{3,3}$	104.18	104.18	104.18	104.18
$X_{3,4}$	128.34	128.34	128.34	128.34
$X_{1,5}$	135.56	135.56	135.56	135.56
$X_{3,6}$	127.21	127.21	127.21	127.21
$X_{3,7}$	118.03	118.03	118.03	118.03
$X_{1,8}$	86.95	126.21	126.21	126.21
$X_{1,9}$	0	96.95	0	0
$X_{3,8}$	39.26	0	0	0
$X_{3,9}$	120.49	33.55	120.49	120.49
$X_{3,10}$	116.73	116.73	116.73	116.73
$T_{1,1}$	236.4	236.4	236.4	236.4
$Y_{1,1}$	15.32	15.32	15.32	15.32
$Y_{2,1}$	11.22	11.22	11.22	11.22
$Y_{3,1}$	37.27	37.27	37.27	37.27
$Y_{4,1}$	42.78	42.78	42.78	42.78
$Y_{5,1}$	31.34	31.34	31.34	31.34
$Y_{6,1}$	25.15	25.15	25.15	25.15
$Y_{7,1}$	32.58	32.58	32.58	32.58
$Y_{8,1}$	48.33	48.33	48.33	48.33
$Y_{9,1}$	39.11	39.11	39.11	39.11
$Y_{10,1}$	12.82	12.82	12.82	12.82
$S_{1,1}$	59.54	59.54	59.54	59.54

TABLE 8: The average of the objective values for each problem instance (in large scale problems).

Problem instance #	PB1	PB2	PB3
15	36913935	35518893	33938760
16	40865814	39330392	36800070
17	45761578	44397034	42238210
18	49926096	48933683	46045200
19	58336893	56463500	53420906
20	64568402	62410560	58308146
21	69765594	67512119	64137591

performance of the solution methods. The steps of Duncan's multiple range test are as follows:

- (1) Create a tree diagram, whose first node includes the average of  $K$  treatments. Then, sort them in an increasing form and call it  $p$  level.
- (2) Create two branches from the first node. The first node includes the average of treatment 1 to  $K - 1$ , and the second node includes the average of treatment 2 to  $K$ . Then, call it  $p - 1$  level.
- (3) Proceed with step (2) pattern to the point that there are only two pairs of each node.

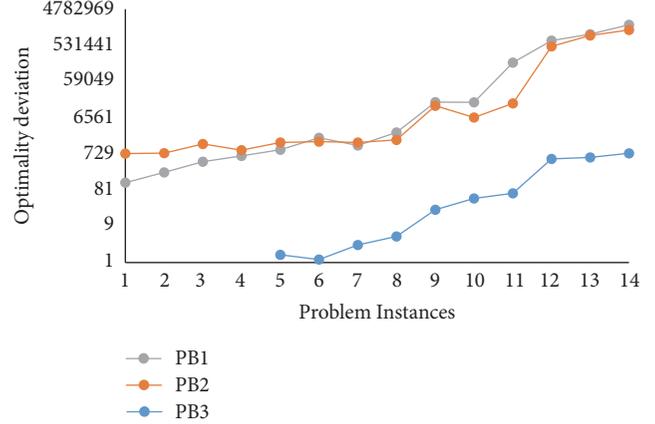


FIGURE 6: The average optimality deviation of the solutions obtained by each solution method within the small and medium size problem instances.

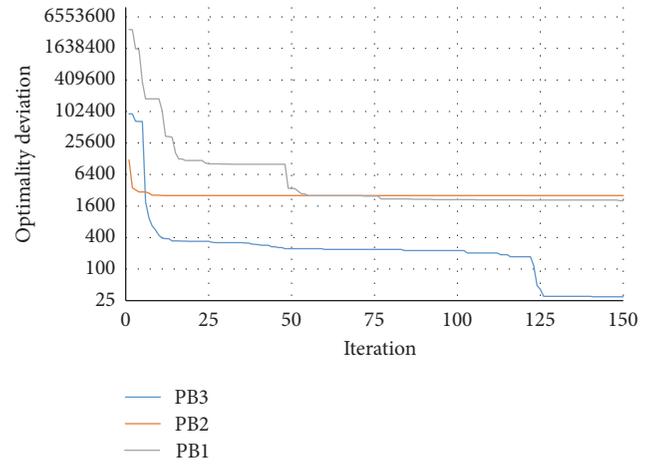


FIGURE 7: The gap between the optimal objective value and the output of the three algorithms over the iterations for a problem instance.

- (4) Calculate the range of  $R$  for each branch of  $p$  level in the tree chart using the following equation:

$$R_p = \frac{S_e r_{\alpha,p,df_e}}{\sqrt{n}}, \quad (21)$$

where  $S_e$  is the standard error in the ANOVA,  $n$  is the number of observations in each treatment, and the values of  $r_{\alpha,p,df_e}$  are obtained from Duncan's multiple range test table for the corresponding  $\alpha$  value.  $r_{\alpha,p,df_e}$  depends on the significance level  $\alpha$ , the number of treatments  $p$  at the node, and the degrees of freedom of the error,  $df_e$ , for the ANOVA.

- (1) Start from the first node and compare the  $R$  value with the least significant level of  $R_p$  Duncan.
  - (a) If  $R > R_p$ , it can be concluded that the two averages are significantly different at the beginning and end of  $p$  level node. Thus, apply

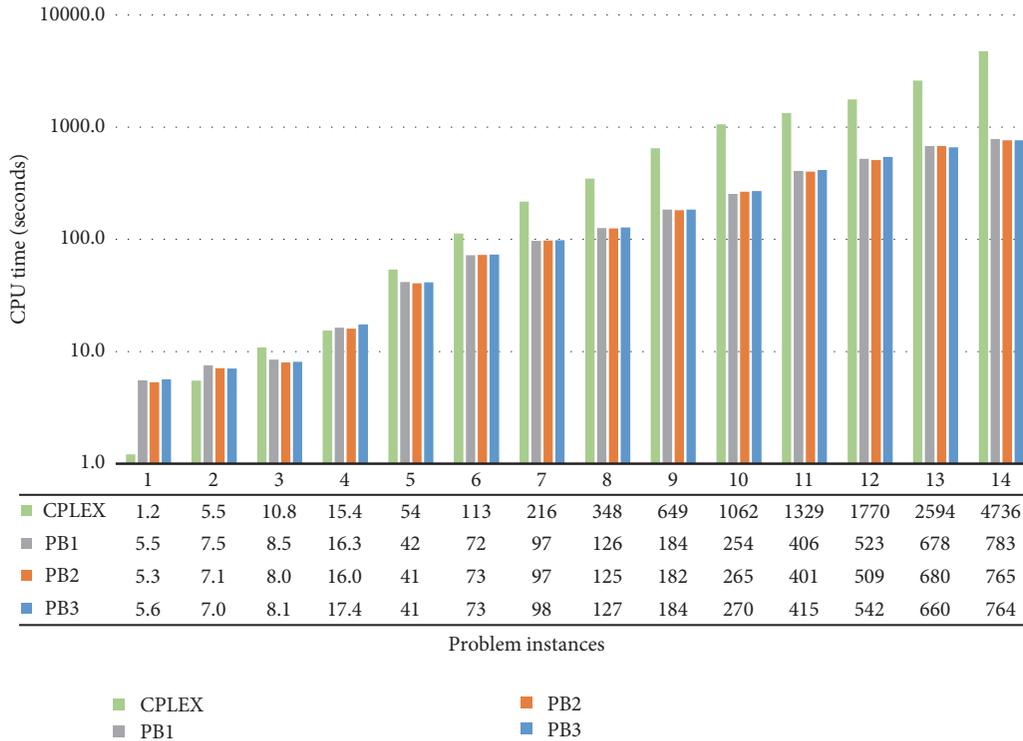


FIGURE 8: The average computational time of methods for each sample problem.

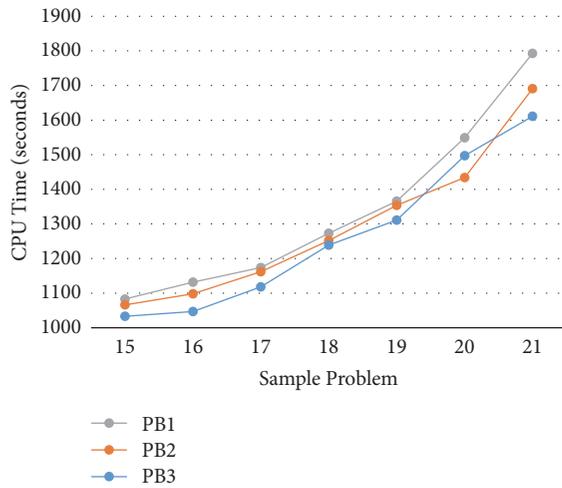


FIGURE 9: The average CPU-Time of each solution method in each large-scale problems.

TABLE 9: The results of the Duncan test among each solution method.

Problem instance #	PB1 v.s. PB2	PB1 v.s. PB3	PB2 v.s. PB3
1	NO	NO	NO
2	NO	NO	NO
3	NO	NO	NO
4	NO	NO	NO
5	NO	YES	YES
6	NO	YES	YES
7	NO	YES	NO
8	NO	NO	YES
9	NO	NO	NO
10	NO	YES	YES
11	NO	YES	YES
12	NO	YES	YES
13	NO	YES	YES
14	NO	YES	YES
15	NO	YES	YES
16	NO	YES	YES
17	YES	YES	YES
18	YES	YES	YES
19	YES	YES	YES
20	YES	YES	YES
21	YES	YES	YES

this test to the next nodes, which is shown by Yes.

- (b) If  $R < R_p$ , it can be concluded that there is no significant difference between the averages of  $p$  level node. Thus, do not apply this test to the next nodes, which is shown by No.

In our test, the aforementioned solution methods were compared via 21 sample problems and the results are reported in Table 9. For example, in the problem instance #12, it can

be seen that there is a significant difference between the averages obtained from application of PB3 and PB2 methods

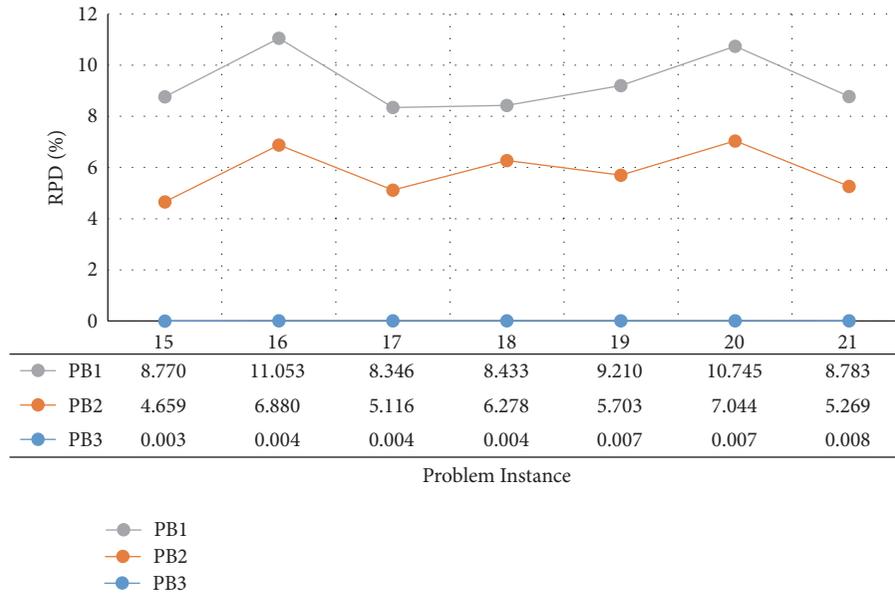


FIGURE 10: The average RPD of each solution method in each large-scale problems.

while there is no significant difference between the averages obtained from application of PB2 and PB1 methods.

### 7. Conclusion and Suggestions for Future Studies

In this paper, a CLSC network including levels of the plants to customer zones, customer zones to collection centers, collection centers to plants, and collection centers to disposal centers was considered. We proposed a mathematical programming model and then developed a new population-based algorithm called the LCA to solve the model.

The proposed solution algorithm was designed by modifying an existing LCA in the literature which is for continuous space. We defined new operators and used them to convert the continuous space into the discrete one and hence we named our algorithm discrete LCA. We also proposed a new priority-based encoding for solving the problems. To demonstrate the effectiveness of the proposed method, an experimental study was conducted to study the small, medium, and large-scale problem instances. The numerical study showed that the outputs of our solution method is very close to the exact optimal solution which are obtained from the solver while it is significantly faster for large size problem instances. Furthermore, from the aspect of objective value, our proposed encoding method outperforms the two other existing methods in the literature which we referred to as PB1 and PB2 within the text whereas there are no significant differences between their computational time.

This work can be extended either with augmenting the model assumption or from the solution approach. For instance, considering a multiobjective model which addresses the sustainability concerns in supply chain network design, either from the customer viewpoint or from the suppliers,

would be an interesting problem to investigate in this framework. In addition, considering uncertainty in the demands and the returned-products can result in a more realistic and challenging problem to deal with.

### Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

### Conflicts of Interest

The authors declare that they have no conflicts of interest.

### Acknowledgments

The research described in this paper was funded by Irans National Elites Foundation, I.R. IRAN, which is gratefully acknowledged here.

### References

- [1] K. Govindan, H. Soleimani, and D. Kannan, "Reverse logistics and closed-loop supply chain: a comprehensive review to explore the future," *European Journal of Operational Research*, vol. 240, no. 3, pp. 603–626, 2015.
- [2] M. Zohal and H. Soleimani, "A hybrid heuristic algorithm for the multistage supply chain network problem," *The International Journal of Advanced Manufacturing Technology*, vol. 26, no. 5, pp. 675–685, 2005.
- [3] F. Du and G. W. Evans, "A bi-objective reverse logistics network analysis for post-sale service," *Computers & Operations Research*, vol. 35, no. 8, pp. 2617–2634, 2008.
- [4] V. Jayaraman, R. A. Patterson, and E. Rolland, "The design of reverse distribution networks: models and solution procedures,"

- European Journal of Operational Research*, vol. 150, no. 1, pp. 128–149, 2003.
- [5] H. R. Krikke, A. Van Harten, and P. C. Schuur, “Business case Océ: reverse logistic network re-design for copiers,” *OR Spectrum*, vol. 21, no. 3, pp. 381–409, 1999.
  - [6] O. Listeş and R. Dekker, “A stochastic approach to a case study for product recovery network design,” *European Journal of Operational Research*, vol. 160, no. 1, pp. 268–287, 2005.
  - [7] N. Aras, D. Aksen, and A. G. Tanuğur, “Locating collection centers for incentive-dependent returns under a pick-up policy with capacitated vehicles,” *European Journal of Operational Research*, vol. 191, no. 3, pp. 1223–1240, 2008.
  - [8] H. Üster, G. Easwaran, E. Akçali, and S. Çetinkaya, “Benders decomposition with alternative multiple cuts for a multi-product closed-loop supply chain network design model,” *Naval Research Logistics (NRL)*, vol. 54, no. 8, pp. 890–907, 2007.
  - [9] Z. Lu and N. Bostel, “A facility location model for logistics systems including reverse flows: the case of remanufacturing activities,” *Computers & Operations Research*, vol. 34, no. 2, pp. 299–323, 2007.
  - [10] R. Wojanowski, V. Verter, and T. Boyaci, “Retail-collection network design under deposit-refund,” *Computers & Operations Research*, vol. 34, no. 2, pp. 324–345, 2007.
  - [11] M. S. Pishvae, K. Kianfar, and B. Karimi, “Reverse logistics network design using simulated annealing,” *The International Journal of Advanced Manufacturing Technology*, vol. 47, no. 1-4, pp. 269–281, 2010.
  - [12] X. Bing, J. M. Bloemhof-Ruwaard, and J. G. A. J. Van Der Vorst, “Sustainable reverse logistics network design for household plastic waste,” *Flexible Services and Manufacturing Journal*, vol. 26, no. 1-2, pp. 119–142, 2014.
  - [13] M. I. Gomes, A. P. Barbosa-Povoa, and A. Q. Novais, “Modelling a recovery network for WEEE: A case study in Portugal,” *Waste Management*, vol. 31, no. 7, pp. 1645–1660, 2011.
  - [14] S. A. Alumur, S. Nickel, F. Saldanha-da-Gama, and V. Verter, “Multi-period reverse logistics network design,” *European Journal of Operational Research*, vol. 220, no. 1, pp. 67–78, 2012.
  - [15] G. Kannan, P. Sasikumar, and K. Devika, “A genetic algorithm approach for solving a closed loop supply chain model: a case of battery recycling,” *Applied Mathematical Modelling: Simulation and Computation for Engineering and Environmental Systems*, vol. 34, no. 3, pp. 655–670, 2010.
  - [16] B. C. Giri and S. Sharma, “Optimizing a closed-loop supply chain with manufacturing defects and quality dependent return rate,” *Journal of Manufacturing Systems*, vol. 35, pp. 92–111, 2015.
  - [17] R. K. Pati, P. Vrat, and P. Kumar, “A goal programming model for paper recycling system,” *Omega*, vol. 36, no. 3, pp. 405–417, 2008.
  - [18] M. Godichaud and L. Amodeo, “Efficient multi-objective optimization of supply chain with returned products,” *Journal of Manufacturing Systems*, vol. 37, pp. 683–691, 2015.
  - [19] M. C. Fonseca, Á. Garca-Sánchez, M. Ortega-Mier, and F. Saldanha-da Gama, “A stochastic bi-objective location model for strategic reverse logistics,” *TOP*, vol. 18, no. 1, pp. 158–184, 2010.
  - [20] E. Özceylan, T. Paksoy, and T. Bektaş, “Modeling and optimizing the integrated problem of closed-loop supply chain network design and disassembly line balancing,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 61, pp. 142–164, 2014.
  - [21] H. Soleimani, M. Seyyed-Esfahani, and G. Kannan, “Incorporating risk measures in closed-loop supply chain network design,” *International Journal of Production Research*, vol. 52, no. 6, pp. 1843–1867, 2014.
  - [22] P. Assarzadegan and M. Rasti-Barzoki, “Minimizing sum of the due date assignment costs, maximum tardiness and distribution costs in a supply chain scheduling problem,” *Applied Soft Computing*, vol. 47, pp. 343–356, 2016.
  - [23] F. T. S. Chan, A. Jha, and M. K. Tiwari, “Bi-objective optimization of three echelon supply chain involving truck selection and loading using NSGA-II with heuristics algorithm,” *Applied Soft Computing*, vol. 38, pp. 978–987, 2016.
  - [24] S. H. Amin and G. Zhang, “A multi-objective facility location model for closed-loop supply chain network under uncertain demand and return,” *Applied Mathematical Modelling: Simulation and Computation for Engineering and Environmental Systems*, vol. 37, no. 6, pp. 4165–4176, 2013.
  - [25] M. Gen, F. Altıparmak, and L. Lin, “A genetic algorithm for two-stage transportation problem using priority-based encoding,” *OR Spectrum*, vol. 28, no. 3, pp. 337–354, 2006.
  - [26] A. Costa, G. Celano, S. Fichera, and E. Trovato, “A new efficient encoding/decoding procedure for the design of a supply chain network with genetic algorithms,” *Computers & Industrial Engineering*, vol. 59, no. 4, pp. 986–999, 2010.
  - [27] R. Jamshidi, S. M. T. F. Ghomi, and B. Karimi, “Multi-objective green supply chain optimization with a new hybrid memetic algorithm using the Taguchi method,” *Scientia Iranica*, vol. 19, no. 6, pp. 1876–1886, 2012.
  - [28] P. Subramanian, N. Ramkumar, T. T. Narendran, and K. Ganesh, “PRISM: PRiority based SiMulated annealing for a closed loop supply chain network design problem,” *Applied Soft Computing*, vol. 13, no. 2, pp. 1121–1135, 2013.
  - [29] M. Zohal and H. Soleimani, “Developing an ant colony approach for green closed-loop supply chain network design: a case study in gold industry,” *Journal of Cleaner Production*, vol. 133, pp. 314–337, 2016.
  - [30] M. Zandieh and A. Chensebli, “Reverse logistics network design: a water flow-like algorithm approach,” *OPSEARCH*, vol. 53, no. 4, pp. 667–692, 2016.
  - [31] R. Singh, A. Chakraborty, M. Kosambia, P. Shah, and A. Karhade, “Reverse logistics using genetic algorithm,” *International Journal of Innovative Research and Development*, vol. 5, no. 2, 2016.
  - [32] S. H. R. Pasandideh and K. Asadi, “A priority-based modified encoding–decoding procedure for the design of a bi-objective sc network using meta-heuristic algorithms,” *International Journal of Management Science and Engineering Management*, vol. 11, no. 1, pp. 8–21, 2016.
  - [33] P. S. Davis and T. L. Ray, “A branch-bound algorithm for the capacitated facilities location problem,” *Naval Research Logistics Quarterly*, vol. 16, no. 3, pp. 331–344, 1969.
  - [34] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization*, vol. 7, John Wiley & Sons, 2000.
  - [35] Z. Michalewicz, G. A. Vignaux, and M. Hobbs, “Nonstandard genetic algorithm for the nonlinear transportation problem,” *ORSA Journal on Computing*, vol. 3, no. 4, pp. 307–316, 1991.
  - [36] A. H. Kashan, “League Championship Algorithm: a new algorithm for numerical function optimization,” in *Proceedings of the International Conference on Soft Computing and Pattern Recognition (SoCPaR '09)*, pp. 43–48, IEEE, Malacca, Malaysia, December 2009.

## Research Article

# Heuristic Determination of Resolving Controls for Exact and Approximate Controllability of Nonlinear Dynamic Systems

Asatur Zh. Khurshudyan 

Department of Dynamics of Deformable Systems and Coupled Fields, Institute of Mechanics,  
National Academy of Sciences of Armenia, Armenia

Correspondence should be addressed to Asatur Zh. Khurshudyan; [khurshudyan@mechins.sci.am](mailto:khurshudyan@mechins.sci.am)

Received 7 February 2018; Revised 12 April 2018; Accepted 18 April 2018; Published 17 May 2018

Academic Editor: Nivaldo Rodríguez

Copyright © 2018 Asatur Zh. Khurshudyan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Dealing with practical control systems, it is equally important to establish the controllability of the system under study and to find corresponding control functions explicitly. The most challenging problem in this path is the rigorous analysis of the state constraints, which can be especially sophisticated in the case of nonlinear systems. However, some heuristic considerations related to physical, mechanical, or other aspects of the problem may allow coming up with specific hierarchic controls containing a set of free parameters. Such an approach allows reducing the computational complexity of the problem by reducing the nonlinear state constraints to nonlinear algebraic equations with respect to the free parameters. This paper is devoted to heuristic determination of control functions providing exact and approximate controllability of dynamic systems with nonlinear state constraints. Using the recently developed approach based on Green's function method, the controllability analysis of nonlinear dynamic systems, in general, is reduced to nonlinear integral constraints with respect to the control function. We construct parametric families of control functions having certain physical meanings, which reduce the nonlinear integral constraints to a system of nonlinear algebraic equations. Regimes such as time-harmonic, switching, impulsive, and optimal stopping ones are considered. Two concrete examples arising from engineering help to reveal advantages and drawbacks of the technique.

## 1. Introduction

The ability of a controlled system to accommodate a required state at a given instant by means of attached controllers is called *controllability*. It is one of the most crucial properties of applied control systems along with stability and reliability. Generally, two main types of controllability are considered for deterministic systems: exact and approximate. If by a specific choice of admissible controls the system can be transmitted from a given state to a required state within a finite amount of time exactly, it is called *exactly controllable*. If the system is not exactly controllable by any choice of admissible control, but its state implemented at the required instant for at least one admissible control is “sufficiently” close to the designated terminal state, it is called *approximately controllable*. Evidently, exact controllability implies approximate controllability with arbitrarily small accuracy. However, in general, approximate controllability does not

imply exact controllability. For further introduction to the concept of controllability with diverse applications, refer to major contributions [1–10] and related references therein. The concept of controllability with applications in engineering has been studied by many authors, a part of which can be found in [11–16] and in related references therein. Results obtained in this paper can be used in relevant applied studies in engineering, e.g., [17–20], for derivation of diverse control regimes.

Let us describe the aforesaid mathematically. Assume that the state of a dynamic system is characterized by the vector-function  $\mathbf{w} : \mathbb{R}^k \times \mathbb{R}^n \times \mathbb{R}^+ \rightarrow \mathbb{R}^m$ ,  $k, n, m \in \mathbb{N}$ , satisfying some constraints, e.g., governing equation, boundary, and initial conditions, among other possible constraints (hereinafter, all those constraints are referred to as state constraints). Then, mathematically the controllability of the system is verified by evaluating the mismatch between the designated state,  $\mathbf{w}_T : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , and the state implemented by a specific choice of

admissible controls  $\mathbf{u} : \mathbb{R}^+ \rightarrow \mathbb{R}^k$  at the given instant  $T$ , i.e., the residue

$$\mathcal{R}_T(\mathbf{u}) = \|\mathbf{w}(\mathbf{u}, \mathbf{x}, T) - \mathbf{w}_T(\mathbf{x})\|_{\mathbf{W}}. \quad (1)$$

Thus, if for at least one admissible control  $\mathbf{u} \in \mathcal{U}$ ,

$$\mathcal{R}_T(\mathbf{u}) = 0, \quad (2)$$

then the system is exactly controllable. If this is not the case, but for at least one admissible control  $\mathbf{u} \in \mathcal{U}$ ,

$$\mathcal{R}_T(\mathbf{u}) \leq \varepsilon, \quad (3)$$

for a given precision  $\varepsilon > 0$ , then the system is approximately controllable.

Here  $\mathbf{u}$  is the control vector-function,  $\mathcal{U}$  is the set of admissible controls,  $T > 0$  is the control time,  $\mathbf{W}$  is the space of the terminal states  $\mathbf{w}_T$  (appropriate Hilbert space), and  $\|\cdot\|_{\mathbf{W}}$  is the norm in  $\mathbf{W}$ . At this, the subscript  $T$  at the residue is a short form of denoting the obvious dependence  $\mathcal{R}_T(\cdot) = \mathcal{R}(\cdot, T)$ .

Hereinafter, we assume that the set of admissible controls has the following general form [10]:

$$\mathcal{U} = \{\mathbf{u} \in \mathbf{U}, |\mathbf{u}| \leq \varepsilon, \text{supp}(\mathbf{u}) \subseteq [0, T]\}, \quad (4)$$

where  $\text{supp}(u) = \overline{\{t \in \mathbb{R}^+, \mathbf{u}(t) \neq \mathbf{0}\}}$  is the support of  $\mathbf{u}$  and  $\mathbf{U}$  is an appropriate Hilbert space. Note that  $\mathcal{U}$  can be complemented by some other possible constraints on  $\mathbf{u}$ . For instance, in the case of boundary controls, if the required state transition is necessarily time-continuous, then  $\mathcal{U}$  is complemented by the compatibility conditions of the initial and boundary conditions (see Section 3 below). The admissible controls providing (2) or (3) are called resolving controls (in corresponding sense). The set of all resolving controls is denoted by  $\mathcal{U}_{\text{res}}$ . Obviously,  $\mathcal{U}_{\text{res}} \subset \mathcal{U}$ . If  $\mathcal{U}_{\text{res}} = \emptyset$ , lack of controllability occurs (see [21–25] and related references therein).

The analysis of controllability for a particular control system can be quite sophisticated and can require burdensome computational costs. This may happen in the case of systems with singularities/discontinuities, uncertain systems, systems with nonlinear state constraints, and so on. However, the evaluation of (1) on  $\mathcal{U}$  can be made less costly if the state vector  $\mathbf{w}$ , that is, the solution of the state constraints, is found explicitly. Unfortunately, there does not exist a unified technique that allows solving general state constraints explicitly. However, there are such powerful techniques as the Adomian decomposition method [26], the homotopy analysis method [27], and nonlinear Green's function method [28, 29] for finding an approximate analytical solution to general types of nonlinear differential equations. Nevertheless, even having the explicit dependence  $\mathcal{R}_T = \mathcal{R}_T(\mathbf{u})$ , it is a very challenging problem to find resolving controls providing (2) or (3) explicitly.

In this paper we develop a systematic algorithm for heuristic determination of explicit expressions for resolving controls providing exact or approximate controllability at the required instant  $T$  (i.e., (2) or (3)) of systems with nonlinear state constraints. Parametric hierarchies of functions are constructed to satisfy the resolving

systems derived by the recently developed Green's function approach [10]. In the case of exact controllability, the resolving system is expanded into series of orthogonal functions. Considered controls include trigonometric, piecewise-constant, impulsive, piecewise-continuous, constant, quasi-polynomial, trigonometric polynomial, and stopping regimes. At this, we cover the cases of both distributed and boundary controls. A rigorous comparison of our approach with the well-known moments problem approach shows that under proper restrictions the known explicit solutions obtained by moments problem approach coincide with our solutions. We test the technique on the examples of a finite elastic beam subjected to an external force with an uncertainty and the viscous Burgers' nonlinear equation. Both cases have diverse applications in modern engineering.

The paper is organized as follows. In Section 2 we outline the two mostly used methods for the analysis of exact and approximate controllability of nonlinear dynamic systems. Then, in Section 3 we derive diverse hierarchies of resolving controls and corresponding constraints on the free parameters for both exact and approximate controllability analysis. In Section 4 we perform a comparative analysis between the controls derived by our approach and those derived by the moments problem approach. Finally, we provide a demonstration of how the derived explicit expressions of resolving controls can be used to ensure exact or approximate controllability of particular dynamic systems.

## 2. Existing Approaches

The verification of (2) or (3) is enough only for establishing whether a particular system is exactly or approximately controllable or not. However, when dealing with real-life control systems, from the practical implementation point of view, it is equally important to find admissible controls providing either type of controllability explicitly, which would significantly reduce controller performance costs. There exist several efficient techniques for analyzing particular systems on exact or approximate controllability. In this section we outline the two most commonly used techniques.

The first technique involves a norm minimizing algorithm to minimize (1) (see [6]). In general statement, the problem is formulated as a constrained minimization problem:

$$\|\mathbf{w}(\mathbf{u}, \mathbf{x}, T) - \mathbf{w}_T(\mathbf{x})\|_{\mathbf{W}} \xrightarrow{\mathbf{u} \in \mathcal{U}} \min, \quad (5)$$

where  $\mathbf{w}$  is subjected to the state constraints. If the minimum is attained and is zero, then

$$\mathbf{u} = \arg \min \|\mathbf{w}(\mathbf{u}, \mathbf{x}, T) - \mathbf{w}_T(\mathbf{x})\|_{\mathbf{W}} \in \mathcal{U} \quad (6)$$

provides exact controllability of the system at  $T$ . If the minimum is not equal to zero, but it remains smaller than the required precision, then the system is approximately controllable. Otherwise, we arrive at the lack of controllability. In other words, following this approach, by a single numerical procedure it is possible to identify whether the system is exactly or approximately controllable or it is not controllable at all. The disadvantage of this approach is in burdensome

computational costs (machinery time) required to run the corresponding numerical scheme in the case of nonlinear state constraints.

The second approach (see [10]) is based on the usage of the system

$$\mathbf{w}(\mathbf{u}, \mathbf{x}, T) - \mathbf{w}_T(\mathbf{x}) = \mathbf{0} \quad \text{a.e. in } \Omega, \quad (7)$$

which is equivalent to exact controllability of the system according to the definition of norm. Here  $\Omega$  is an open subset of  $\mathbb{R}^n$  occupied by the system. Then the set of resolving controls is defined as

$$\mathcal{U}_{\text{res}} = \{\mathbf{u} \in \mathcal{U}, (7)\}. \quad (8)$$

In some exceptional cases,  $\mathbf{w}$  is found from the state constraints explicitly, so that (7) becomes an explicit constraint on  $\mathbf{u}$ . Then, the Newton-Raphson iteration (or similar others) can be involved to determine resolving controls approximately with required precision. The disadvantage of this approach is that the rigorous derivation of  $\mathbf{w}$  from the state constraints is complicated and, in some cases, is impossible. Another disadvantage of this approach may be the necessity of using derivatives of (7) in the numerical procedure.

### 3. Heuristic Determination of Resolving Controls

In this section we describe a method of heuristic determination of resolving controls providing exact or approximate controllability of dynamic systems with nonlinear state constraints. Using Green's function approach [10], the state constraints are reduced to nonlinear integral constraints on the admissible controls. Parametric solutions to the reduced constraints are constructed explicitly. Eventually, a system of nonlinear algebraic equations for the parameters is derived. For the sake of simplicity, hereinafter we restrict ourselves to the one-dimensional case. All the derivations and transformations can be straightforwardly generalized to the case of higher dimensions.

**3.1. Exact Controllability.** In general statement, the exact controllability of a particular system in one space dimension is equivalent to equality type constraint of the form [10]

$$\int_0^T K_T(u, x, \tau) d\tau = f_T(x) \quad \text{a.e. in } [a, b], \quad (9)$$

where  $K_T$  and  $f_T$  are given functions. The subscript  $T$  indicates that the corresponding quantity depends on  $T$ .

The explicit determination of resolving controls from (9) by direct methods can be quite complicated since its both sides depend on  $x$ , while the control depends only on  $t$ . This means that (9) cannot be considered as a Fredholm integral equation of the first kind. Nevertheless, in cases when

$$K_T(u, x, t) = f_T(x) g_T(u, t) \quad \text{in } \mathcal{U} \times [a, b] \times [0, T], \quad (10)$$

for an integrable function  $g_T$ , (9) becomes a Fredholm integral equation of the first kind

$$\int_0^T g_T(u, \tau) d\tau = 1, \quad (11)$$

which can be solved efficiently for generic kernels  $g_T$  (refer to [30] for details).

An efficient way of solving (9) in general can be developed using the expansion of  $K_T$  and  $f_T$  into series of orthogonal functions. Let  $\{\varphi_n\}_{n=1}^{\infty}$  be a family of orthogonal functions in  $[a, b]$  (in some specific cases a family of orthogonal functions with some weight can also be involved); that is,

$$\int_a^b \varphi_m(x) \varphi_n(x) dx = \delta_m^n, \quad (12)$$

where  $\delta_m^n$  is the Kronecker's delta:

$$\delta_m^n = \begin{cases} 1, & m = n, \\ 0, & m \neq n. \end{cases} \quad (13)$$

Denote by  $f_{Tn}$  and  $K_{Tn}$  the expansion coefficients of the functions  $f_T$  and  $K_T$ , respectively, so that (in real problems,  $K_T$  and  $f_T$  are expressed in terms of Green's function of the system under study and its integral [10], so their expansion is convergent)

$$f_T(x) = \sum_{n=1}^{\infty} f_{Tn} \varphi_n(x) \quad (14)$$

$$\text{with } f_{Tn} = \int_a^b f_T(x) \varphi_n(x) dx,$$

$$K_T(u, x, \tau) = \sum_{n=1}^{\infty} K_{Tn}(u, \tau) \varphi_n(x) \quad (15)$$

$$\text{with } K_{Tn}(u, \tau) = \int_a^b K_T(x, \tau) \varphi_n(x) dx.$$

Then, since  $\{\varphi_n\}_{n=1}^{\infty}$  are orthogonal, (9) is equivalent to the infinite system

$$\int_0^T K_{Tn}(u, \tau) d\tau = f_{Tn}, \quad n \in \mathbb{N}. \quad (16)$$

Note that (16) can be treated in different ways. It is an infinite system of Fredholm integral equations of the first kind, which can be efficiently solved numerically [30, 31]. On the other hand, it can be treated as an infinite-dimensional problem of moments (see Section 4 below). In specific problems, depending on, for example, the required accuracy of computations, the infinite system (16) is truncated and considered for some finite  $N$ .

System (16) also can be resolved heuristically. Namely, based on some considerations, say, physical treatment of the problem, controller capability, and so forth, the control function is chosen to have a specific form containing a set of free parameters. Then, this function is substituted into system (16) or its truncated version and a discrete system of, in general, nonlinear algebraic equations is derived with respect to the set of free parameters.

Consider some specific solutions. Let the infinite system (16) be truncated for some finite  $N \in \mathbb{N}$ . Then, the

control function can be sought, for instance, in the form of trigonometric polynomial

$$u(t) = \sum_{m=1}^M u_m \sin(\omega_m t + \gamma_m), \quad (17)$$

where  $M \in \mathbb{N}$  and  $u_m$ ,  $\omega_m$ , and  $\gamma_m$  are free parameters determined appropriately to satisfy (16) exactly. Substituting (17) into the truncated part of system (16), the system of nonlinear equations

$$\mathbf{K}_T(u_m, \omega_m, \gamma_m) = \mathbf{f}_T \quad (18)$$

is obtained. Here

$$\begin{aligned} \mathbf{K}_T &= (\bar{K}_{T1} \ \cdots \ \bar{K}_{TN})^T, \\ \mathbf{f}_T &= (f_{T1} \ \cdots \ f_{TN})^T, \\ \bar{K}_{Tn}(u_m, \omega_m, \gamma_m) & \\ &= \int_0^T K_{Tn} \left( \sum_{m=1}^M u_m \sin(\omega_m \tau + \gamma_m), \tau \right) d\tau, \end{aligned} \quad (19)$$

and the superscript T means transposition.

Consider also the piecewise-constant regime

$$u(t) = \sum_{m=1}^M u_m \theta(t - t_m), \quad (20)$$

corresponding to a finite jump  $|u_{m+1} - u_m|$  from the constant regime  $u(t_m)$  to  $u(t_{m+1})$  when  $t$  switches from  $t_m$  to  $t_{m+1}$ . Here  $u_m$  and  $t_m$  are free parameters. At this, instants  $t_m$  satisfy the inequality type constraints

$$0 \leq t_1 < t_2 < \cdots < t_M \leq T, \quad (21)$$

foreclosing the overlap of different regimes. Here  $\theta$  is the Heaviside function:

$$\theta(t) = \begin{cases} 1, & t > 0, \\ \frac{1}{2}, & t = 0, \\ 0, & t < 0. \end{cases} \quad (22)$$

In this case, (16) is reduced to

$$\mathbf{K}_T(u_m, t_m) = \mathbf{f}_T, \quad (23)$$

with

$$\bar{K}_{Tn}(u_m, t_m) = \int_0^T K_{Tn} \left( \sum_{m=1}^M u_m \theta(\tau - t_m), \tau \right) d\tau. \quad (24)$$

Further, consider the impulsive regime

$$u(t) = \sum_{m=1}^M u_m \delta(t - t_m), \quad (25)$$

formally describing instantaneous impacts  $u_m$  at  $t = t_m$ . Here  $\delta$  is the Dirac function:

$$\delta(t) = \begin{cases} 0, & t \neq 0, \\ \infty, & t = 0. \end{cases} \quad (26)$$

In this case also the free parameters are  $u_m$  and  $t_m$ . Substituting (25) into (16) leads to

$$\mathbf{K}_T(u_m, t_m) = \mathbf{f}_T, \quad (27)$$

with

$$\bar{K}_{Tn}(u_m, t_m) = \int_0^T K_{Tn} \left( \sum_{m=1}^M u_m \delta(\tau - t_m), \tau \right) d\tau. \quad (28)$$

Moreover, here also the instants  $t_m$  satisfy

$$0 \leq t_1 < t_2 < \cdots < t_M \leq T. \quad (29)$$

In many applications the piecewise-continuous control

$$u(t) = \sum_{m=1}^M u_m(t) \chi_{[t_m, t_{m+1}]}(t), \quad (30)$$

$$0 \leq t_1 < \cdots < t_M < t_{M+1} = T,$$

is considered, where

$$\chi_{[t_0, t_1]}(t) = \theta(t - t_0) - \theta(t - t_1) \quad (31)$$

is the characteristic function of the interval  $[t_0, t_1]$ .

This regime corresponds to switching between the time-dependent regimes  $u_m$ . At this, any of the regimes (17), (20), and (25) can be used as  $u_m$ . In this case the resolving equation is reduced to

$$\mathbf{K}_T(u_m, t_m) = \mathbf{f}_T, \quad (32)$$

with

$$\begin{aligned} \bar{K}_{Tn}(u_m, t_m) & \\ &= \int_0^T K_{Tn} \left( \sum_{m=1}^M u_m(\tau) \chi_{[t_m, t_{m+1}]}(\tau), \tau \right) d\tau. \end{aligned} \quad (33)$$

In the case of boundary control,  $K_T$  is linear in  $u$ ; that is

$$K_T(u, x, \tau) = K_{T0}(x, \tau) u(\tau) + K_{T1}(x, \tau), \quad (34)$$

for given functions  $K_{T0}$  and  $K_{T1}$ .

In addition,  $u$  satisfies the boundary conditions

$$\begin{aligned} u(0) &= u_0, \\ u(T) &= u_T, \end{aligned} \quad (35)$$

reduced from the compatibility of given boundary and initial data.

Then, (18), (23), and (27) are reduced to linear systems for  $u_m$ . Indeed, in the case of, for example, (25), the resulting system for the free parameters will be

$$\mathbf{K}_{T0}(u_m, t_m) = \mathbf{f}_T - \mathbf{K}_{T1}, \quad (36)$$

where

$$\begin{aligned} \mathbf{K}_{T0} &= (\bar{K}_{T01} \ \cdots \ \bar{K}_{T0N})^T, \\ \mathbf{K}_{T1} &= (\bar{K}_{T11} \ \cdots \ \bar{K}_{T1N})^T, \\ &\int_0^T \left[ K_{T0n}(\tau) \sum_{m=1}^M u_m \delta(\tau - t_m) + K_{T1n}(\tau) \right] d\tau \\ &= \sum_{m=1}^M u_m \int_0^T K_{T0n}(\tau) \delta(\tau - t_m) d\tau \\ &\quad + \int_0^T K_{T1n}(\tau) d\tau \\ &= \sum_{m=1}^M u_m K_{T0n}(t_m) + \int_0^T K_{T1n}(\tau) d\tau \\ &:= \bar{K}_{T0n}(u_m, t_m) + \bar{K}_{T1n}. \end{aligned} \quad (37)$$

Here  $K_{T0n}$  and  $K_{T1n}$  are the expansion coefficients of  $K_{T0}$  and  $K_{T1}$  into series of  $\{\varphi_n\}_{n=1}^\infty$ , respectively. At this, (35) implies additional restrictions on  $u_m$ : when  $t_1 = 0$  and  $t_M = T$ , then

$$\begin{aligned} u_1 &= u_0, \\ u_M &= u_T. \end{aligned} \quad (38)$$

Otherwise, (25) is applied in cases when  $u_0 = u_T = 0$ .

Eventually, different order numerical methods can be involved to approximate the nonlinear systems derived above for the free parameters. See, for instance, [32, 33].

*Remark 1.* In general, the  $N$ -dimensional system (18) contains  $3M$  unknowns; therefore it might be irresolvable. Nevertheless, if some of the free parameters, say  $\omega_m$  and/or  $\gamma_m$ ,  $m = 1, \dots, M$ , are prescribed, then (18) may become solvable. Moreover, in the case of boundary control, (18) is reduced to a linear system for  $u_m$ . Therefore, if  $\omega_m$  and  $\gamma_m$  are prescribed and  $M = N$ ,  $u_m$  are found straightforwardly. Otherwise, that is, when  $M \neq N$ , for finding specific solutions, techniques of nonlinear programming [34] must be involved.

The same reasoning concerns also systems (23) and (27) containing, in general,  $2M$  unknowns.

*Remark 2.* Any solution derived from the truncated version of the infinite-dimensional system (16) is approximate, which means that (9), in general, is satisfied approximately.

**3.2. Approximate Controllability.** Assume that the system under study is linear in control. Then, its approximate

controllability is reduced to the evaluation of an integral equation of the form [10]

$$\int_0^T g_T(\tau) |u(\tau)| d\tau = M_T, \quad (39)$$

for  $M_T \geq 0$ , bounded kernel  $g_T \geq 0$ , and  $u \in \mathcal{U}$ . In the case of boundary control, the control function is additionally constrained by the discrete constraints (35).

It is easy to see that one of the obvious solutions to (39) is the constant regime

$$u = u^o = \text{const}, \quad (40)$$

leading (39) to

$$|u^o| = \frac{M_T}{s_T} \quad \text{with } s_T = \int_0^T g_T(\tau) d\tau. \quad (41)$$

When the control is carried out by the boundary data, this regime is applicable only in the case when  $u_0 = u_T = u^o \neq 0$ .

In vibration control problems usually time-harmonic controls of the form

$$u(t) = u^o \sin(\omega t + \gamma) \quad (42)$$

are considered, leading (39) to the nonlinear equation

$$|u^o| \int_0^T g_T(\tau) |\sin(\omega \tau + \gamma)| d\tau = M_T \quad (43)$$

with respect to the free parameters  $u^o$ ,  $\omega$ , and  $\gamma$ . In the case of boundary control, (35) provides the additional constraint

$$u_0 \sin(\omega T + \gamma) = u_T \sin(\gamma). \quad (44)$$

A particular solution of (39), (35) can be constructed using the quasi-polynomial control

$$u(t) = \sum_m \sum_n u_{mn} t^m (T-t)^n, \quad (45)$$

where  $u_{mn}$  and  $m, n \in \mathbb{R}^+$  are free parameters satisfying system (39) (note that in a proper treatment,  $m$  and  $n$  may also accept negative values). Assume, for simplicity, that  $\text{sign } u \geq 0$ . Then, (39) provides

$$\sum_m \sum_n u_{mn} P_T^{m,n} = M_T, \quad (46)$$

for determination of  $u_{mn}$ ,  $m$ , and  $n$ . Here

$$P_T^{m,n} = \int_0^T g_T(\tau) \tau^m (T-\tau)^n d\tau. \quad (47)$$

Evidently, when  $m$  and  $n$  are prescribed, (46) becomes a linear equation for  $u_{mn}$ .

In the case of boundary control, in order to satisfy (35) by the quasi-polynomial regime, the term

$$u^0(t) = u_0 + [u_T - u_0] \frac{t}{T} \quad (48)$$

must be added to (45). Then, (46) is reduced to

$$\sum_m \sum_n u_{mn} P_T^{m,n} = M_T - u_0 P_T^{0,0} - \frac{u_T - u_0}{T} P_T^{1,0}. \quad (49)$$

A fast verification of approximate controllability is provided by the trigonometric regime

$$u(t) = \sum_{m=1}^M \sum_n u_{mn} \left[ 1 - \cos\left(\frac{2\pi mt}{T}\right) \right]^n, \quad (50)$$

where  $u_{mn}$ ,  $m$ ,  $M \in \mathbb{N}$ , and  $n \in \mathbb{R}^+$  are the free parameters to satisfy (39). Assuming that  $\text{sign } u \geq 0$ , for the determination of the free parameters, the nonlinear equation

$$\sum_{m=1}^M \sum_n u_{mn} C_T^{m,n} = M_T \quad (51)$$

is derived, where

$$C_T^{m,n} = \int_0^T g_T(\tau) \left[ 1 - \cos\left(\frac{2\pi m\tau}{T}\right) \right]^n d\tau. \quad (52)$$

In the case of boundary control, the term  $u^0$  above should be added to the control.

Other particular solutions appropriate for the physical treatment of the problem are also possible. In many applied problems it becomes necessary to involve sliding modes [35]. As an example, consider the switching or piecewise-constant control regime

$$u(t) = \sum_{m=1}^M u_m \theta(t - t_m), \quad (53)$$

subject to the inequality type constraints

$$0 \leq t_1 < \dots < t_M \leq T. \quad (54)$$

Here  $u_m$ ,  $t_m$ , and  $M$  are free parameters. Note that piecewise-continuous regimes with  $u_m = u_m(t)$  are also often considered. In such cases, any of continuous regimes above can be considered as  $u_m$ .

Assuming that  $\text{sign } u \geq 0$ , (39) can be reduced to the nonlinear equation

$$\sum_{m=1}^M \Theta_T^m(u_m, t_m) = M_T, \quad (55)$$

where

$$\begin{aligned} \Theta_T^m(u_m, t_m) &= \int_0^T g_T(\tau) u_m(\tau) \theta(\tau - t_m) d\tau \\ &= \int_{t_m}^T g_T(\tau) u_m(\tau) d\tau. \end{aligned} \quad (56)$$

Note that in the case of boundary control, the additional constraints

$$\begin{aligned} u_1 &= u_0, \\ \sum_{m=1}^M u_m(T) &= u_T \end{aligned} \quad (57)$$

are derived when  $t_1 = 0$ . Otherwise,  $u_0 = u(0)$  should be added to the right hand side of (53).

Another application of sliding mode control is the so-called optimal stopping regime usually given by

$$u(t) = u^o(t) \theta(t^o - t), \quad (58)$$

and (39) must be satisfied by an appropriate choice of  $u^o$  and  $t^o$ . In addition,  $t^o$  satisfies the inequality type constraint

$$0 < t^o \leq T. \quad (59)$$

In this case, (39) is reduced to the nonlinear equation

$$\Theta_T(u^o, t^o) = M_T, \quad (60)$$

where

$$\begin{aligned} \Theta_T(u^o, t^o) &= \int_0^T g_T(\tau) |u^o(\tau)| \theta(t^o - \tau) d\tau \\ &= \int_0^{t^o} g_T(\tau) |u^o(\tau)| d\tau. \end{aligned} \quad (61)$$

Therefore, if  $t^o$  is prescribed and  $u^o = \text{const}$ , then

$$|u^o| = \frac{M_T}{\Theta_T^0} \quad \text{with} \quad \Theta_T^0 = \int_0^{t^o} g_T(\tau) d\tau. \quad (62)$$

In the case of boundary control, if  $t^o < T$ , only the first condition in (35) can be satisfied by  $u^o$  and necessarily  $u_T$  should be zero. Otherwise, both conditions must be satisfied by  $u^o$ .

The set of reachable terminal states can be significantly extended, by complementing  $\mathcal{U}$  with impulsive actions of the form

$$u(t) = \sum_{m=1}^M u_m \delta(t - t_m), \quad (63)$$

subject to the inequality type constraints

$$0 \leq t_1 < \dots < t_M \leq T. \quad (64)$$

The free parameters are  $u_m$ ,  $t_m$ , and  $M$ . If in addition all  $u_m \geq 0$ , then (39) is reduced to the nonlinear constraint

$$\sum_{m=1}^M u_m D_T(t_m) = M_T, \quad (65)$$

where

$$D_T(t_m) = \int_0^T g_T(\tau) \delta(\tau - t_m) d\tau = g_T(t_m). \quad (66)$$

In the case of boundary control, if  $t_1 = 0$  and  $t_M = T$ , (35) is satisfied only when

$$\begin{aligned} u_1 &= u_0, \\ u_N &= u_T. \end{aligned} \quad (67)$$

Otherwise, (63) is applicable if and only if

$$\begin{aligned} u_0 &= 0, \\ u_N &= u_T \\ \text{or } u_1 &= u_0, \\ u_N &= 0 \\ \text{or } u_0 &= 0, \\ u_T &= 0, \end{aligned} \tag{68}$$

respectively.

*Remark 3.* In all cases above, together with constraints above, according to the definition of  $\mathcal{U}$  above, free parameters must satisfy also the inequality type constraint

$$|u| \leq \epsilon \quad \text{in } [0, T]. \tag{69}$$

This means that more likely techniques of nonlinear programming must be involved.

*Remark 4.* As it was noted above, the derived systems of nonlinear algebraic equations can be efficiently approximated by existing powerful numerical techniques. In applied problems the derivation of (9) and (39) remains the most difficult step towards application of this method.

#### 4. Comparison with the Moments Problem Approach

One of the well developed and widely used approaches that allows finding resolving controls explicitly is Krasovskii's moments problem approach. It has been initially developed in [36] for the linear finite-dimensional moments problem. See also [2] for more details. Later the algorithm has been extended to nonlinear finite-dimensional systems in [37] related to applications in mobile control problems [38]. Using an integral representation formula for the general solution of the state constraints (equivalent to Green's function solution) and satisfying the terminal conditions, the explicit determination of the resolving controls is reduced to a moments problem. At this, for concentrated parameter systems, the resulting moments problem is finite-dimensional, while for distributed parameter systems, it is infinite-dimensional.

When (16) is linear in  $u$ , its treatment as a problem of moments has several advantages, including the availability of explicit  $L^p$ -optimal solutions for  $1 \leq p \leq \infty$  and necessary and sufficient conditions for the optimal solution existence. Let us formulate the following theorem giving the explicit form of the  $L^p$ -optimal for  $1 \leq p \leq \infty$  solution of (16) linear in  $u$  (see [39] for details). For the nonlinear case see [37].

**Theorem 5.** *The general solution  $u \in L^p[0, T]$ ,  $1 < p \leq \infty$ , of the linear finite-dimensional moments problem*

$$\int_0^T K_{Tn}(\tau) u(\tau) d\tau = f_{Tn}, \quad n = 1, \dots, N, \tag{70}$$

exists, is unique, and reads as

$$u^o(t) = \lambda_N^q \left| \sum_{n=1}^N l_n^o K_{Tn}(t) \right|^{q-1} \cdot \text{sign} \left[ \sum_{n=1}^N l_n^o K_{Tn}(t) \right], \tag{71}$$

$$\frac{1}{p} + \frac{1}{q} = 1,$$

where  $\lambda_N$  and the Lagrange multipliers  $\{l_n^o\}_{n=1}^N$  are determined from the following two equivalent problems of conditional extremum:

(i) find

$$\lambda_N = \sum_{n=1}^N l_n^o f_{Tn} = \max_{\{l_n\}_{n=1}^N} \sum_{n=1}^N l_n f_{Tn}, \tag{72}$$

under the condition

$$\left[ \int_0^T \left| \sum_{n=1}^N l_n K_{Tn}(\tau) \right|^q d\tau \right]^{1/q} = 1; \tag{73}$$

(ii) find

$$\frac{1}{\lambda_N} = \min_{\{l_n\}_{n=1}^N} \left[ \int_0^T \left| \sum_{n=1}^N l_n K_{Tn}(\tau) \right|^q d\tau \right]^{1/q}, \tag{74}$$

under the condition

$$\sum_{n=1}^N l_n f_{Tn} = 1. \tag{75}$$

Moreover, the minimal value of the norm  $\|u\|_{L^p[0, T]}$  is equal to

$$\|u\|_{L^p[0, T]} = \lambda_N. \tag{76}$$

In the special case when  $p = 1$  ( $q = \infty$ ), the resolving admissible controls are determined as a weak limit of (71) as follows:

$$u^o(t) = \sum_{k=1}^K u_k^o \delta(t - t_k^o) \cdot \text{sign} \left[ \sum_{n=1}^N l_n^o K_{Tn}(t_k^o) \right], \tag{77}$$

where  $t_k^o$  are determined from

$$\begin{aligned} \text{vrai max}_{t \in [0, T]} \left| \sum_{n=1}^N l_n^o K_{Tn}(t) \right| \\ = \min_{\{l_n\}_{n=1}^N} \text{vrai max}_{t \in [0, T]} \left| \sum_{n=1}^N l_n K_{Tn}(t) \right|, \end{aligned} \tag{78}$$

and  $u_k^o$ -from (70) with substituted (77) into it.

Thus, (71) and (77) are the solutions of (70) minimizing

$$\kappa[u] = \|u\|_{L^p[0, T]} \quad \text{on } \mathcal{U}. \tag{79}$$

It is easy to see that, under some restrictions on the parameters  $u_m$  and  $t_m$ , the control regime (25) obtained above heuristically coincides with the  $L^1$ -optimal solution (77). It is noteworthy that when  $p = 2$  and  $K_{T_n}$  is a set of sine functions or quasi-polynomials, then, under proper constraints on the free parameters, the heuristically determined solutions (17) and (45) also coincide with the corresponding  $L^2$ -optimal solutions. Finally, when  $p = \infty$ , the solution of the moments problem (70) is represented in the form of a switching regime, which, under proper constraints, coincides with (20).

The explicit solution of the general finite-dimensional nonlinear moments problem is not known. It is hard to obtain explicit solutions even for some simple forms of  $K_{T_n} = K_{T_n}(u, t)$  [40]. Therefore, the explicit expressions of controls heuristically determined above can play into hands of applied mathematicians and engineers to directly identify control regimes suitable in their studies of nonlinear control problems.

## 5. Examples

In this section we demonstrate the ways of using the heuristically determined controls in practical cases.

**5.1. Beam Subjected to Load with Uncertainty.** Consider an elastic beam of finite length  $l$  subjected to a concentrated dynamical load of intensity  $P$ . The exact location of the load application, denoted by  $x_0$ , is not known. However it is given that  $x_0 \in (l_0, l_1) \subset (0, l)$  for given  $l_0$  and  $l_1$ . Assume that  $x = l$  end of the beam is simply supported, while at  $x = 0$  only the bending moment is fixed, and the transverse deflection is controlled. The aim is to find boundary controls providing controllability of the beam in a given finite time  $T > 0$ . Suppose that the load vanishes at some  $T_0$ ,  $0 < T_0 < T$ . Since there exists an uncertainty in the system, it is hardly possible to provide exact controllability in finite time [41]; therefore approximate controllability of the beam is studied.

Assume also that the beam is sufficiently thin and the load intensity varies in such a range that the beam undergoes merely infinitesimal strains. In that case, the Euler-Bernoulli assumptions can be involved for deriving the mathematical model of the beam. Then, limiting the consideration by the linear elasticity, the transverse deflection of the beam, denoted here by  $w$ , satisfies the fourth order differential equation

$$EJ \frac{\partial^4 w}{\partial x^4} + \rho S \frac{\partial^2 w}{\partial t^2} = P(t) \chi_{[0, T_0]}(t) \delta(x - x_0) \quad (80)$$

in  $(0, l) \times \mathbb{R}^+$ ,

subject to the boundary conditions

$$\begin{aligned} w(0, t) &= u_b(t), \\ w(l, t) &= \frac{\partial^2 w}{\partial x^2} \Big|_{x=0, l} = 0 \end{aligned} \quad (81)$$

in  $\mathbb{R}^+$ .

Here  $E$  is the Young's modulus,  $\rho$  is the density,  $J$  is the cross-sectional moment of inertia, and  $S$  is the cross-sectional

area of the beam. The quantity  $EJ$  measures the resistance of the beam against bending load and is referred to as bending stiffness.

At the initial time instant  $t = 0$  the state of the beam is known:

$$\begin{aligned} w(x, 0) &= w_0(x), \\ \frac{\partial w}{\partial t} \Big|_{t=0} &= w_{01}(x) \end{aligned} \quad (82)$$

in  $[0, l]$ .

Mathematically, the problem is to find such a boundary control function  $u_b$  that provides the inequality

$$\mathcal{R}_T(u_b) = \|w(x, T) - w_T(x)\|_{L^2[0, l]} \leq \varepsilon, \quad (83)$$

in a required finite  $T$  with desired precision  $\varepsilon > 0$ . The functions  $w_T, w_{T1} \in L^2[0, l]$  are given. As in many applications; here it is assumed that  $w_T \equiv 0$ ; i.e., it is required to suspend the vibrations of the beam.

The initial and boundary data are supposed to be consistent:

$$\begin{aligned} u_b(0) &= w_0(0), \\ \dot{u}_b(0) &= w_{01}(0), \\ u_b(T) &= w_T(0), \\ \dot{u}_b(T) &= w_{T1}(0). \end{aligned} \quad (84)$$

Then, the set of admissible controls is considered to be

$$\mathcal{U} = \{u \in L^2[0, T], |u| \leq \varepsilon, \text{supp}(u) \subseteq [0, T], (84)\}. \quad (85)$$

For further analysis it is convenient to use the dimensionless quantities

$$\frac{x}{l}, \frac{ct}{l}, \frac{w}{l}, \frac{Pl^2}{EJ}, \quad (86)$$

for the coordinate, time, deflection, and load intensity, respectively. Here  $c$  is the speed of elastic wave propagation in the beam:

$$c = \sqrt{\frac{E}{\rho}}. \quad (87)$$

New symbols for those quantities are not introduced, in order to make the reading convenient. Then, the general solution of (80), (81), and (82) according to the Green's representation formula is given by

$$\begin{aligned} w(x, t) &= \int_0^1 \left[ w_0(\xi) \frac{\partial G(x, \xi, t)}{\partial t} + w_{01}(\xi) \right. \\ &\cdot G(x, \xi, t) \Big] d\xi + \int_0^t \int_0^1 P(\tau) \chi_{[0, T_0]}(\tau) \\ &\cdot \delta(\xi - x_0) G(x, \xi, t - \tau) d\xi d\tau + \int_0^t u_b(\tau) \\ &\cdot G_b(x, t - \tau) d\tau \quad \text{in } [0, 1] \times \mathbb{R}^+. \end{aligned} \quad (88)$$

Here [42]

$$\begin{aligned}
 G(x, \xi, t) &= \frac{2\alpha}{\pi^2} \sum_{n=1}^{\infty} \frac{1}{n^2} \varphi_n(x) \varphi_n(\xi) \psi_n(t), \\
 G_b(x, t) &= 2\pi\alpha \sum_{n=1}^{\infty} n \varphi_n(x) \psi_n(t), \\
 \varphi_n(x) &= \sin(\pi n x), \\
 \psi_n(t) &= \sin\left(\frac{\pi^2 n^2}{\alpha} t\right), \\
 \alpha^2 &= \frac{S l^2}{J}.
 \end{aligned} \tag{89}$$

Obviously,  $w \in L^2([0, 1] \times \mathbb{R}^+)$ , so that the residue (83) is well-defined.

Evaluating the expression (88) at  $t = T$  and substituting into the residue (83), by virtue of the triangle and Cauchy-Schwartz inequalities, the following estimate is derived:

$$\mathcal{R}_T(u_b) \leq \int_0^T g(T - \tau) |u_b(\tau)| d\tau + \|M_T\|_{L^2[0,1]}, \tag{90}$$

where

$$g(t) = \|G_b(x, t)\|_{L^2[0,1]} = \sqrt{2\pi\alpha} \sqrt{\sum_{n=1}^{\infty} n^2 \psi_n^2(t)},$$

$$\begin{aligned}
 M_T(x, x_0) &= \int_0^1 \left[ w_0(\xi) \frac{\partial G(x, \xi, t)}{\partial t} + w_{01}(\xi) G(x, \xi, t) \right] \Big|_{t=T} d\xi \\
 &+ \int_0^{T_0} P(\tau) G(x, x_0, T - \tau) d\tau \\
 &= \sum_{n=1}^{\infty} A_n(x_0, T) \varphi_n(x),
 \end{aligned} \tag{91}$$

$$\begin{aligned}
 A_n(x_0, T) &= 2w_{0n}\psi_{1n}(T) + \frac{2\alpha}{\pi^2 n^2} w_{01n}\psi_n(T) \\
 &+ \frac{2\alpha}{\pi^2 n^2} P_n(T_0, T) \varphi_n(x_0),
 \end{aligned}$$

$$w_{0n} = \int_0^1 w_0(\xi) \varphi_n(\xi) d\xi,$$

$$P_n(T_0, T) = \int_0^{T_0} P(\tau) \psi_n(T - \tau) d\tau.$$

Furthermore, the direct integration leads to

$$\|M_T\|_{L^2[0,1]}^2 = \frac{1}{2} \sum_{n=1}^{\infty} A_n^2(x_0, T). \tag{92}$$

It is evident from the expression for  $A_n$  that when  $x_0 \searrow 0$  or  $x_0 \nearrow 1$ , then  $M_T$  becomes independent of  $x_0$ . It is also obvious that the “worst” influence of  $x_0$  on  $\|M_T\|_{L^2[0,1]}$  occurs when  $x_0 \rightarrow 1/2$ . A consequence which should be expected.

However, computations show that when  $P_0 \sim 10^5$ , the elastic displacements of the beam are of  $10^{-5}$  order when the load is active and of  $10^{-9}$  order when it vanishes (see Figure 1).

Therefore the required precision must be at least  $\varepsilon \sim 10^{-10}$ . On the other hand, the residual axial stresses arising in the beam after the load vanishes have larger values and can serve as controllability criterion. Thus, as a residue considering the quantity

$$\overline{\mathcal{R}}_T(u) = \left\| E \frac{\partial^2 w}{\partial x^2} \Big|_{t=T} - \sigma_0 \right\|_{L^2[0,l]}, \tag{93}$$

where

$$\sigma_x(x, T) = E \frac{\partial^2 w}{\partial x^2} \Big|_{t=T} \quad \text{in } [0, l], \tag{94}$$

is the axial stress of the beam evaluated at  $t = T$  and  $\sigma_0$  is a prescribed threshold value. Thus,  $\overline{\mathcal{R}}_T$  evaluates how close is the axial stress to a given threshold at the instant  $T$ .

For specific analysis, let us return to the quantities with dimension. Consider a beam of length  $l = 1$  m and of square cross-section with dimensions  $0.1$  m  $\times$   $0.1$  m; therefore  $J = b^4/12 \approx 8.3 \cdot 10^{-6}$  m<sup>4</sup>. Let the material of the beam be made of copper; i.e.,  $E = 1.1 \cdot 10^{11}$  N/m<sup>2</sup>,  $\rho = 8960$  kg/m<sup>3</sup>. Limit the computations to the case when  $x_0 = l/2$ . Assume that

$$P(t) = P_0 \sin(\pi t) [\theta(t - 0) - \theta(t - T_0)], \tag{95}$$

with  $T_0 = 1$  s. Set  $T = 1.2$  s.

Note that the maximal value of  $\sigma_x(x, T) = 300$  N/m<sup>2</sup> (see Figure 2(b)).

In order to reduce this value, the threshold value is set to  $\sigma_0 = 200$  N/m<sup>2</sup> and the optimal stopping boundary control regime

$$u(t) = u^o \sin(\omega_1 t + \omega_2) [\theta(t - 0) - \theta(t - t^o)] \tag{96}$$

is chosen to ensure the inequality

$$\overline{\mathcal{R}}_T(u) \leq \varepsilon \tag{97}$$

for (93) with the precision  $\varepsilon = 10^{-5}$ . Above  $u^o$ ,  $t^o$ ,  $\omega_1$ , and  $\omega_2$  are free parameters. At this,  $t^o$  is constrained by  $0 < t^o \leq T$ . Then, inequality (97) holds with  $\varepsilon = 10^{-5}$  when

$$\begin{aligned}
 u^o &= 10^{-9}, \\
 \omega_1 &= 1, \\
 \omega_2 &= 0, \\
 t^o &= 0.85.
 \end{aligned} \tag{98}$$

Moreover, Figure 3 shows that beside ensuring (97), the considered control regime reduces the total displacement of the beam almost twice.

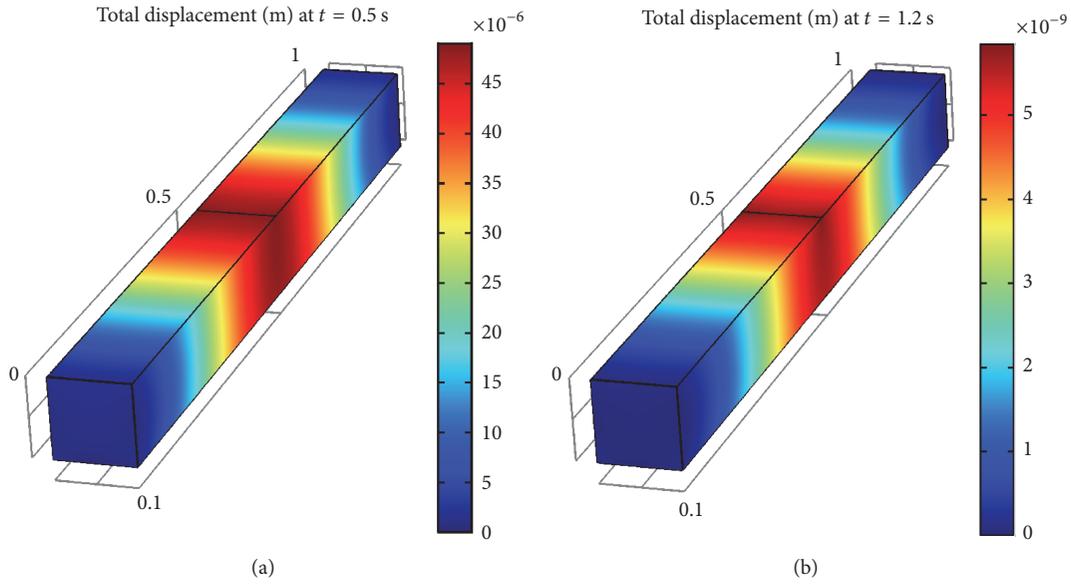


FIGURE 1: Maximal uncontrolled displacement at  $t = 0.5$  s (a) and at  $t = T$  (b).

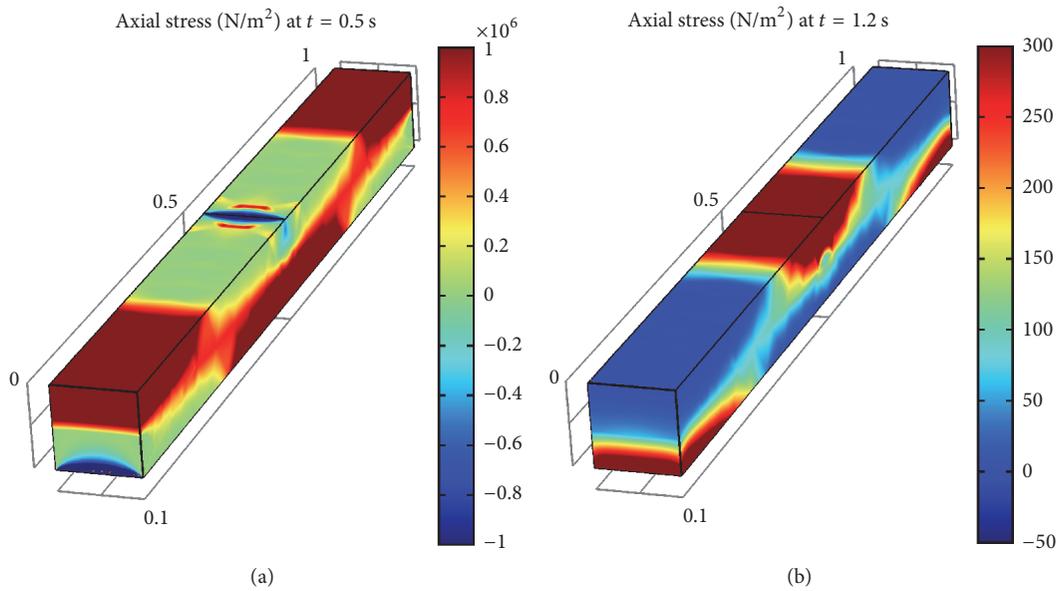


FIGURE 2: Maximal uncontrolled axial stress at  $t = 0.5$  s (a) and at  $t = T$  (b).

Consider now the switching regime

$$u(t) = u_1 [\theta(t - 0) - \theta(t - t_1)] + u_2 [\theta(t - t_2) - \theta(t - t_3)], \quad (99)$$

where  $u_1$ ,  $u_2$ ,  $t_1$ ,  $t_2$ , and  $t_3$  are free parameters. Let the threshold value in this case be  $\sigma_0 = 150 \text{ N/m}^2$ . Then,

$$u_1 = u_2 = 10^{-9}, \\ t_1 = 0.25,$$

$$t_2 = 0.4,$$

$$t_3 = 0.65,$$

(100)

provide (97) with  $\varepsilon = 10^{-5}$ . Note that the total displacement in this case also is reduced almost twice (see Figure 4).

5.2. *Burgers' Equation.* Consider the nonlinear Burgers' equation

$$\frac{\partial w}{\partial t} + w \frac{\partial w}{\partial x} = d \frac{\partial^2 w}{\partial x^2}, \quad (101)$$

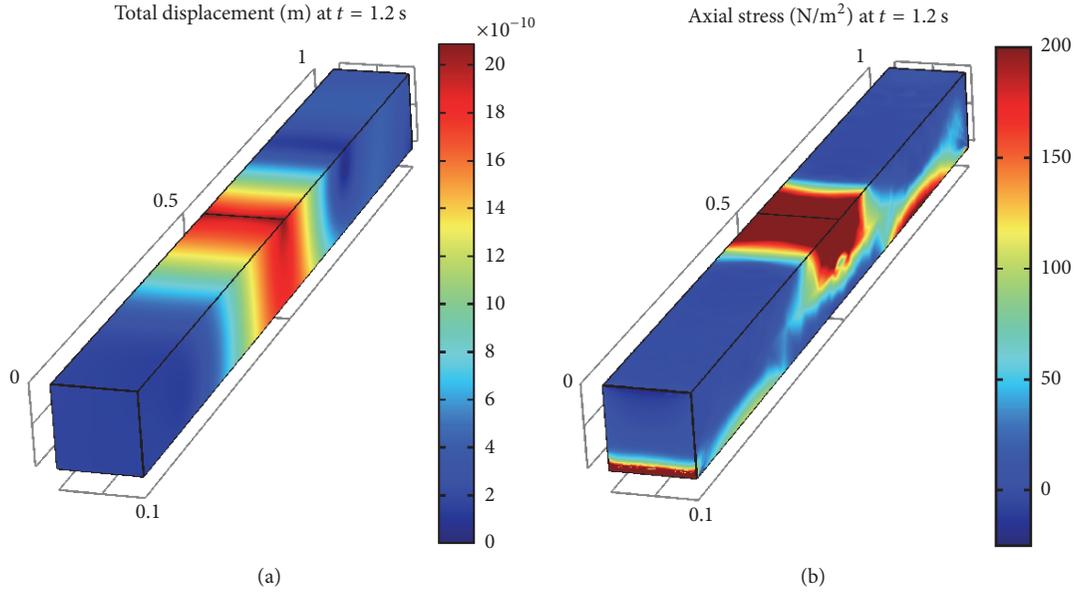


FIGURE 3: Controlled total displacement (a) and axial stress of the beam at  $t = T$ : optimal stopping control.

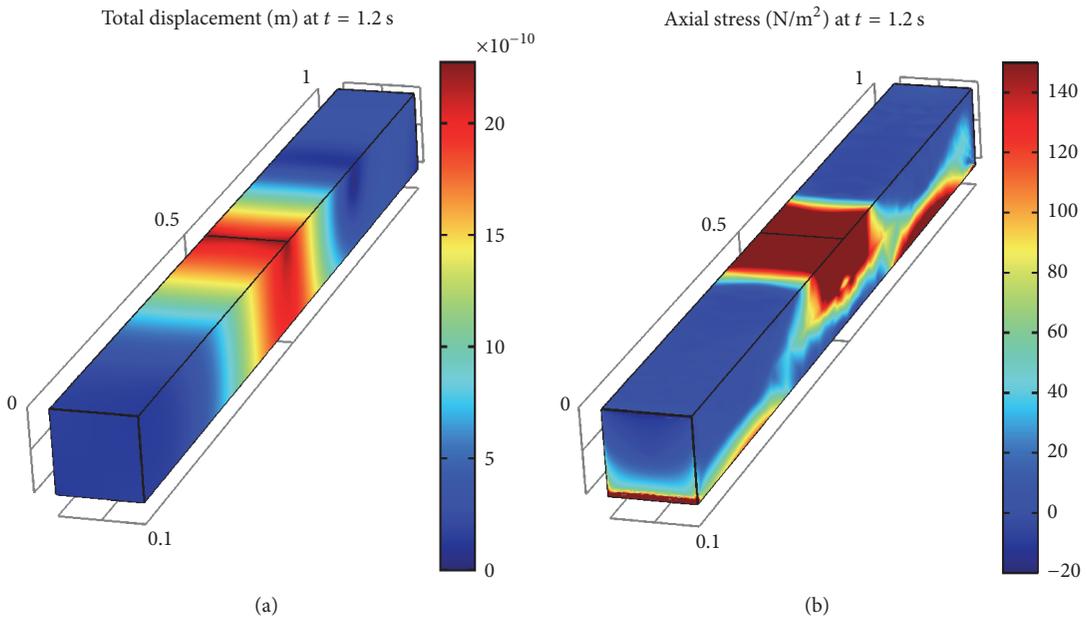


FIGURE 4: Controlled total displacement (a) and axial stress of the beam at  $t = T$ : switching control.

where  $d$  is a positive constant, arising in fluid mechanics, nonlinear acoustics, traffic flow, etc. Using the well-known Hopf-Cole transformation

$$w = -2d \frac{1}{\Theta} \frac{\partial \Theta}{\partial x}, \quad (102)$$

it is reduced to the linear heat equation

$$\frac{\partial \Theta}{\partial t} = d \frac{\partial^2 \Theta}{\partial x^2}. \quad (103)$$

The controllability of the Burgers' equation is studied separately by many authors, e.g., in [43–46]. See also the related references therein.

Our aim is to determine a distributed control regime

$$u \in \mathcal{U} = \{u \in L^\infty [0, T], |u| \leq \epsilon, \text{supp}(u) \subseteq [0, T]\}, \quad (104)$$

governing the heat equation

$$\frac{\partial \Theta}{\partial t} = d \frac{\partial^2 \Theta}{\partial x^2} + u(t) v(x) \quad \text{in } \mathbb{R} \times \mathbb{R}^+, \quad (105)$$

with the aim of providing at a given instant  $T$  the terminal condition

$$w(x, T) = w_T(x) \quad \text{in } \mathbb{R}, \quad (106)$$

exactly or approximately for the viscous Burgers' flow (102). Here  $v$  is a given nonnegative function, and  $w_T \in L^\infty(\mathbb{R})$  is a given function.

In terms of the residue

$$\mathcal{R}_T(u) = \|w(x, T) - w_T(x)\|_{L^\infty(\mathbb{R})}, \quad (107)$$

the problem is to determine admissible controls  $u \in \mathcal{U}$  ensuring at the given instant  $T$  the equality

$$\mathcal{R}_T(u) = 0, \quad (108)$$

or the inequality

$$\mathcal{R}_T(u) \leq \varepsilon \quad (109)$$

with a given accuracy  $\varepsilon > 0$ .

From (102) it becomes obvious that as soon as

$$\widetilde{\mathcal{R}}_T(u) = \|\Theta(x, T) - \Theta_T(x)\|_{L^\infty(\mathbb{R})} = 0 \quad (110)$$

holds with

$$\Theta_T(x) = \exp\left[-\frac{1}{2d} \int_{-\infty}^x w_T(\xi) d\xi\right], \quad (111)$$

then (108) holds as well. Similarly, it is shown that (108) implies (110) as a particular case. Therefore, in the case of exact controllability, (108) and (110) are equivalent. This means that the controls satisfying (110) are resolving for (108).

Assume that  $v(x) = \chi_{[-1,1]}(x)$ . Then, on the basis of Green's function method [42], (110) is equivalent to the equality

$$\int_0^T G_v(x, T - \tau) u(\tau) d\tau = M_T(x) \quad \text{in } \mathbb{R}, \quad (112)$$

where

$$\begin{aligned} M_T(x) &= \Theta_T(x) \\ &- \int_{-\infty}^{\infty} \exp\left[-\frac{1}{2d} \int_{-\infty}^{\xi} w(\chi, 0) d\chi\right] G(x - \xi, t) d\xi, \\ G(x, t) &= \frac{1}{\sqrt{4\pi dt}} \exp\left[-\frac{x^2}{4dt}\right], \\ G_v(x, t) &= \int_{-\infty}^{\infty} v(\xi) G(x - \xi, t) d\xi \\ &= \frac{1}{2} \left[ \operatorname{erf}\left[\frac{x+1}{\sqrt{4dt}}\right] - \operatorname{erf}\left[\frac{x-1}{\sqrt{4dt}}\right] \right]. \end{aligned} \quad (113)$$

Therefore, (112) can be expanded into series of orthogonal functions as it was done above in Section 3.1.

Consider the initial state

$$w(x, 0) = w_0(x) \quad \text{in } \mathbb{R}, \quad (114)$$

for a given  $w_0 \in L^\infty(\mathbb{R})$ . Let  $w_0$  and  $w_T$  be positive. Then, both  $G_v$  and  $M_T$  have exponential decay in  $x$  meaning that we can consider (112) in  $[-l, l]$ ,  $l > 1$ . Expanding  $G_v$  and  $M_T$  into Fourier series and taking into account the fact that both functions are even, we obtain

$$\begin{aligned} M_T(x) &= \frac{1}{2} M_{T0} + \sum_{n=1}^{\infty} M_{Tn} \cos\left(\frac{\pi n}{l} x\right), \\ M_{Tn} &= \frac{2}{l} \int_0^l M_T(x) \cos\left(\frac{\pi n}{l} x\right) dx, \end{aligned} \quad (115)$$

$$G_v(x, t) = \frac{1}{2} G_{v0}(t) + \sum_{n=1}^{\infty} G_{vn}(t) \cos\left(\frac{\pi n}{l} x\right),$$

$$G_{vn}(t) = \frac{2}{l} \int_0^l G_v(x, t) \cos\left(\frac{\pi n}{l} x\right) dx.$$

The resolving system is formed by equating the coefficients of cosines' for corresponding  $n$  in both sides of the equation above:

$$\begin{aligned} \int_0^T G_{v0}(T - \tau) u(\tau) d\tau &= M_{T0}, \\ \int_0^T G_{vn}(T - \tau) u(\tau) d\tau &= M_{Tn}, \quad n \in \mathbb{N}. \end{aligned} \quad (116)$$

As in Section 3.1, truncate system (116) by a finite  $N$ . In order to derive a consistent system of algebraic equations, consider

$$u(t) = \frac{1}{2} u_0 + \sum_{n=1}^N u_n \cos\left(\frac{\pi n}{T} t\right), \quad (117)$$

where  $u_n$  are unknown constants that need to be determined. Substituting  $u$  into the resolving system, the following system of linear algebraic equations is derived:

$$\mathbf{A} \mathbf{u} = \Theta_T, \quad (118)$$

where

$$\begin{aligned} \mathbf{A} &= \{A_{mn}\}_{m,n=0}^N, \\ A_{mn} &= \int_0^T G_{dm}(T - \tau) \cos\left(\frac{\pi n}{T} \tau\right) d\tau, \\ \mathbf{u} &= (u_0 \ \cdots \ u_N)^T, \\ \Theta_T &= (\Theta_{T0} \ \cdots \ \Theta_{TN})^T. \end{aligned} \quad (119)$$

*Remark 6.* The unknown coefficients can be found from the linear system (118) efficiently. Nevertheless, because of truncation, the control (117) will ensure only approximate controllability of the Burger's equation.

Now let us consider the case of approximate controllability. Obviously, the equivalency of exact controllability established above does not occur here. Therefore, we need to verify (109) directly.

Consider a flow governed by (102) in a thin infinite layer. Assume that the flow source is located at  $x = -1$  section of the layer and generates a harmonic flux

$$\varphi(t) = [\theta(t - 0) - \theta(t - 1)] \exp(i\pi t), \quad (120)$$

which vanishes when  $t > 1$ . Therefore, the Burgers' equation must be complemented by the condition

$$w(-1, t) = \begin{cases} \varphi(t), & t \in [0, 1], \\ 0, & \text{else.} \end{cases} \quad (121)$$

Let in particular

$$v(x) = \text{rect}(2x). \quad (122)$$

The problem is to find such an admissible control  $u \in \mathcal{U}$  that ensures

$$\mathcal{R}_T(u) = \|w(x, T)\|_{L^\infty[-1,1]} \leq \varepsilon \quad (123)$$

with  $\varepsilon = 10^{-4}$ . Obviously, the desired  $T$  must satisfy  $T < T_0$ , where  $T_0$  satisfies

$$\mathcal{R}_{T_0}(0) \leq \varepsilon. \quad (124)$$

Restricting the consideration by  $d = 1$ , it is obtained that  $T_0 = 10.35$  (see Figure 5).

Involving the distributed control regime (42), it is possible to achieve the approximate null-controllability of the flow for  $T = 9.2$  when  $u^o = -1$ ,  $\omega = 2\pi$ , and  $\gamma = 0$  (see Figure 6). Furthermore, it is evident from Figure 7 that when  $u^o = 1$ ,  $\omega = \pi$ , and  $\gamma = -\pi/2$ , the flow is approximately null-controllable at  $T = 7.15$ .

On the other hand, the same accuracy can be achieved using a single impulsive action

$$u(t) = u^o \delta(t - t^o), \quad (125)$$

where  $0 < t^o < T_0$ . Indeed, Figure 8 shows that when  $u^o = 1$  and  $t^o = 1$ , the flow is approximately null-controllable at  $T = 8.5$ .

Furthermore, involving the switching regime (53), it is possible to achieve approximate null-controllability of the flow for less  $T$ . Specifically, choosing

$$u(t) = 0.1 [\theta(t - 0) - \theta(t - 1.2)], \quad (126)$$

approximate null-controllability is achieved for  $T = 6$  (see Figure 9).

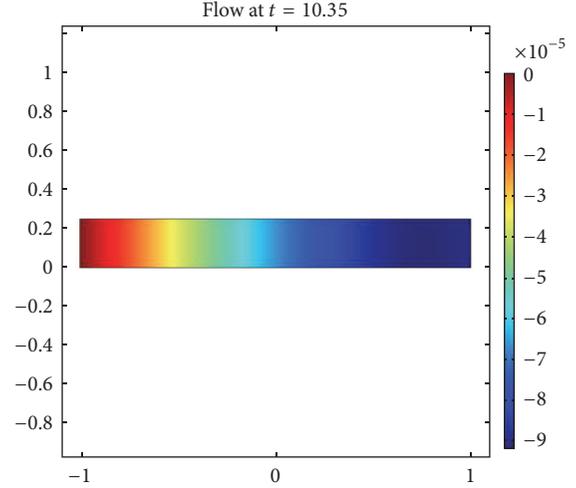


FIGURE 5: Uncontrolled flow in the layer at  $t = 10.35$ .

## 6. Conclusions

A systematic technique for deriving resolving controls for exact and approximate controllability analysis of systems with nonlinear state constraints heuristically is proposed. Representing the solution of the state constraints in terms of Green's function, the exact and approximate controllability are reduced to nonlinear integral type constraints. The exact solution of the constraints is highly complicated, so that numerical schemes are usually applied. The proposed technique allows representing some of the resolving controls explicitly and provides nonlinear algebraic constraints on the unknown coefficients. The hierarchy of the heuristic controls includes (i) polynomial, (ii) time-harmonic, (iii) switching or piecewise-continuous, (iv) optimal stopping, and (v) impulsive regimes. The corresponding constraints are derived straightforwardly. The derived regimes show a good correspondence with the  $L^p$ -optimal solutions,  $p = 1, 2, \infty$ , of linear moments problem derived earlier rigorously. The technique is checked on two specific examples: elastic finite beam bent under external force with uncertainty and a viscous flow governed by Burger's nonlinear equation. The technique is efficiently applied in both cases. Numerical simulations confirm theoretical predictions.

New studies about practical implementation of the technique in other problems of mechanical engineering are initiated. In proceeding papers we plan to report some possibilities of determination of resolving controls extremizing cost functionals that have specific importance in the context of the considered problem.

## Data Availability

No data were used to support this study.

## Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

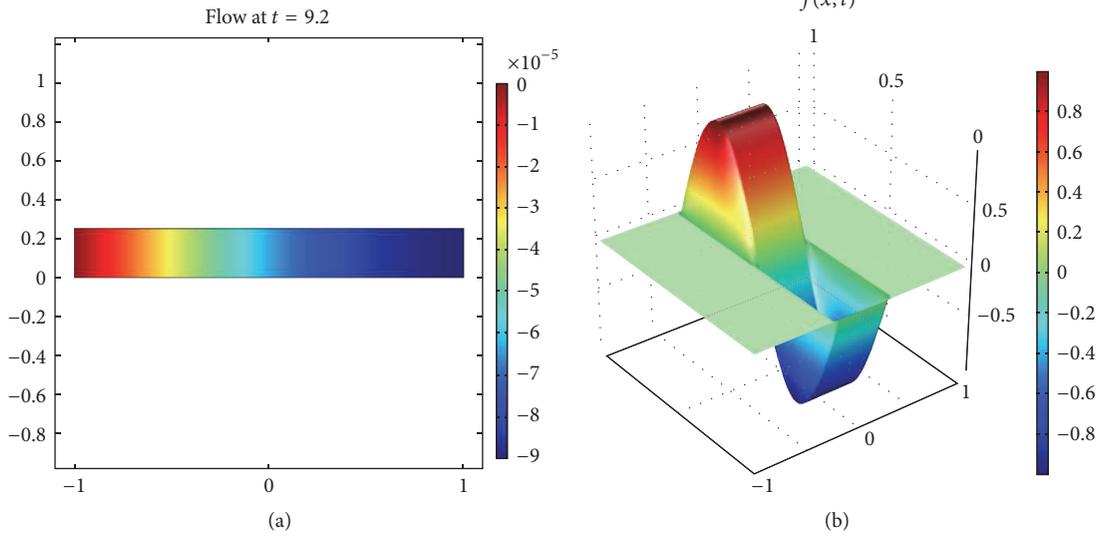


FIGURE 6: Flow in the layer at  $t = 9.2$  (a) and corresponding time-harmonic control regime (b).

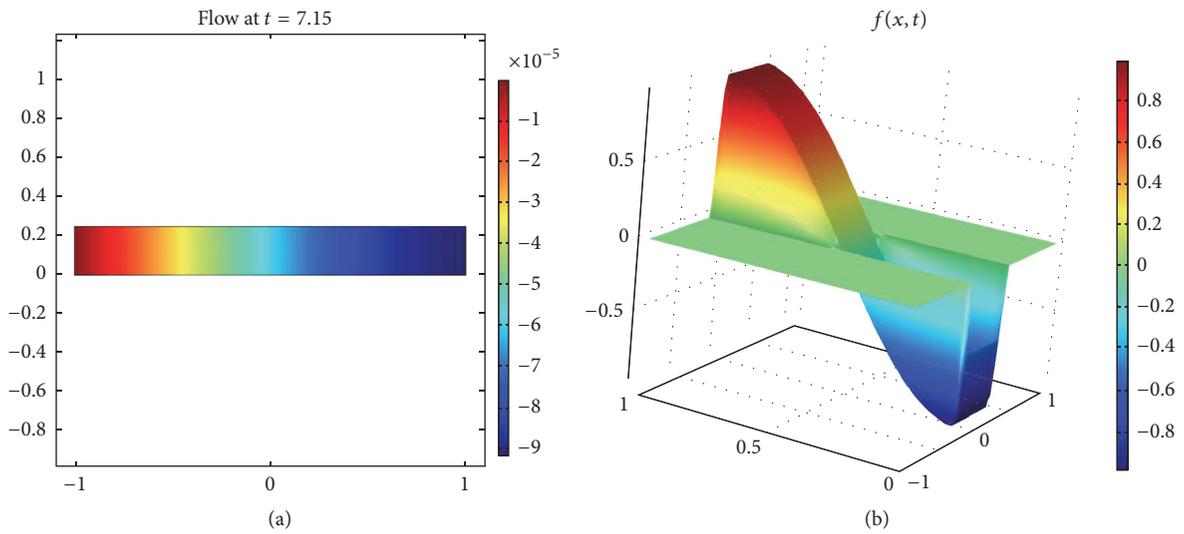


FIGURE 7: Flow in the layer at  $t = 7.15$  (a) and corresponding time-harmonic control regime (b).

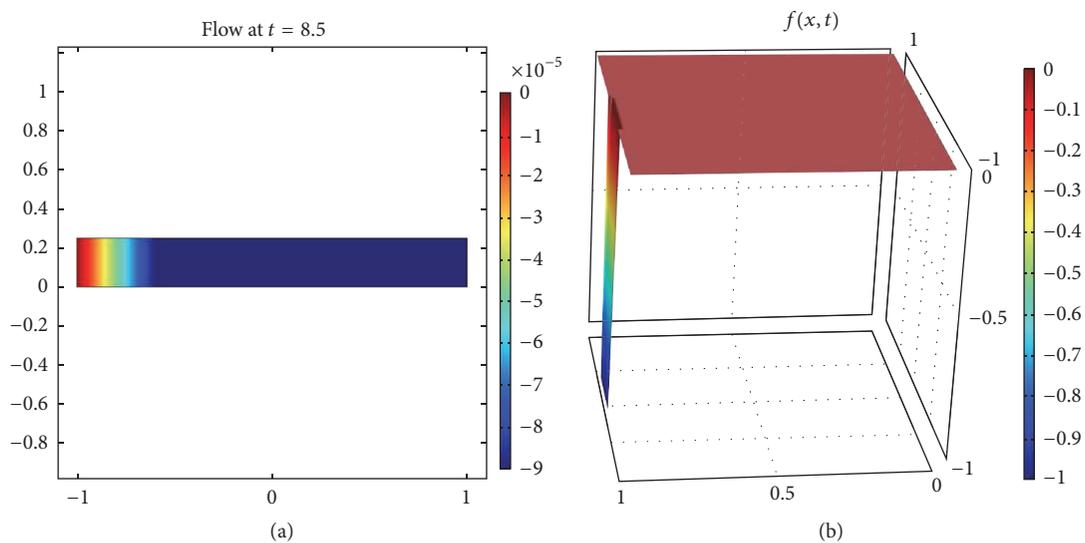


FIGURE 8: Flow in the layer at  $t = 8.5$  (a) and corresponding impulsive control regime (b).

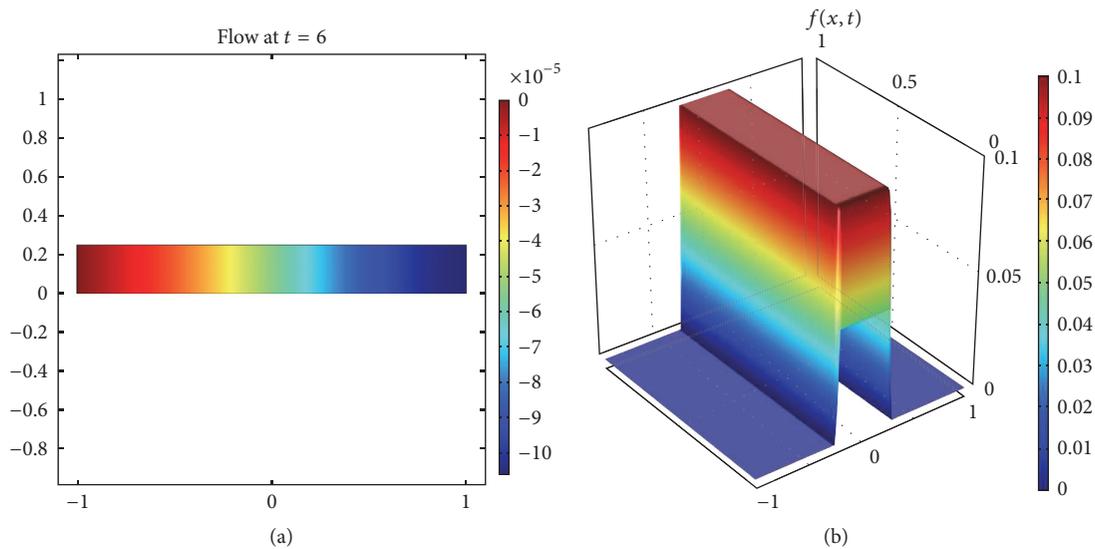


FIGURE 9: Flow in the layer at  $t = 8.5$  (a) and corresponding impulsive control regime (b).

## Acknowledgments

Some specific ideas of the manuscript were raised in intensive discussions with Professor Ara S. Avetisyan (Institute of Mechanics, National Academy of Sciences of Armenia), to whom the author is heartily thankful.

## References

- [1] V. Komornik, *Exact Controllability and Stabilization. The Multiplier Method*, RAM: Research in Applied Mathematics, John Wiley & Sons, Chichester, UK, 1994.
- [2] S. A. Avdonin and S. A. Ivanov, *Families of Exponentials. The Method of Moments in Controllability Problems for Distributed Parameter Systems*, Cambridge University Press, New York, NY, USA, 1995.
- [3] A. V. Fursikov and O. Y. Imanuvilov, *Controllability of Evolution Equations*, vol. 34 of *Lecture Notes Series*, Seoul National University, Research Institute of Mathematics, Global Analysis Research Center, Seoul, South Korea, 1996.
- [4] A. A. Agrachev and Y. L. Sachkov, *Control Theory from The Geometric Viewpoint*, vol. 87 of *Encyclopaedia of Mathematical Sciences*, Springer, Berlin, Germany, 2004.
- [5] E. Zuazua, "Controllability and observability of partial differential equations: some results and open problems," in *Handbook of Differential Equations: Evolutionary Equations*, vol. 3 of *Handb. Differ. Equ.*, pp. 527–621, Elsevier, North-Holland, Amsterdam, 2007.
- [6] R. Glowinski, J.-L. Lions, and J. He, *Exact and Approximate Controllability for Distributed Parameter Systems: A Numerical Approach*, Cambridge University Press, New York, NY, USA, 2008.
- [7] J. Lévine, *Analysis and Control of Nonlinear Systems: A Flatness-Based Approach*, Springer Science & Business Media, Berlin, Germany, 2009.
- [8] I. Lasiecka and R. Triggiani, *Control Theory for Partial Differential Equations: Volume 2, Abstract Hyperbolic-like Systems over a Finite Time Horizon*, Cambridge University Press, Cambridge, UK, 2011.
- [9] A. Y. Khapalov, *Mobile Point Sensors and Actuators in the Controllability Theory of Partial Differential Equations*, Springer, Cham, Switzerland, 2017.
- [10] A. S. Avetisyan and As. Zh. Khurshudyan, *ontrollability of Dynamic Systems: The Green's Function Approach*, Cambridge Scholars Publishing, Cambridge, UK, 2018.
- [11] K. Hiramoto, "Cooperative control method of active and Semiactive control: New framework for vibration control," *Mathematical Problems in Engineering*, vol. 2014, Article ID 518276, pp. 1–11, 2014.
- [12] L. Xin, Q. Wang, and Y. Li, "A new fast nonsingular terminal sliding mode control for a class of second-order uncertain systems," *Mathematical Problems in Engineering*, vol. 2016, Article ID 1743861, pp. 1–12, 2016.
- [13] B. Liu, Y. Jin, C. Chen, and H. Yang, "Speed control based on ESO for the pitching axis of satellite cameras," *Mathematical Problems in Engineering*, vol. 2016, Article ID 2138190, pp. 1–9, 2016.
- [14] A. Babiarczyk and M. Niezabitowski, "Controllability problem of fractional neutral systems: a survey," *Mathematical Problems in Engineering*, vol. 2017, Article ID 4715861, pp. 1–15, 2017.
- [15] M. Gugat, "Exact boundary controllability for free traffic flow with Lipschitz continuous state," *Mathematical Problems in Engineering*, vol. 2016, Article ID 2743251, pp. 1–11, 2016.
- [16] A. Qi, X. Ju, Q. Zhang, and Z. Chen, "Structural controllability of discrete-time linear control systems with time-delay: a delay node inserting approach," *Mathematical Problems in Engineering*, Article ID 1429164, pp. 1–9, 2016.
- [17] R. Li, Q. Wu, Q. Yang, and H. Ye, "Adaptive constrained control for uncertain nonlinear time-delay system with application to unmanned helicopter," *Mathematical Problems in Engineering*, vol. 2018, Article ID 8410360, pp. 1–11, 2018.
- [18] F. Cui, Y. Su, S. Xu, F. Liu, and G. Yao, "Optimization of the physical and mechanical properties of a spline surface fabricated by high-speed cold roll beating based on taguchi theory," *Mathematical Problems in Engineering*, vol. 2018, Article ID 8068362, pp. 1–12, 2018.
- [19] P. Xia, Y. Deng, Z. Wang, and H. Li, "Speed adaptive sliding mode control with an extended state observer for permanent

- magnet synchronous motor,” *Mathematical Problems in Engineering*, vol. 2018, Article ID 6405923, pp. 1–13, 2018.
- [20] E. Mallea-Zepeda, E. Lenes, and E. Valero, “Boundary control problem for heat convection equations with slip boundary condition,” *Mathematical Problems in Engineering*, vol. 2018, Article ID 7959761, pp. 1–14, 2018.
- [21] R. Triggiani, “Lack of exact controllability for wave and plate equations with finitely many boundary controls,” *Differential and Integral Equations: International Journal for Theory and Applications*, vol. 4, no. 4, pp. 683–705, 1991.
- [22] S. Micu and E. Zuazua, “Null controllability of the heat equation in unbounded domains,” in *Unsolved Problems in Mathematical Systems and Control Theory*, V. D. Blondel and A. Megretski, Eds., Princeton University Press, Princeton, NJ, USA, 2004.
- [23] A. Shirikyan, “Euler equations are not exactly controllable by a finite-dimensional external force,” *Physica D: Nonlinear Phenomena*, vol. 237, no. 10–12, pp. 1317–1323, 2008.
- [24] S. Guerrero and O. Y. Imanuvilov, “Remarks on non controllability of the heat equation with memory,” *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 19, no. 1, pp. 288–300, 2013.
- [25] A. Halanay and L. Pandolfi, “Approximate controllability and lack of controllability to zero of the heat equation with memory,” *Journal of Mathematical Analysis and Applications*, vol. 425, no. 1, pp. 194–211, 2015.
- [26] G. Adomian, *Solving Frontier Problems of Physics: The Decomposition Method*, Kluwer Academic, Dordrecht, The Netherlands, 1994.
- [27] S. J. Liao, *Homotopy Analysis Method in Nonlinear Differential Equations*, Springer, Higher Education Press, Berlin, Germany, 2012.
- [28] M. Frasca, “Green function method for nonlinear systems,” *Modern Physics Letters A*, vol. 22, no. 18, pp. 1293–1299, 2007.
- [29] A. Z. Khurshudyan, “New Green’s functions for some nonlinear oscillating systems and related PDEs,” *International Journal of Modern Physics C*, vol. 29, Article ID 1850032, 9 pages, 2018.
- [30] A. D. Polyanin and A. V. Manzhirov, *Handbook of Integral Equations*, Chapman & Hall/CRC, Boca Raton, Fla, USA, 2nd edition, 2008.
- [31] A.-M. Wazwaz, *Linear and Nonlinear Integral Equations Methods and Applications*, Springer, Heidelberg, Germany, 2011.
- [32] S. Linge and H. P. Langtangen, “Solving nonlinear algebraic equations,” in *Programming for Computations - MATLAB/Octave*, vol. 14 of *Texts in Computational Science and Engineering*, Springer, 2016.
- [33] R. E. White, *Computational Mathematics: Models, Methods, and Analysis with MATLAB and MPI*, Chapman & Hall/CRC Press, Boca Raton, Fla, USA, 2nd edition, 2016.
- [34] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, vol. 116 of *International Series in Operations Research & Management Science*, Springer, New York, NY, USA, 3rd edition, 2008.
- [35] V. I. Utkin, *Sliding Modes in Control Optimization*, Communications and Control Engineering, Springer, Berlin, Germany, 1991.
- [36] N. N. Krasovskii, *Motion Control Theory*, Nauka, Moscow, Russia, 1968.
- [37] V. A. Kubyshkin, “On solutions of a finite dimensional nonlinear problem of moments for control,” *Automation and Remote Control*, vol. 45, no. 3, pp. 298–308, 1984.
- [38] A. G. Butkovskii, “Some problems of control of the distributed-parameter systems,” *Automation and Remote Control*, vol. 72, no. 6, pp. 1237–1241, 2011.
- [39] As. Zh. Khurshudyan, *Structural optimization and control of vibrations of deformable systems with variable parameters [Ph.D. thesis]*, Institute of Mechanics, National Academy of Sciences of Armenia, 2015.
- [40] Sh. Kh. Arakelyan and As. Zh. Khurshudyan, “The Bubnov-Galerkin procedure for solving mobile control problems for systems with distributed parameters,” *Institute of Mechanics - National Academy of Sciences - Armenia*, vol. 68, no. 3, pp. 54–75, 2015.
- [41] J. Lian and S. H. Žak, “Control of Uncertain Systems,” in *Springer Handbook of Automation*, S. Y. Nof, Ed., Springer, Berlin, Germany, 2009.
- [42] A. D. Polyanin and V. E. Nazaikinskii, *Handbook of Linear Partial Differential Equations for Engineers and Scientists*, Chapman & Hall/CRC Press, Boca Raton, Fla, USA, 2nd edition, 2016.
- [43] C. I. Byrne, D. S. Gilliam, and V. I. Shubov, “Boundary control for a viscous Burgers’ equation,” in *Identification Control for Systems Governed by Partial Differential Equations*, H. T. Banks, R. H. Fabiano, and K. Ito, Eds., pp. 171–185, SIAM, Philadelphia, Pa, USA, 1993.
- [44] J.-M. Lellouche, J.-L. Devenon, and I. Dekeyser, “Boundary control of Burgers’ equation—a numerical approach,” *Computers & Mathematics with Applications*, vol. 28, no. 5, pp. 33–44, 1994.
- [45] H. V. Ly, K. D. Mease, and E. S. Titi, “Distributed and boundary control of the viscous Burgers’ equation,” *Numerical Functional Analysis and Optimization*, vol. 18, no. 1–2, pp. 143–188, 1997.
- [46] A. Shirikyan, *Control Theory for the Burgers equation: Agrachev-Sarychev Approach*, University of Iași, 2010.

## Research Article

# An Adaptive Gradient Projection Algorithm for Piecewise Convex Optimization and Its Application in Compressed Spectrum Sensing

Tianjing Wang <sup>1</sup>, Hang Shen,<sup>2</sup> Xiaomei Zhu,<sup>2</sup> Guoqing Liu,<sup>1</sup> and Hua Jiang<sup>1</sup>

<sup>1</sup>School of Physical and Mathematical Science, Nanjing Tech University, Nanjing 211816, China

<sup>2</sup>College of Computer Science and Technology, Nanjing Tech University, Nanjing 211816, China

Correspondence should be addressed to Tianjing Wang; wangtianjing@njtech.edu.cn

Received 18 November 2017; Revised 20 March 2018; Accepted 2 April 2018; Published 14 May 2018

Academic Editor: Paula L. Zabala

Copyright © 2018 Tianjing Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Signal sparse representation has attracted much attention in a wide range of application fields. A central aim of signal sparse representation is to find a sparse solution with the fewest nonzero entries from an underdetermined linear system, which leads to various optimization problems. In this paper, we propose an Adaptive Gradient Projection (AGP) algorithm to solve the piecewise convex optimization in signal sparse representation. To find a sparser solution, AGP provides an adaptive stepsize to move the iteration solution out of the attraction basin of a suboptimal sparse solution and enter the attraction basin of a sparser solution. Theoretical analyses are used to show its fast convergence property. The experimental results of real-world applications in compressed spectrum sensing show that AGP outperforms the traditional detection algorithms in low signal-to-noise-ratio environments.

## 1. Introduction

The marked advances in signal processing in recent years have been driven by the emergence of new signal models and their applications. Signal sparse representation is an effective model for solving real-world problems, such as brain signal processing [1], face recognition [2], compressed spectrum sensing [3], and singing voice separation [4].

Given a signal  $y \in R^m$ , signal sparse representation aims to identify the sparsest solution  $x \in R^n$  from an underdetermined linear system  $y = Ax$ , where  $A \in R^{m \times n}$  is a full row rank matrix. The sparsity of a solution can be measured by  $l_0$ -norm, which leads to the following optimization problem

$$\begin{aligned} \min_x \quad & \|x\|_0 \\ \text{s.t.} \quad & y = Ax. \end{aligned} \quad (1)$$

Unfortunately, problem (1) is NP-hard [5]. Many methods (e.g., the greedy algorithm [6], the  $l_1$ -norm minimization [7], the  $l_p$ -norm ( $0 < p < 1$ ) minimization [8, 9], and

the Bayesian method [10]) have been used to find sparse solutions. Because  $l_p$ -norm is the most effective measurement of the sparsity, some researchers are interested in  $l_p$ -norm minimization

$$\begin{aligned} \min_x \quad & \|x\|_p^p \\ \text{s.t.} \quad & y = Ax. \end{aligned} \quad (2)$$

The piecewise convex optimization (2) can be solved using the existing algorithms, including the Focal Underdetermined System Solver (FOCUSS) [11], the Affine Scaling Transformation (AST) method [12], the Iteratively Reweighted  $l_1$  minimization (IRL1) [13], and the Iteratively Thresholding Method (ITM) [14]. The solutions they obtain, however, may be suboptimal sparse solutions.

In this paper, we propose a novel Adaptive Gradient Projection (AGP) algorithm for the piecewise convex optimization (2). This algorithm moves the iteration solution out of the attraction basin of a suboptimal sparse solution and finds a sparser solution in another attraction basin. The convergence analysis reveals that AGP performs better than

AST when finding the global optimal sparse solution. The experimental results show that the detection performances of compressed spectrum sensing based on AGP are greatly improved compared to other algorithms.

The remainder of the paper is organized as follows. In Section 2, we derive an Adaptive Gradient Projection algorithm that can find a sparser solution than AST. Section 3 presents the application of AGP to compressed spectrum sensing. The detection performances based on AGP are compared to the traditional spectrum sensing method. Finally, conclusions are presented in Section 4.

## 2. Adaptive Gradient Projection Algorithm for Piecewise Convex Optimization

*2.1. Description of Affine Scaling Transformation Method.* For convenience, problem (2) can be rewritten as

$$\begin{aligned} \min_x \quad & E^{(p)}(x) = \sum_{i=1}^n |x(i)|^p \\ \text{s.t.} \quad & y = Ax. \end{aligned} \quad (3)$$

Note that the objective function is nondifferentiable with zero components. AST uses an affine scaling transformation to solve problem (3). For the  $(k+1)$ th iteration, it defines a symmetric scaling matrix  $W_{k+1} = \text{diag}(|x(i)|^{1-p/2})$  and a scaled variable

$$q = (W_{k+1})^{-1} x, \quad \text{equivalently } x = W_{k+1}q. \quad (4)$$

Thus, problem (3) in  $x$  is transformed to the problem in  $q$

$$\begin{aligned} \min_q \quad & E^{(p)}(W_{k+1}q) \\ \text{s.t.} \quad & y = A_{k+1}q, \end{aligned} \quad (5)$$

where  $A_{k+1} = AW_{k+1}$ .

Given a search direction  $l_k = p(I - A_{k+1}^+ A_{k+1})q_k$ , the new solution  $x_{k+1}$  is

$$x_{k+1} = W_{k+1}q_{k+1} = x_k - \mu_k W_{k+1}l_k, \quad (6)$$

where  $I \in R^{n \times n}$  is an identity matrix,  $A_{k+1}^+ = A_{k+1}^T (A_{k+1} A_{k+1}^T)^{-1}$  is a Moore-Penrose pseudoinverse matrix, and  $\mu_k$  is a stepsize.

Using a fixed stepsize  $\mu_k = 1/p$ , AST is summarized as

$$W_{k+1} = \text{diag}(|x_k(i)|^{1-p/2}), \quad (7)$$

$$q_{k+1} = A_{k+1}^+ y, \quad \text{where } A_{k+1} = AW_{k+1}, \quad (8)$$

$$x_{k+1} = W_{k+1}q_{k+1}. \quad (9)$$

The convergence theorem of AST is as follows.

**Theorem 1.** *Starting from an initial point  $x_0$ , AST generates a sequence  $\{x_k\}_{k=1}^{\infty}$  converging to a sparse solution  $x^*$  of problem (3).*

From (9), we see that some small entries of iteration solution converge to zero, because they are sequentially compressed by the scaling elements in  $W_{k+1}$ . Thus, a sequence of iteration solutions of AST will converge to a sparse solution  $x^*$ , which may be close to  $x_0$ . However, this solution may be not the sparsest solution of problem (3). Making the iteration solution enter the attraction basin of other sparse solution is very important to reduce the effect of the initial point. Furthermore, Theorem 1 shows that AST obtains  $x^*$  within an infinite number of iterations, which affects the convergence rate. How to enhance the convergence speed of AST is another problem to be solved.

*2.2. Derivation of Adaptive Gradient Projection Algorithm.* To solve the above two problems, we first consider the convergence process of AST if an iteration solution has some zero entries.

**Lemma 2.** *Given a block matrix  $P = \begin{pmatrix} P_1 & P_2 \\ P_3 & P_4 \end{pmatrix}$ , we get*

$$P^{-1} = \begin{pmatrix} -U^{-1}P_4P_2^{-1} & U^{-1} \\ P_2^{-1} + P_2^{-1}P_1U^{-1}P_4P_2^{-1} & -P_2^{-1}P_1U^{-1} \end{pmatrix}, \quad (10)$$

where  $U = P_3 - P_4P_2^{-1}P_1$  is a Schur complement.

**Lemma 3.** *If  $x_k$  with  $S$  zero entries is a solution of problem (3), then these zero entries will not change in the remaining iterations.*

For simplicity, let the front  $S$  entries of  $x_k$  be zero (i.e.,  $x_k = (x_k^1, x_k^2)^T = (0, x_k^2)^T$ ), where  $x_k^1 = (x_k(1), \dots, x_k(S))^T$ ,  $x_k^2 = (x_k(S+1), \dots, x_k(n))^T$ . Then,  $x_{k+1}$  can be computed as follows:

$$\begin{aligned} x_{k+1} &= W_{k+1}q_{k+1} = W_{k+1}A_{k+1}^+ Ax_k \\ &= W_{k+1}(AW_{k+1})^+ Ax_k, \end{aligned} \quad (11)$$

where  $W_{k+1} = \text{diag}(W_{k+1}^1, W_{k+1}^2) = \text{diag}(0, W_{k+1}^2)$ . Partitioning  $A$ , we calculate  $AW_{k+1}$  in (11) to be

$$\begin{aligned} AW_{k+1} &= \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & W_{k+1}^2 \end{pmatrix} \\ &= \begin{pmatrix} 0 & A_{12}W_{k+1}^2 \\ 0 & A_{22}W_{k+1}^2 \end{pmatrix}, \end{aligned} \quad (12)$$

and its Moore-Penrose pseudoinverse matrix is

$$(AW_{k+1})^+ = (AW_{k+1})^T (AW_{k+1} (AW_{k+1})^T)^{-1}, \quad (13)$$

where

$$\begin{aligned} &(AW_{k+1} (AW_{k+1})^T)^{-1} \\ &= \begin{pmatrix} A_{12} (W_{k+1}^2)^2 A_{12}^T & A_{12} (W_{k+1}^2)^2 A_{22}^T \\ A_{22} (W_{k+1}^2)^2 A_{12}^T & A_{22} (W_{k+1}^2)^2 A_{22}^T \end{pmatrix}^{-1}. \end{aligned} \quad (14)$$

Because  $AW_{k+1}$  is full row rank,  $AW_{k+1}(AW_{k+1})^T$  is also full rank, which has an invertible submatrix. For convenience, assume that  $A_{12}(W_{k+1}^2)^2A_{22}^T$  is an invertible submatrix. Otherwise, we partition  $(AW_{k+1}(AW_{k+1})^T)^{-1}$  to get an invertible submatrix according to Lemma 2. Defining  $P_1 = A_{12}(W_{k+1}^2)^2A_{22}^T$ ,  $P_2 = A_{12}(W_{k+1}^2)^2A_{22}^T$ ,  $P_3 = A_{22}(W_{k+1}^2)^2A_{12}^T$ ,  $P_4 = A_{22}(W_{k+1}^2)^2A_{22}^T$ , we get

$$(AW_{k+1}(AW_{k+1})^T)^{-1} = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}, \quad (15)$$

where  $U = P_3 - P_4P_2^{-1}P_1$ ,  $B_{11} = -U^{-1}P_4P_2^{-1}$ ,  $B_{22} = -P_2^{-1}P_1U^{-1}$ ,  $B_{12} = U^{-1}$ ,  $B_{21} = P_2^{-1} + P_2^{-1}P_1U^{-1}P_4P_2^{-1}$ .

Substituting (12) and (15) into (13), we have

$$\begin{aligned} & (AW_{k+1})^+ \\ &= \begin{pmatrix} 0 & 0 \\ (A_{12}W_{k+1}^2)^T & (A_{22}W_{k+1}^2)^T \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 \\ C_{21} & C_{22} \end{pmatrix}, \end{aligned} \quad (16)$$

where  $C_{21} = (A_{12}W_{k+1}^2)^TB_{11} + (A_{22}W_{k+1}^2)^TB_{21}$ ,  $C_{22} = (A_{12}W_{k+1}^2)^TB_{12} + (A_{22}W_{k+1}^2)^TB_{22}$ . Equation (11) then becomes

$$\begin{aligned} x_{k+1} &= W_{k+1}(AW_{k+1})^+ Ax_k \\ &= \begin{pmatrix} 0 & 0 \\ 0 & W_{k+1}^2 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ C_{21} & C_{22} \end{pmatrix} \begin{pmatrix} A_{12}x_k^2 \\ A_{22}x_k^2 \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ W_{k+1}^2 (C_{21}A_{12} + C_{22}A_{22}) x_k^2 \end{pmatrix}. \end{aligned} \quad (17)$$

The front  $S$  entries of  $x_{k+1}$  in (17) are still zero, and the reverse cannot occur in the remaining iterations.

Lemma 3 implies that an unknown sparse solution can be identified in smaller and smaller subspace. It motivates us to accelerate convergence by sequentially shrinking the solving range. Meanwhile, the choice of subspace cannot be limited by the initial point; that is, the iteration solution is able to enter the subspace that does not contain the initial point. AST chooses a fixed stepsize, so it cannot move the iteration solution from one octant to another distant octant. These solutions concentrate in the attraction basin of the suboptimal sparse solution. Moving the iteration solutions out of the current attraction basin is a goal of the Adaptive Gradient Projection algorithm.

To find the search direction at an iteration solution  $x$ , we define the gradient

$$g = \nabla_x E^{(p)}(x) = \left( \frac{\partial E^{(p)}(x)}{\partial x(1)}, \dots, \frac{\partial E^{(p)}(x)}{\partial x(n)} \right)^T, \quad (18)$$

where

$$\frac{\partial E^{(p)}(x)}{\partial x(i)} = \begin{cases} p|x(i)|^{p-2}x(i), & \text{if } x(i) \neq 0 \\ \text{delete}, & \text{if } x(i) = 0, \end{cases} \quad (19)$$

and delete represents that the partial derivative is not computed as  $x(i) = 0$ , and the corresponding component is deleted from  $g$ . Definition (18) avoids the nondifferentiable problem of  $l_p$ -norm minimization such that the gradient can be calculated in subspace. Then, we provide the algorithm derivation below.

Initially, we reset an initial point  $x_0 = A^+y$  as

$$x_{D_0}(j) = \begin{cases} x_0(i), & \text{if } x_0(i) \neq 0 \\ \text{delete}, & \text{if } x_0(i) = 0, \end{cases} \quad (20)$$

where the index set  $D_0 = (\alpha_1, \dots, \alpha_{j_0})$  records the locations of entries as  $x_0(i) \neq 0$ . For example, if  $x_0 = (0.2, -0.1, 0.5, 0, -0.2)^T$ , then  $D_0 = (1, 2, 3, 5)$  and  $x_{D_0} = (0.2, -0.1, 0.5, -0.2)^T$ .

For the  $(k+1)$ th iteration, let  $\bar{W}_{k+1} = \text{diag}(|x_{D_k}(i)|^{1-p/2})$  and  $\bar{A}_{k+1} = A_{D_k}\bar{W}_{k+1}$ ; the gradient  $\bar{g}_k$  and the search direction  $\bar{l}_k$  are

$$\begin{aligned} \bar{g}_k &= \left( \frac{\partial E^{(p)}(x_{D_k})}{\partial x_{D_k}(1)}, \dots, \frac{\partial E^{(p)}(x_{D_k})}{\partial x_{D_k}(J_k)} \right)^T, \\ \bar{l}_k &= (I_k - \bar{A}_{k+1}^+ \bar{A}_{k+1}) \bar{g}_k, \end{aligned} \quad (21)$$

where  $D_k = (\alpha_1, \dots, \alpha_{J_k})$  records the locations of entries as  $x_{D_k}(j) \neq 0$  ( $j = 1, \dots, J_k$ ), the column vectors of  $A_{D_k}$  are selected from  $A$  due to  $D_k$ , and  $I_k \in R^{J_k \times J_k}$  is an identity matrix. In the span space of  $A_{D_k}$ , the new solution  $\bar{x}_{k+1}$  is defined due to (6):

$$\bar{x}_{k+1} = x_{D_k} - \mu_k \bar{W}_{k+1} \bar{l}_k, \quad (22)$$

where  $\mu_k$  is a stepsize.

The major challenge in solving problem (3) is to identify the most appropriate subspace where the sparse solution locates. Equation (22) states that  $\mu_k$  is important to find an appropriate subspace; thus, a function  $E^{(p)}(\bar{x}_{k+1})$  with respect to  $\mu$  is defined to investigate the property of the stepsize

$$F(\mu) = E^{(p)}(x_{D_k} - \mu \bar{W}_{k+1} \bar{l}_k). \quad (23)$$

Figure 1(b) shows that the piecewise convex function  $F(\mu)$  has nonunique minimum points that correspond to the zero entries of  $\bar{x}_{k+1}$ . We can compute these extreme points.

Without loss of generality, let every entry of  $\bar{x}_{k+1}$  equal to zero (i.e.,  $\bar{x}_{k+1}(j) = 0$ ). Then, the extreme point  $\mu_{k,j}$  is computed as

$$\mu_{k,j} = \frac{x_{D_k}(j)}{\bar{V}_{k+1}(j)}, \quad (24)$$

where  $j = 1, \dots, J_k$  and  $\bar{V}_{k+1}(j) = \bar{W}_{k+1} \bar{l}_k$ . An adaptive stepsize is chosen to find the minimal objective function value

$$\mu_k^* = \arg \min_{\mu_{k,j}} E^{(p)}(x_{D_k} - \mu_{k,j} \bar{W}_{k+1} \bar{l}_k), \quad (25)$$

and the new solution  $\bar{x}_{k+1}$  is written as

$$\bar{x}_{k+1} = x_{D_k} - \mu_k^* \bar{W}_{k+1} \bar{l}_k. \quad (26)$$

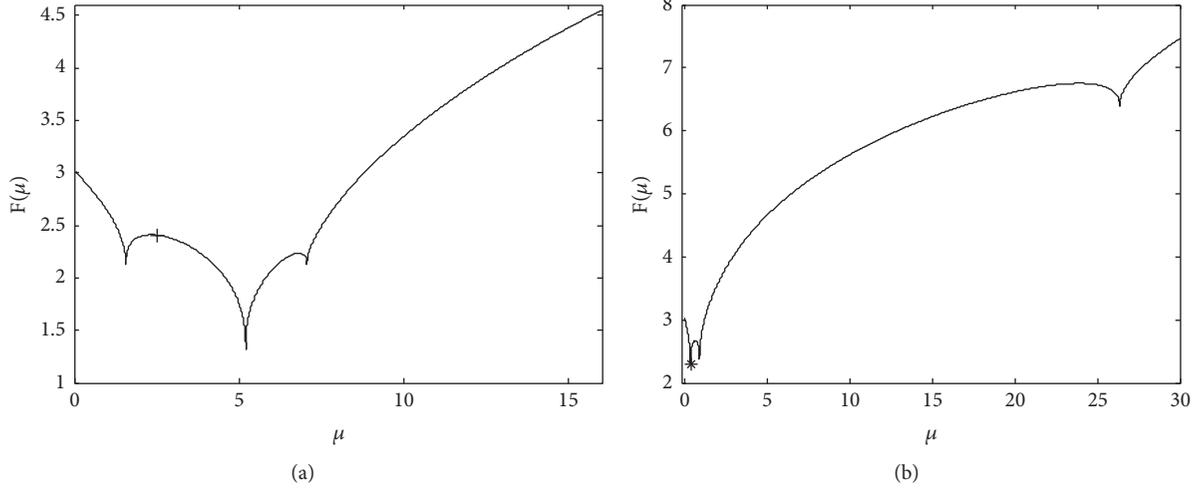


FIGURE 1: Function curves of  $F(\mu)$  with respect to  $\mu$ . (a) Fixed stepsize selected by AST, marked as “+.” (b) Adaptive stepsize selected by AGP, marked as “\*.”

The minimum point  $(\mu_k^*, E^{(p)}(\tilde{x}_{k+1}))$ , marked as “\*” in Figure 1(b), is determined by comparing three extreme values. Using the adaptive stepsize to obtain an iteration solution is beneficial to accelerate convergence. However, AST cannot obtain a minimum point at the search direction  $W_{k+1}\tilde{l}_k$  in (6), and the corresponding point  $(1/p, E^{(p)}(x_{k+1}))$  is marked as “+” in Figure 1(a). Using the fixed stepsize makes the iteration solutions gather in the adjacent region of the initial point. On the other hand, we set  $\tilde{x}_{k+1}(j) = 0$  as  $|\tilde{x}_{k+1}(j)| < \varepsilon$  ( $j = 1, \dots, J_k$ ), where  $\varepsilon$  is a threshold. AGP then determines multiple zero entries at each iteration so that it can quickly identify subspace where the sparse solution locates.

After some entries of  $\tilde{x}_{k+1}$  become zero, the set of their indices  $d_{k+1}$  is deleted from  $D_k$  (i.e.,  $D_{k+1} = D_k \setminus d_{k+1}$ ). We define  $x_{D_{k+1}} \leftarrow \tilde{x}_{k+1}(D_{k+1})$  that represents the entries of  $\tilde{x}_{k+1}$  located in  $D_{k+1}$  that are assigned to  $x_{D_{k+1}}$ . Thus, (26) can be restated as

$$x_{D_{k+1}} \leftarrow x_{D_k} - \mu_k^* \widetilde{W}_{k+1} \tilde{l}_{k+1}. \quad (27)$$

For example, if  $\tilde{x}_{k+1} = (0.3, 1 \times 10^{-6}, 0, 0, -0.1)^T$  and  $\varepsilon = 1 \times 10^{-4}$ , then  $D_{k+1} = (1, 5)$  and  $x_{D_{k+1}} = (0.3, -0.1)^T$ . The iterations are performed until the needed support set is determined.

Returning the final solution  $x_{D_L}$  to  $n$ -dimensional space is convenient to obtain the sparse solution  $x^*$ . By presetting  $x^* = 0 \in R^n$ , then  $x^*(D_L) \leftarrow x_{D_L}$  represents that the entries of  $x_{D_L}$  located in  $D_L$  that are assigned to  $x^*$ . For example, if  $x_{D_L} = (0.3, -0.1)^T$  and  $D_L = (1, 5)$ , then  $x^* = (0.3, 0, 0, 0, -0.1)^T$ .

As discussed above, AGP is summarized as follows.

**Algorithm 4** (Adaptive Gradient Projection Algorithm).

**Step 1** (initialization). Given a threshold  $\tau$ , compute an initial point  $x_0 = A^+ y$  and set an index set  $D_0 = (\alpha_1, \dots, \alpha_{j_0})$  and the iteration index  $k = 0$ .

**Step 2** (iteration)

**while** Stopping criterion not met **do**

    Compute  $\tilde{g}_k$  and  $\tilde{l}_k$  due to (21),

    Choose an adaptive stepsize by

$$\mu_k^* = \arg \min_{\mu_{k,j}} E^{(p)}(x_{D_k} - \mu_{k,j} \widetilde{W}_{k+1} \tilde{l}_k), \quad (28)$$

    Compute a solution  $\tilde{x}_{k+1} = x_{D_k} - \mu_k^* \widetilde{W}_{k+1} \tilde{l}_k$ ,

    Update the solution  $x_{D_{k+1}} \leftarrow x_{D_k} - \mu_k^* \widetilde{W}_{k+1} \tilde{l}_k$ ,

    Increase iteration index  $k$ .

**end while**  $|E^{(p)}(x_{D_{k+1}}) - E^{(p)}(x_{D_k})| < \tau$

**Step 3** (solution). An optimal sparse solution  $x^*(D_{k+1}) \leftarrow x_{D_{k+1}}$ .

**2.3. Convergence Analysis.** The convergence property of AGP is discussed as follows.

**Lemma 5.** *AGP can determine at least one zero entry at each iteration.*

Due to (26), there are three cases that AGP determines zero entry at each iteration. The first case is that one entry of  $x_{D_k}$  is set to zero, if  $x_{D_k}$  moves from one octant to the coordinate surface of another distant octant. In the second case, more than one entry of  $x_{D_k}$  become zero when the new iteration solution  $\tilde{x}_{k+1}$  exactly locates on the coordinate axis of another distant octant. In the third case, if  $\tilde{x}_{k+1}$  locates on the side of the coordinate axis of another distant octant, we set  $\tilde{x}_{k+1}(j) = 0$  as  $|\tilde{x}_{k+1}(j)| < \varepsilon$  ( $j = 1, \dots, J_k$ ). Therefore, AGP can determine at least one zero entry at each iteration.

**Theorem 6.** *Let  $x^* \in R^n$  with  $K$  nonzero entries be a sparse solution of problem (3). The number of iterations of AGP is no more than  $n - K$ .*

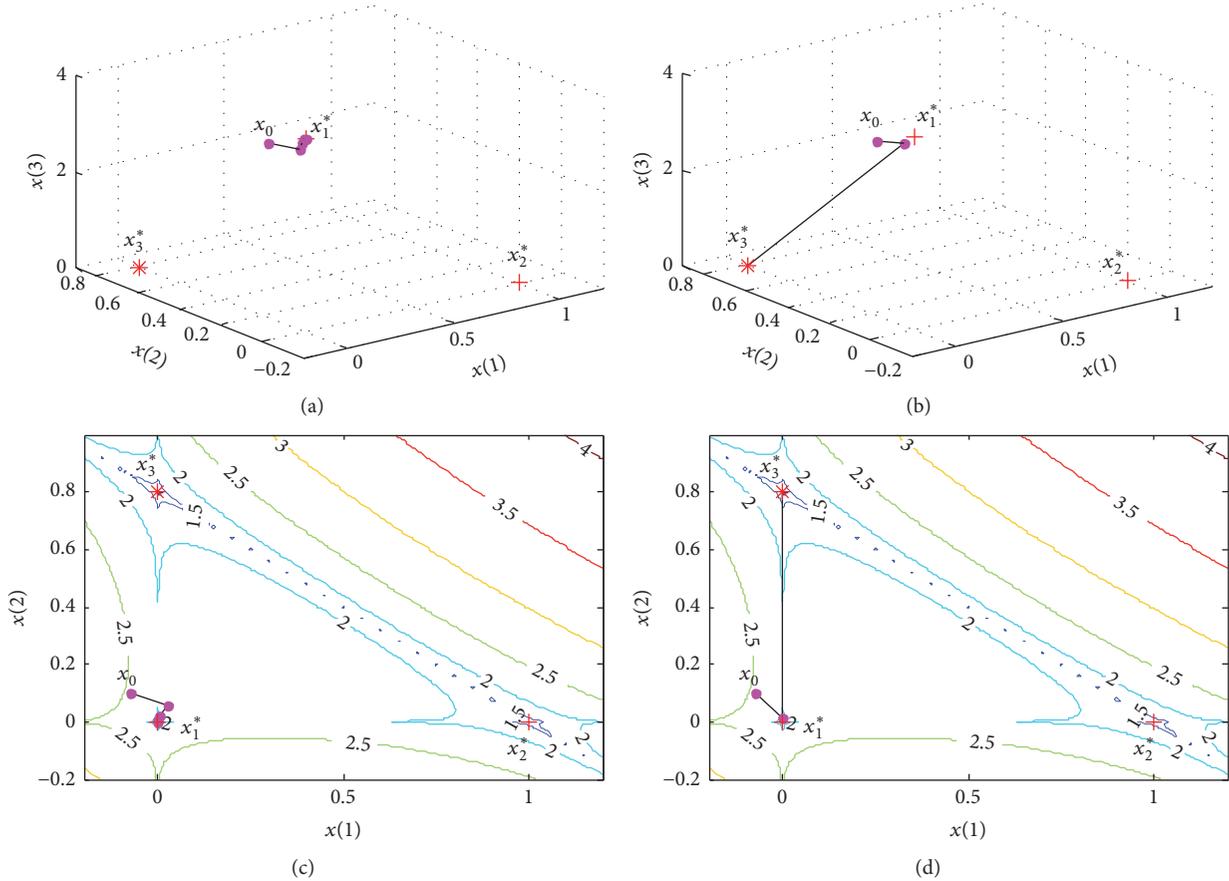


FIGURE 2: Iteration processes of AST and AGP. (a) Sequence solved by AST converges to  $x_1^*$ . (b) Sequence solved by AGP converges to  $x_3^*$ . (c) Iteration process of AST. (d) Iteration process of AGP.

If  $x^*$  has  $K$  nonzero entries, then the locations of  $n - K$  zero entries need to be determined. According to Lemma 5, AGP obtains multiple zero entries at each iteration, and these zero entries do not change in the remaining iterations based on the definition in (20). Therefore, the number of iterations of AGP is no more than  $n - K$ .

*Remark 7.* Theorem 6 shows that AGP obtains a sparse solution within a finite number of iterations, while Theorem 1 shows that AST requires an infinite number of iterations to obtain a sparse solution. In theory, the number of iterations of AGP is less than AST.

Figure 2 gives an example to show the improved convergence performance of AGP compared to AST. There are three sparse solutions  $x_1^*, x_2^*, x_3^* \in R^3$ , where  $x_3^*$  is the global optimal sparse solution. To clearly display the iteration process of AGP, we return all  $x_{D_k}$  to three dimensional space, which form a sequence  $\{x_k\}_{k=1}^L$ , where  $x_k(D_k) \leftarrow x_{D_k}$  represents that the entries of  $x_{D_k}$  located in  $D_k$  which are assigned to  $x_k$ .

Starting from  $x_0$ , a sequence  $\{x_k\}_{k=1}^6$  solved by AST converges to  $x_1^*$  in Figure 2(a). Figure 2(c) shows that all iteration solutions concentrate in the attraction basin of  $x_1^*$  in the contour map. The second iteration solution of AGP in

Figure 2(b), however, moves away from  $x_1^*$  and reaches  $x_3^*$ . In Figure 2(d), the iteration solution moves out of the attraction basin of  $x_1^*$  and enters the attraction basin of  $x_3^*$ , in which the adaptive stepsize plays an important role. This example verifies that AGP can find a sparser solution than AST by calculating the minimum point of the search direction at each iteration.

*2.4. Comparison of Convergence Performance.* To compare the convergence performance of AST and AGP, we give an experiment by the following  $l_p$ -norm problem

$$\begin{aligned}
 \min \quad & E^{(0.5)}(x) = \sum_{i=1}^8 |x(i)|^{0.5} \\
 \text{s.t.} \quad & x_1 + 9x_2 + x_3 + 9x_4 + 2x_5 + x_6 + 6x_7 + 7x_8 \\
 & = 11.5 \\
 & 11x_1 + 9x_2 + 4x_3 + 5x_4 + 3x_5 + 10x_6 + x_7 \\
 & + 4x_8 = 13 \\
 & 10x_2 + 3x_3 + 10x_4 + x_5 + 6x_6 + 10x_7 + 6x_8 \\
 & = 16.2.
 \end{aligned} \tag{29}$$

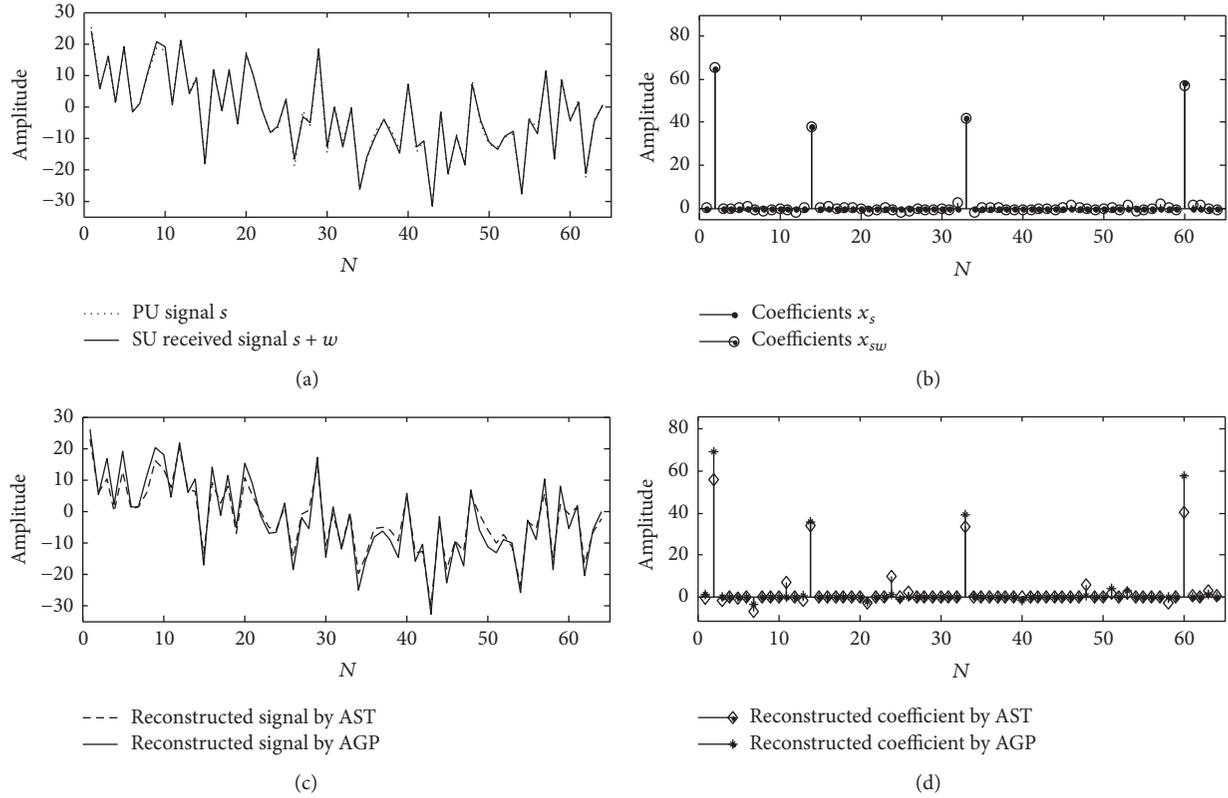


FIGURE 3: Reconstruction of the SU received signal using AST and AGP, respectively. (a) PU signal  $s$  and SU received signal  $s + w$ . (b) Coefficients  $x_s$  and  $x_{sw}$ . (c) Reconstructed SU received signals solved by AST and AGP. (d) Reconstructed coefficients solved by AST and AGP.

TABLE 1: List of the partial iteration solutions solved by AST.

$k$	$x_k^T$
1	(0.0339, 0.5510, 0.0625, 0.3716, 0.0024, 0.4874, 0.3344, 0.0860)
3	(-0.0000, 0.7245, 0.0007, 0.2784, -0.0000, 0.4747, 0.3310, 0.0017)
5	(-0.0000, 0.8575, 0.0000, 0.1115, -0.0000, 0.4334, 0.3910, 0.0000)
7	(-0.0000, 0.9379, 0.0000, 0.0094, -0.0000, 0.4084, 0.4276, 0.0000)
9	(-0.0000, 0.9453, 0.0000, 0.0000, -0.0000, 0.4061, 0.4310, 0.0000)

There exist two sparse solutions  $x_1^* = (0, 0.9453, 0, 0, 0, 0.4061, 0.4310, 0)^T$  and  $x_2^* = (0, 0, 0, 1.2, 0, 0.7, 0, 0)^T$ , where  $x_2^*$  is the global optimal sparse solution. We choose  $x_0 = A^+ y$  as an initial point. Comparing the results in Table 1 with Table 2, we see that AGP quickly finds  $x_2^*$  in smaller and smaller subspace, while AST limited by the initial point just obtains the suboptimal solution  $x_1^*$ . On the other hand, the computing times of AST and AGP are 0.0063 s and 0.0089 s, respectively. AGP costs some time to find an adaptive stepsize at each iteration, but it can obtain the global minimizer. Obviously, it is more important to find the global optimal sparse solution of problem (3).

### 3. Application of Adaptive Gradient Projection Algorithm in Compressed Spectrum Sensing

The Compressive Spectrum Sensing (CSS) is considered for this study, because it performs the same tasks as signal sparse representation. In [15, 16], the model of CSS is formulated as follows:

$$\begin{aligned} H_0 : y &= \Phi w, \quad \text{PU absent} \\ H_1 : y &= \Phi (s + w), \quad \text{PU present,} \end{aligned} \quad (30)$$

where  $y$  is a measurement,  $s$  is a Primary User (PU) signal,  $w$  is a Gaussian noise,  $s + w$  is a Secondary User (SU) received signal, and  $\Phi \in R^{m \times n}$  is a Gaussian random matrix. Assume that  $s$  and  $w$  can be represented on the discrete cosine basis  $\Psi$ , (i.e.,  $s = \Psi x_s$  and  $w = \Psi x_w$ ), where  $x_s$  and  $x_w$  are spectrum coefficients. Let  $A = \Phi \Psi$ ; the model in (30) can be reformulated as

$$\begin{aligned} H_0 : y &= A x_w, \quad \text{PU absent} \\ H_1 : y &= A (x_s + x_w) = A x_{sw}, \quad \text{PU present.} \end{aligned} \quad (31)$$

CSS intends to reconstruct  $x_{sw}^*$  from  $y = A x_{sw}$ , so the reconstruction error  $\varepsilon_r = \|\Phi x_{sw}^* - \Phi x_{sw}\|_2 / \|\Phi x_{sw}^*\|_2$  is used to evaluate the reconstruction performance.

Corresponding to  $s$  and  $s + w$  in Figure 3(a), Figure 3(b) shows that  $x_s$  has  $K = 4$  nonzero entries, while  $x_w$  is

TABLE 2: List of the partial iteration solutions solved by AGP.

$k$	$x_{D_k}^T$	$D_k$
1	(0.0030, 0.6181, 0, 0.3752, -0.0115, 0.5101, 0.2972, 0.0410)	(1, 2, 3, 4, 5, 6, 7, 8)
2	(-0.0005, 0.6860, 0.3288, 0.0017, 0.4870, 0.3128)	(1, 2, 4, 5, 6, 7)
3	(-0.0070, 1.1946, 0.0229, 0.7034, 0.0010)	(1, 4, 5, 6, 7)
4	(1.2000, 0.7000)	(4, 6)

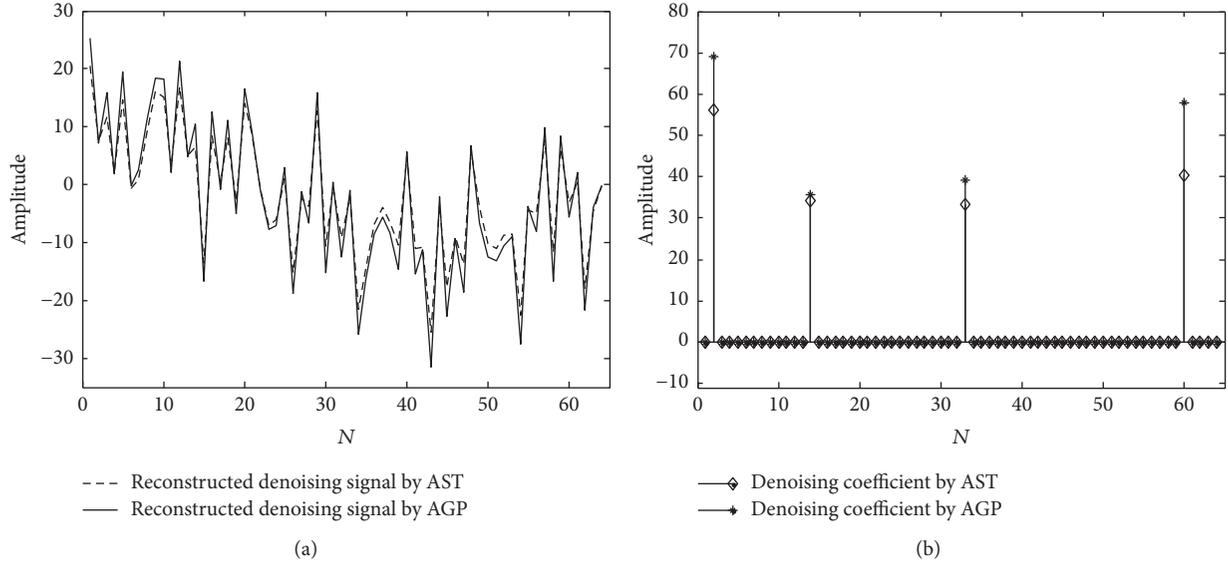


FIGURE 4: SU received signals are reconstructed with the denoising spectrum coefficients solved by AST and AGP, respectively. (a) Reconstructed SU received signals based on AST and AGP. (b) Denoising spectrum coefficients solved by AST and AGP.

distributed throughout the frequency domain. The reconstructed spectrum coefficient is obtained by solving the following problem

$$\min_x E^{(0.7)}(x_{sw}) = \sum_{i=1}^{64} |x_{sw}(i)|^{0.7} \quad (32)$$

$$\text{s.t. } y = Ax_{sw},$$

where  $A \in R^{20 \times 64}$ . After 21 iterations,  $x_{AST}^*$  solved by AST in Figure 3(d) is unsatisfactory, while  $x_{AGP}^*$  solved by AGP converges to  $x_{sw}$  after 17 iterations. The corresponding computing times of AST and AGP are 0.0154 s and 0.0227 s, respectively. The reconstructed signals  $\Phi x_{AST}^*$  and  $\Phi x_{AGP}^*$  are shown in Figure 3(c), and the reconstruction errors are 27.22% and 9.99%, respectively. At the cost of a little computing time, the reconstruction performance of AGP is improved compared to AST in noise environment, so AGP exhibits better performance of noise suppression than AST.

Note that the characteristic of noise suppressing can greatly improve the detection performance of CSS, especially when the number of nonzero entries  $K$  is not given in advance. We can reconstruct the SU received signal by choosing some larger nonzero entries of  $x_{sw}^*$  with a threshold  $\eta$ . Setting  $\eta = 10$ , Figure 4(b) shows the denoising spectrum coefficients are sparser than the coefficients in Figure 3(d). The corresponding reconstructed SU received signals are shown in Figure 4(a). The reconstruction errors reduce to

21.71% and 8.50%. Meanwhile, the variance  $\sigma_w^2$  of white noise  $w$  reduces to  $\sigma_w^2$ .

Next, we consider the detection performance using the reconstructed SU signal with denoising coefficient. Let  $P_f$  be a false alarm probability and  $\lambda$  be a judgment threshold. Then, a binary hypothesis testing problem is used to determine whether the PU is present

$$P_d = \begin{cases} 0, & H_0 : e(s_d) < \lambda \\ 1, & H_1 : e(s_d) \geq \lambda, \end{cases} \quad (33)$$

where  $e(s_d) = \|s_d\|_2^2$  denotes the energy of the detection signal  $s_d = \Psi x_d$ . The  $N_t$  Monte Carlo experiments are performed to test the detection probability  $P_d = N_d/N_t$ , where  $N_d$  is the number of accurately detecting PU signal. Given  $P_f = 0.05$ ,  $N_t = 100$ , Figure 5 shows that the Energy Detection (ED) method [17] exhibits high mistaken probabilities in low signal-to-noise-ratio (SNR) environments. However, the detection probabilities of AST, AGP, IRL1, and ITM are greatly enhanced, when the reconstructed SU received signals are solved by the denoising spectrum coefficients. For example, when SNR equals to -5 dB, the detection probabilities using AST, AGP, IRL1, and ITM improve by 75.47%, 79.25%, 58.49%, and 47.17% compared with ED. Furthermore, when SNR changes from -15 dB to -1 dB, AGP shows better detection performance than AST, IRL1, and ITM because of its improved reconstruction performance.

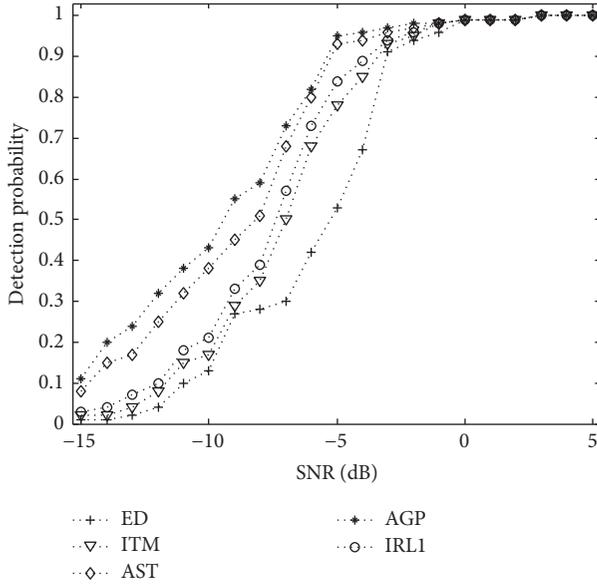


FIGURE 5: Comparison of the detection performance using ED, ITM, AST, AGP, and IRL1 for different SNRs.

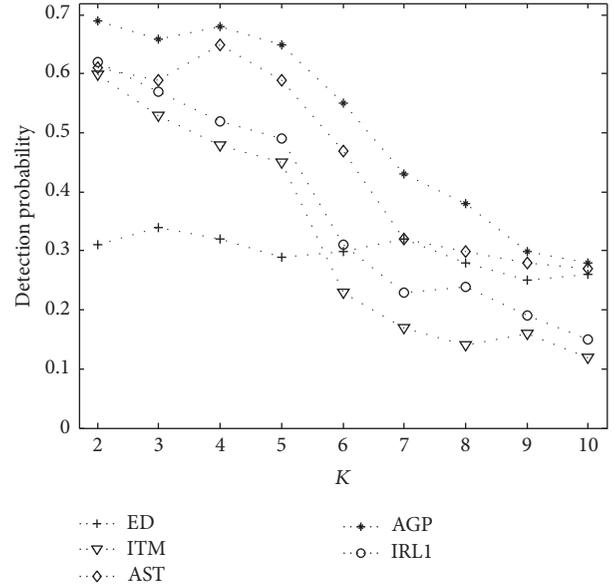


FIGURE 7: Comparison of the detection performance using ED, ITM, AST, AGP, and IRL1 for different sparsity of the spectrum coefficient vector.

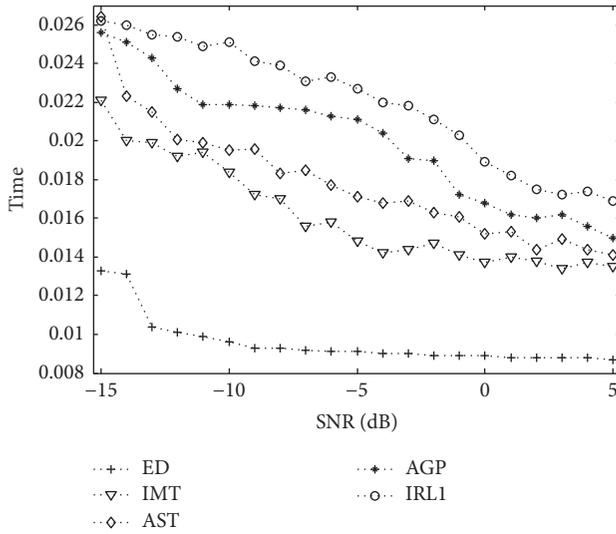


FIGURE 6: Comparison of the computing time using ED, ITM, AST, AGP, and IRL1 for different SNRs.

Figure 6 shows the corresponding computing time of five reconstruction algorithms, in which the time consumption of AGP increases at most 43.66% and 23.39% compared with ITM and AST. Spending a little more computing time, AGP can attain the best result of spectrum sensing especially in low SNR environment.

Note that the sparsity of the spectrum coefficient vector has impact on the reconstruction performance of  $l_p$ -norm minimization. Given SNR is  $-7$  dB, Figure 7 displays that the detection probabilities of AST, AGP, IRL1, and ITM descend when  $K$  changes from 2 to 10. This is because the measurement vector  $y \in R^{20}$  in (32) is not able to attain the whole information of a SU received signal when  $K$  is larger than 6. Therefore, the unsatisfactory reconstruction

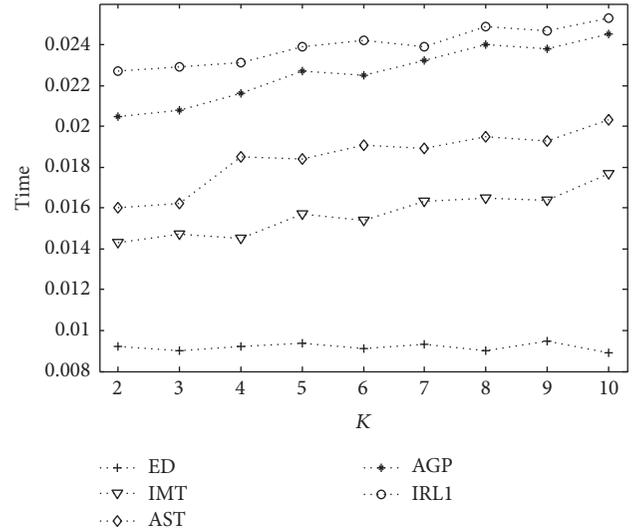


FIGURE 8: Comparison of the computing time using ED, ITM, AST, AGP, and IRL1 for different sparsity of the spectrum coefficient vector.

results of  $l_p$ -norm minimization greatly affect the detection performance of CSS via AST, IRL1, and ITM, while the performance degradation of CSS via AGP is slower than that of the three reconstruction algorithms. Meanwhile, AST, AGP, IRL1, and ITM cost more computing time to find the sparse solution in Figure 8. The above results reveal that the detection performance of CSS needs to be improved when the number of measurement is insufficient. ED determines the state of PU by measuring the energy of a SU received signal, so the property of the sparsity has little impact on its detection result and computing time.

## 4. Conclusions

Signal sparse representation has become a fundamental tool that is embedded into various application systems. One of its fundamental problems is finding a sparse coefficient. In this paper, we develop a novel AGP algorithm to solve the  $l_p$ -norm minimization. Theoretical analysis demonstrates that AGP can find a sparser solution than AST, because it avoids the iteration solutions concentrating in the attraction basin of a suboptimal sparse solution. Applying AGP to compressed spectrum sensing, it can obtain the better detection performance than ED, AST, IRL1, and ITM by spending a little more computing time. Future research will extend AGP to more scenarios.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This research was supported by the National Natural Science Foundation of China (nos. 61501224, 61501223, and 61502230), the Natural Science Foundation of Jiangsu Province (no. BK20150960), and the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (no. 17KJB510024).

## References

- [1] Y. Li, Z. L. Yu, N. Bi, Y. Xu, Z. Gu, and S.-I. Amari, "Sparse representation for brain signal processing: a tutorial on methods and applications," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 96–106, 2014.
- [2] Y. Wong, M. T. Harandi, and C. Sanderson, "On robust face recognition via sparse coding: The good, the bad and the ugly," *IET Biometrics*, vol. 3, no. 4, pp. 176–189, 2014.
- [3] J. Yin, J. Sun, and X. Jia, "Sparse Analysis Based on Generalized Gaussian Model for Spectrum Recovery with Compressed Sensing Theory," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2752–2759, 2015.
- [4] T.-S. T. Chan and Y.-H. Yang, "Informed group-sparse representation for singing voice separation," *IEEE Signal Processing Letters*, vol. 24, no. 2, pp. 156–160, 2017.
- [5] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, 1995.
- [6] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.
- [7] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.
- [8] C.-B. Song and S.-T. Xia, "Sparse signal recovery by  $\ell_q$  minimization under restricted isometry property," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1154–1158, 2014.
- [9] F. Y. Wu and F. Tong, "Gradient optimization p-norm-like constraint LMS algorithm for sparse system estimation," *Signal Processing*, vol. 93, no. 4, pp. 967–971, 2013.
- [10] H. S. Mousavi, V. Monga, and T. D. Tran, "Iterative convex refinement for sparse recovery," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1903–1907, 2015.
- [11] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 600–616, 1997.
- [12] B. D. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," *IEEE Transactions on Signal Processing*, vol. 47, no. 1, pp. 187–200, 1999.
- [13] E. J. Candès, M. B. Wakin, and S. Boyd, "Enhancing Sparsity by Reweighted  $\ell_1$  Minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5–6, pp. 877–905, 2008.
- [14] Y. She, "Thresholding-based iterative selection procedures for model selection and shrinkage," *Electronic Journal of Statistics*, vol. 3, pp. 384–415, 2009.
- [15] D. Romero and G. Leus, "Wideband spectrum sensing from compressed measurements using spectral prior information," *IEEE Transactions on Signal Processing*, vol. 61, no. 24, pp. 6232–6246, 2013.
- [16] S. Ren, Z. Zeng, C. Guo, and X. Sun, "A Low Complexity Sensing Algorithm for Wideband Sparse Spectra," *IEEE Communications Letters*, vol. 21, no. 1, pp. 92–95, 2017.
- [17] S. Atapattu, C. Tellambura, H. Jiang, and N. Rajatheva, "Unified Analysis of Low-SNR Energy Detection and Threshold Selection," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 11, pp. 5006–5019, 2015.

## Research Article

# Allocation of Distributed Energy Systems at District-Scale over Wide Areas for Sustainable Urban Planning with a MILP Model

Yeşim Ok <sup>1</sup> and Mehmet Atak<sup>2</sup>

<sup>1</sup>Industrial Engineering Department, Ataturk University, 25040 Erzurum, Turkey

<sup>2</sup>Industrial Engineering Department, Gazi University, 06070 Ankara, Turkey

Correspondence should be addressed to Yeşim Ok; [yesim.ok@atauni.edu.tr](mailto:yesim.ok@atauni.edu.tr)

Received 14 December 2017; Revised 5 March 2018; Accepted 1 April 2018; Published 13 May 2018

Academic Editor: Paula L. Zabala

Copyright © 2018 Yeşim Ok and Mehmet Atak. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An optimal allocation of centralized district-scale distributed energy resource (DER) systems with district heating and cooling network (DHCN) is studied. A generic mixed integer linear programming (MILP) model is constructed to increase the system efficiency and decrease costs by reducing energy distribution losses and transportation costs in energy distribution network. Initial investment costs based on size and type (co/trigeneration) of the facility and demand-weighted transportation costs are minimized by the capacitated fixed charge facility location (FCFL) model. However, unlike the standard FCFL model, by adding the maximum coverage distance in the set covering problem, a new method has been proposed. Thus, its aim is to avoid assigning a demand point to a candidate facility from a point farther than the predetermined distance. That means that the weakness of disregarding the distance between supply and demand points in the FCFL problem is eliminated. Additionally, this model, in which the annual inputs are used, has a generic framework suitable to form infrastructure needs in consideration of distributed energy systems in the general planning level for sustainable urban planning. For this purpose, the applications of the model, both case study and tests, have been made over wide areas, with annual demand and capacity values. Consequently, a case study with different coverage distances has been conducted to see the effect of coverage distance on the model, and also test problems with different sizes have been carried out to demonstrate the capability of the proposed model.

## 1. Introduction

The prominence of effective energy generation is clearer when the intense environmental pollution created by the conventional energy generation process is taken into account. The meaning of “efficiency” for energy is both to do the same job with less energy and to prevent transmission and distribution losses.

Distributed energy systems generate electricity, even heat and cold energy, at many small-scale local units [1]. With an electricity generation unit located at or near end users, it is aimed at reducing transmission losses significantly, and also simultaneous generation systems, cogeneration or trigeneration systems, cause high thermodynamic efficiency and primary energy saving as well as reducing gas emissions

because of utilization of exhaust heat. Additionally, these systems offer the opportunity to utilize local renewable energy sources (e.g., solar, wind, and biomass) and add flexibility to the generation system [2].

Although cogeneration is not a new concept, the residential and commercial usage is becoming widespread, nowadays, especially in developed European cities. International Energy Agency has foreseen that the annual distributed electricity generation will increase by 4.2% from 2000 to 2030 and reach 35 GW by 2030 [3].

Systematic analysis and evaluation are essential for optimal design and operation of DERs. In that sense, mathematical programming techniques have been widely used for this purpose. Because of its flexibility and robustness, mixed integer linear programming (MILP) model has been more

preferred. Both energetic and environmental advantages can increase if DER system limits are extended from building level to district-scale with distribution networks.

We can divide the optimization models for optimal design and operation of DER systems into two main groups: while the majority of the initial studies ([2, 4–9]) deal with the determination of optimal configuration, in other words, optimal system combination and operational strategies for a single DER system, more recent works ([10–24]) also take into account the pipeline network, in more detail. Nevertheless, there is no doubt that the complexity is further increased when the DER systems are designed for covering the demands of multiple end users at the neighborhood scale, which often involves also the design of a heating pipeline network, concurrently.

Lozano et al. [10] examined the systemic impact of legal restrictions of CCHP systems including thermal storage on a 5000-apartment complex in Spain. Weber and Shah [11] proposed a single-level optimization model called DESDOP (district energy system design and optimization) for an ecological city that is used to determine the optimal technology combination reducing CO<sub>2</sub> emissions, considering 5% heat loss.

Ortiga et al. [12] conducted a scenario analysis about system efficiency, investment cost, investment return period, and CO<sub>2</sub> emissions of new a polygeneration system including DHCN in Spain for the Polycity project. Mehleri et al. [13] proposed a superstructure MILP model for a DER system design and operation with heating pipeline network at the neighborhood level consisting of 10 to 20 buildings considering sale of excess electricity to the grid as income. Besides determining the optimal combination and allocation of DER technologies and their operation strategies, the economic and environmental impacts are also discussed under different scenarios including thermal storage tanks in their other study [14] at a neighborhood of 5 buildings in Greece. Similarly, a MILP model has been developed by Ameri and Besharati [15] in order to design and operate an optimal CCHP/photovoltaic (PV)/DHC energy system in a residential town in Tehran. Since a much larger area of 50 hectares with 137 buildings was taken into account, 15% thermal loss and 5% cold loss per km for the pipeline have been also included into the model.

Omu et al. [16] introduced a model at the objective of impact assessment of strategic policies about energy systems, to design a centralized district-scale DER system for an area of 300 hectares with six buildings and distribution network, based on local climate data, consumers' energy demands, electricity, gas tariffs, and so on. Bracco et al. [17] proposed a MILP model determining the design and the operation of a CHP distributed generation system in an urban area in order to minimize both annual costs and CO<sub>2</sub> emissions. Later they extended their model to a general mathematical model, to optimally design a distributed energy system composed of different technologies that provide heating, cooling, and electricity to a set of buildings [18].

Yang et al. structured a model determining the optimal sites, types, capacities, numbers for the energy distribution

stations, and optimal routes of the energy distribution networks with district-scale [19]. A multiobjective optimization model of total cost versus carbon emissions was presented to determine the benefits of decentralized heating networks in the context of DERs where carbon emissions are expressed with a carbon tax by Morvaj et al. [21].

Sameti and Haghghat [22] developed a two-level optimization methodology to help design a trigeneration system for a given district system which satisfies the heating, cooling, and hot water demands and, at the same time, minimizes the annual total costs and CO<sub>2</sub> emissions. Marquant et al. [23] proposed a framework to enable large-scale optimization of urban energy systems at neighborhood scale by optimally clustering buildings into small-scale district heating networks and then optimally linking these clusters with further network connections.

Zheng et al. formulated a MINLP model to determine the optimal combination and operational strategies of various technologies to meet the energy requirements under different circumstances. Unlike previous studies, which ignored the real time variations of energy demand, they considered the effect of hourly demand fluctuations [24].

In this study, as a contribution to the current literature, maximum coverage distance of set covering model has been integrated into capacitated fixed charge facility location problem in order not to allow assignment of a demand point to a candidate facility from a point farther than the predetermined distance. By this way, it is aimed at minimizing the energy transmission losses and the transportation costs with eliminating the weakness of disregarding the distance between the customer and the facility in the FCFL problem. Besides, this model, based on annual demand values over wide areas, could be considered as a useful tool for general planning level of spatial urban planning process in which distributed energy systems are involved.

The rest of this paper is organized as follows: Firstly, the problem definition is made in Section 2 and further methodology with the proposed mathematical model is described in Section 3. The case study with different coverage distances is depicted in Section 4, and then the results are presented in Section 5. Additionally, test phase with 100 test problems with different parameter ranges is dealt with in Section 6. Finally, conclusions are drawn in Section 7.

## 2. Problem Definition

This facility location and assignment problem for DER systems is based on fixed charge model from median based models. Median based models are better suited to make distribution plans, as transport costs are taken into account when real distances are used, whereas with covering models, a maximum distance exists a priori [25].

Operational optimization models evaluate how to integrate and assign generation technologies at district or building scales. According to classification of optimization models in district energy systems by Sameti and Haghghat [26], our proposed model is included in superstructure models.

District-scale DER system is an energy network system in which energy is produced outside the buildings and

distributed to the demand points through energy distribution networks to be consumed. In this study, it is assumed that the DERs are located outside the energy-consuming buildings and that the generated energy is delivered to the demand points via the energy distribution networks. Unlike the previous studies, users' electricity, thermal, and cold energy demands were considered in total annual values. As mentioned before, this is preferred to facilitate the general planning level of spatial urban planning process in which distributed energy systems are involved.

In this proposed model, when the users are assigned to the facilities, the most important issue to be regarded is whether the distance between the facility and the user is exceeding the maximum coverage distance or not. If it exceeds it, the user is assigned to another facility if there is one that is closer. As the installation costs of DHCN constitute almost half of the whole system, the fewer the pipes are, the lower the costs and losses will be. That is why the maximum coverage distance is added to the proposed model.

With this model, while the assignment of district-scale DER systems has been made with the least initial and transportation costs, the facility type (cogeneration/trigeneration), facility size (production capacity), and structure of district heating and cooling network are also decided, simultaneously.

Moreover, both thermal losses and electricity transmission losses are taken into consideration. When these losses are included in the model, they are treated as a percentage increase for each hectometer on demand for the amount of energy involved, not as a percentage decrease in the capacity.

In the energy planning, urban planners emphasize the need for a higher level of integration between both land and energy planning. The consistent land and energy planning approach is one of the basic principles of sustainable urban planning. The spatial planning could be divided into three levels: general planning, special planning, and detailed planning. The main aim of the general planning is to decide the land use and determine the infrastructure needs. Energy planning at this level is preliminary energy infrastructure planning in order to reserve land for the infrastructure needs. The maximum possible energy demands and designing of corresponding energy infrastructure are evaluated so that they could be supplied [27].

Spatial planning is crucial to implement successful district heating and cooling systems because of the need for optimal infrastructure planning. One of the advantages of this model is its capability to be used for general level of spatial urban planning. As distributed generation is spreading, while creating urban plans, the land use and infrastructure requirements for distributed energy systems should be considered at the creation level of urban plans.

This proposed model provides a generic calculating framework for general plans in urban scale planning by using annual demand and capacity values. The use of annual data should not to be seen as simplification; with this approach, it is aimed at establishing the most appropriate infrastructure network for distributed energy systems. Considering the previous studies with seasonal demand changes, it is seen that implementation was made in a narrow area [13, 16, 20].

However, this study would be appropriate to form a general framework as a view, especially for energy infrastructure, with the inclusion of much wider areas for utilization in spatial urban planning. It can be directly applied to different scale areas in any location with different building types and energy demands for general level of sustainable urban planning.

### 3. Mathematical Formulation

In the set covering model developed by Toregas et al. in 1971 [28] where  $I$  is the set of demand points and  $J$  is the set of candidate facilities, for  $\forall i \in I$  and  $\forall j \in J$ ,  $d_{ij}$  is the distance between  $i$  and  $j$  points.  $D_c$  is the coverage distance while  $N_i = \{j \in J, d_{ij} \leq D_c\}$ ;  $N_i$  is the set of all  $j$  facilities covering demand point  $i$ , where  $X_j$  equals 1 if the candidate facility is opened at point  $j$  and 0 if not, with 0-1 being binary variables.

The objective of the FCFL model, developed by Balinski in 1965 [29], is to minimize investment and transportation costs while determining the optimal number and location of the facilities to be opened and the assigning of demand points to the opened facilities. The point to be noted here is that, due to the capacity constraint in the fixed charge model, the demand points may not be assigned to the closest one. In order to eliminate the weakness of ignoring the distance between facility and user, maximum coverage distance has been included in the proposed model. This ensures that the distance between the facility and the demand point remaining is within the predetermined limits while the distribution network structure is constructed.

**3.1. Mathematical Model.** The fixed charge facility location model is an effective tool for analysis of how many distribution centers should be built and where they should be located. We add the maximum coverage distance from the set covering model into our capacitated FCFL model with the assignment restriction. Only one facility is assigned for demand node  $i$  that is covered by candidate site  $j$  if  $d_{ij} \leq D_c$ , where  $D_c$  is the coverage distance.

The name of the proposed model is maximum coverage distance-added fixed charge facility location (MCD + FCFL) model.

#### 3.1.1. Nomenclature

##### Indices

$i$ : Index for demand node ( $i \in I$ )

$j$ : Index for candidate facility site ( $j \in J$ )

$k$ : Index for energy type ( $k =$  electricity, thermal, and cooling)

$l$ : Index for facility size ( $l =$  small, medium, large)

$m$ : Index for facility type (cogeneration or trigeneration)

$N_i = \{j \mid d_{ij} \leq D_c\}$ , the set of all candidate sites that can cover demand node  $i$ .

### Parameters

$f_{lm}$ : Fixed initial investment cost of locating an  $m$  type facility with  $l$  size at site  $j$  (million €)

$h_{ik}$ : Demand quantity of energy type  $k$  at demand point  $i$  (GWh/year)

$d_{ij}$ : Distance between customer  $i$  and facility  $j$  (hectometers)

Dc: Maximum coverage distance (hectometers)

$S_{lk}$ :  $k$  production capacity of facility  $j$  with size  $l$  (GWh/year)

$c_1$ : Unit transport cost for electricity (million €/hectometer)

$c_2$ : Unit transportation cost for district heating and cooling network (million €/hectometer).

**Decision Variables.** There are two binary decision variables for this model:

$x_{jlm} = 1$  if a facility of type  $m$  with size  $l$  is opened at candidate facility site  $j$ ; otherwise, it equals 0;

$y_{ijk} = 1$  if a demand node  $i$  is serviced with energy type  $k$  from a facility at site  $j$ ; otherwise, it equals 0.

### 3.1.2. Objective Function

$$\min Z = C_{\text{inv}} + C_{\text{elec}} + C_{\text{net}}, \quad (1)$$

$$\begin{aligned} \min Z &= \sum_j \sum_l \sum_m f_{lm} * x_{jlm} + c_1 * \sum_i \sum_j h_{i1} * d_{ij} * y_{ij1} \\ &+ c_2 * \sum_i \sum_j \sum_{k=2}^3 h_{ik} * d_{ij} * y_{ijk}. \end{aligned} \quad (2)$$

The objective of the model is to minimize the annual cost of supplying energy to consumers through the distribution of the energy generated by the DER system. This annual economic cost, given in (2), is the sum of the initial investment cost, electricity transportation cost, and district heating and cooling network cost ( $C_{\text{inv}}$ ,  $C_{\text{elec}}$ , and  $C_{\text{net}}$ , resp.). Another noteworthy point here is that the transportation costs are demand-weighted.

### 3.1.3. Constraints

$$\sum_l \sum_m x_{jlm} \leq 1, \quad \forall j \in J. \quad (3)$$

Constraint (3) states that only one facility can be opened at one candidate point.

$$\sum_{j \in N_i} y_{ijk} \geq \frac{h_{ik}}{M}, \quad \forall i \in I, \forall k \in K. \quad (4)$$

Constraint (4) requires that each demand node can be assigned to only one facility for each product. In addition,

with constraint (4), coverage distance (Dc) in set clustering model is incorporated into our FCFL model by the set  $N_i$ , which is the set of all candidate sites that can cover demand node  $i$  [ $N_i = (j \in J : d_{ij} \leq \text{Dc})$ ].

As you may have noticed, demand is divided into  $M$ , a big number to avoid the assignment of relevant energy type  $k$ , when  $h_{ik} = 0$ . Although the case of no demand for any product is not considered in our case study, because the model we have proposed is generic one, we added  $M$  against possible situations.

$$y_{ij3} \leq \sum_l x_{jl2}, \quad \forall i \in I, \forall j \in J, \quad (5)$$

$$y_{ijk} \leq \sum_l \sum_m x_{jlm}, \quad \forall i \in I, \forall j \in J, k = 1, 2. \quad (6)$$

Constraints (5) and (6) restrict demand node assignments only to open facilities. There are two separate constraints for similar purpose because we have two different facility types: cogeneration and trigeneration. For the situation index  $k = 3$  (demand energy type is cooling) constraint (5) ensured that the relative demand node  $i$  assigns a trigeneration facility; otherwise, for  $k = 1$  or 2 the same demand node  $i$  can be assigned to either trigeneration or cogeneration facility  $j$  with constraint (6).

$$\sum_i (1 + 0.005 * d_{ij}) * h_{ik} * y_{jk} \leq \sum_l \sum_m S_{lk} * x_{jlm}, \quad \forall j \in J, k = 1, \quad (7)$$

$$\sum_i (1 + 0.03 * d_{ij}) * h_{ik} * y_{jk} \leq \sum_l \sum_m S_{lk} * x_{jlm}, \quad \forall j \in J, k = 2, 3. \quad (8)$$

Constraints (7) and (8) are capacity constraints. While limiting the demand  $i$  for the product  $k$  to the product capacity of the facility  $j$ , transmission losses for both thermal/cold and electricity are treated as two separate constraints.

These transmission losses are 3% for thermal and cold energy per hectometer (hm) and 0.5% for electric energy per hm. The calculation is considered by the distance between demand point  $i$  and facility point  $j$ ; also, the loss of the model is taken into account as the increase in demand, not the decrease in capacity.

$$x, y \in \{0, 1\}. \quad (9)$$

Constraint (9) established the decision variables as binary.

## 4. Experimental Case

In order to demonstrate the applicability of the proposed model, a simple case study is dealt with step-by-step. This model, encoded in the GAMS interface, was solved using CPLEX 8.1 solvent.

Let us assume 20 demand nodes, for especially residential and commercial use such as hospitals, hotels, shopping centers, schools, university campuses, stadiums, public buildings, and satellite towns and 8 candidate energy generation

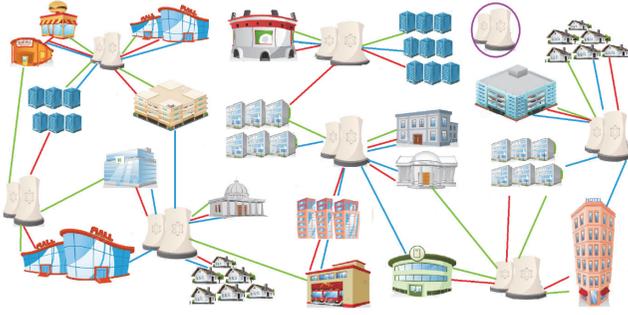


FIGURE 1: Illustration of energy generation network for DERs at district-scale.

sources illustrated as in Figure 1. An area of approximately 150 hectares (1,500,000 m<sup>2</sup>) was regarded for our sample application.

#### 4.1. Constitution of Parameters

**4.1.1. Annual Demand Values of Users ( $h_{ik}$ ).** Electricity demand is randomly generated between 5 and 35 million kWh, annually. When calculating a customer's thermal energy demand, a value of 30% to 150% of the predetermined annual electricity demand of the same customer is taken randomly. In a similar way, a random value between 5% and 30% of the predetermined annual electricity demand is taken for cooling energy. As mentioned before the main reason for using annual total demand values without considering seasonal fluctuations is that this proposed model is designed as a generic model that can be used in general level of spatial planning for sustainable city planning. Table 1 gives the annual demand values for each energy type.

**4.1.2. Distances between Users and Candidate Facility Points ( $d_{ij}$ ).** Euclidean distances were calculated using random coordinates generated at specific intervals for the demand points and the candidate facility points. Table 2 presents the Euclidean distances,  $d_{ij}$ , between demand point  $i$  and candidate facility  $j$ .

**4.1.3. Annual Energy Generation Capacities of Candidate Facilities ( $S_{lk}$ ).** Candidate facility points are classified as small, medium, and large considering their production capacities between 1–5 MW, 5–10 MW, and 10–20 MW, respectively. Electric energy production capacities are produced randomly for each size between the limits. Thermal and cold energy capacities are obtained by taking a certain proportion of electricity energy generation capacity as mentioned before. And then, production capacities are converted into annual values with the assumption that the facilities operate 18 hours a day, 365 days a year. Table 3 gives the facility capacities for three different type energy generations. As an assumption it is assumed that the type of equipment to be used in all candidate facilities is the same and that cost variation occurs only according to their sizes.

TABLE 1: Annual demand values for 20 energy demand points.

Demand points	Demand value for different energy types (GWh)		
	Electricity energy	Thermal energy	Cooling energy
1	24.3	14.3	1.2
2	24.4	29.3	6.2
3	14.5	14.8	1.8
4	21.4	25.8	5.3
5	29.9	32.0	7.4
6	10.0	7.1	0.8
7	20.4	17.5	3.2
8	13.2	5.5	2.1
9	15.4	21.8	3.3
10	8.3	9.5	0.9
11	13.7	6.8	1.1
12	19.8	16.5	1.6
13	33.7	39.3	8.7
14	14.1	11.8	2.3
15	8.6	2.9	1.6
16	15.4	9.0	1.3
17	12.1	9.6	1.4
18	24.4	25.0	2.7
19	19.2	26.6	3.9
20	20.1	22.4	2.2

**4.1.4. Initial Investment Costs of Facilities ( $f_{lm}$ ).** As a result of research for the energy market, the initial investment cost for 1kW power is predetermined as 500 €. Thus, for cogeneration facilities, the initial investment costs were obtained by multiplying electricity generation capacities with 500. For trigeneration ones, additional costs between € 16,000 and € 450,000 according to capacity size are added for chillers included in the cogeneration system. Table 4 provides the fixed costs of energy generation facilities for our sample application.

**4.1.5. Maximum Coverage Distance ( $D_c$ ).** For this example, maximum coverage distance was experienced for 2, 3, 6, 7, 8, 10, and 12 hectometers.

**4.1.6. Unit Transportation Costs ( $c_1$  and  $c_2$ ).** Assuming the insulated pipeline is used for the distribution of thermal and cold energy, the unit transportation costs of these two energy types ( $c_1$ ) are evaluated differently from the unit transportation cost of electricity energy ( $c_2$ ). For our sample problem, while  $c_1$  is taken as 0.000001 million €,  $c_2$  is determined as 0.0006 million € per hm.

**4.1.7. Thermal and Electricity Losses.** Thermal and electrical losses occurring in the distribution network in the problem have also been considered as in the literature [14, 15]. As

TABLE 2: Euclidean distances between  $i$  and  $j$  (hm).

Euclidean distances	Candidate facility points								
	1	2	3	4	5	6	7	8	
Demand points									
1	4.12	4.47	4.00	1.41	5.38	6.40	4.12	5.09	
2	4.47	4.12	7.28	5.00	8.94	8.25	7.07	8.06	
3	2.24	7.62	7.07	6.33	9.22	6.08	6.40	7.21	
4	3.16	5.39	6.71	5.00	8.60	7.07	6.33	7.28	
5	5.83	13.45	7.28	9.43	9.05	3.16	6.33	6.08	
6	5.39	4.00	8.25	5.83	9.85	9.22	8.06	9.05	
7	6.00	12.53	5.00	7.81	6.33	2.00	4.24	3.61	
8	8.94	2.24	8.06	5.00	8.25	11.31	8.60	9.43	
9	5.83	13.45	7.28	9.43	9.06	3.16	6.33	6.08	
10	11.31	3.61	11.71	8.54	12.17	14.42	12.08	13.00	
11	6.71	9.06	11.40	10.00	13.45	10.44	10.82	11.66	
12	2.24	8.60	3.16	4.47	5.39	2.24	2.24	2.83	
13	2.83	10.63	6.08	7.28	8.25	2.83	5.10	5.39	
14	6.33	11.71	3.61	6.71	4.47	2.83	3.16	2.24	
15	3.61	11.40	7.07	8.25	9.22	3.61	6.08	6.33	
16	7.07	1.00	8.06	5.00	9.06	10.30	8.25	9.22	
17	6.40	10.20	2.00	5.10	2.24	4.12	2.24	1.41	
18	7.21	1.00	7.28	4.12	8.00	10.00	7.62	8.54	
19	5.39	12.81	6.33	8.60	8.06	2.24	5.39	5.10	
20	10.30	3.00	9.43	6.40	9.49	12.73	10.00	10.82	

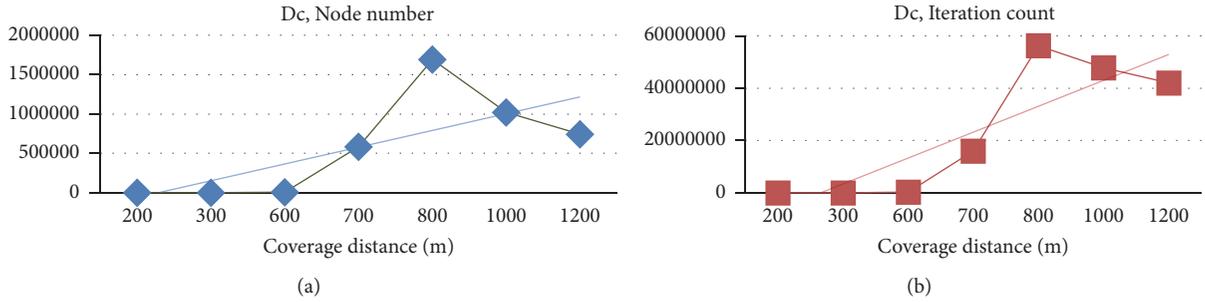


FIGURE 2: The effect of different coverage distances on model complexity with (a) node number (b) iteration count.

TABLE 3: Facility capacities according to their sizes (GWh).

Facility size	Facility generation capacity		
	Electricity	Thermal	Cool
Small	25.8	22	12.9
Medium	44.8	33.6	17.9
Large	88.2	70.5	22

TABLE 4: Initial investment costs (million €).

Facility size	Facility type	
	Cogeneration	Trigeneration
Small	2	2
Medium	3.4	3.6
Large	6.7	6.7

seen in the model discussed above, the corresponding losses depending on the distance between  $i$  and  $j$  are taken as 3% for thermal and cold energy and as 0.5% for electricity energy with the capacity constraints (6) and (7).

### 5. Results and Discussions

Maximum coverage distance has been tried for different values as 2, 3, 6, 7, 8, 10, and 12 hectometers and the effect has been analyzed by the comparison of the results.

**5.1. Results Based on Different Coverage Distances.** It is seen from Table 5 that the coverage distance is an important parameter affecting the solution time. As the  $D_c$  value decreases, the solution time also decreases; nonetheless, the number of satisfied users also decreases and, thus, unmet demand situation becomes an issue.

The  $D_c$  value of this problem is taken in the range of 2–12. The iteration counts and node numbers are two of the complexity indicators in the model solution algorithm. They are affected to a certain extent by the increase of the coverage distance as seen from Figure 2. For  $D_c = 8$  hm, these values have peaked and the solution time has reached the upper limit, that is, 7200 seconds. The model met all requests with a deviation of 1.14% from the best possible solution (from the top) at this point. It is also seen that the increase in the

coverage distance has directly influenced and increased the solution time of the model.

The results of these two assignments are also shown schematically with the real coordinate values for our case study in Figure 3.

The number of uncovered demand points is noteworthy while coverage distance is less. (15 for  $D_c = 200$  m, 5 for  $D_c = 300$  m). Although the total cost of the system is low, the multiplicity of unmet demands is not a desirable situation. After all the demand points have been evaluated, the effect of coverage distance on the total cost is only reflected with transportation costs.

When we examine the assignment results for 3 and 10 as coverage distances, in Table 5, we see that the same number of facilities (7) is opened in both cases. However, in Table 6, when coverage distance 3 is taken, 5 of the 20 demand points cannot be serviced, and 11 demand points met all energy demands from the same facility. Thus, there are 5 trigeneration and 2 cogeneration facilities with 3 of large, 2 of medium, and 2 of small scale; moreover, the proposed model calculated the total cost including the distribution network as 32.1 million €, optimally.

The following conclusions can be drawn from Table 7:

When  $D_c$  is received as 10, all of the 20 demand points are serviced, and there are 4 trigeneration and 3 cogeneration facilities with 5 of large and 2 of small scale. In this case, the model calculated a total cost value of 38.81 million € within a time limit of 7200 seconds, where the best possible integer solution is 37.98. The deviation from the best possible solution is found to be 2.2%. Among other solutions in which all demand points are serviced, this is the best result. Candidate facility with number 5 was not opened and 5 of 20 demand points met all three types of energy demands from the same facility.

### 6. Test Phase of the Model

**6.1. Test Parameters.** At this phase of the study, 100 test problems were created at different parameter ranges in order to determine the limits of the mathematical model. 10 test problems were solved in 10 different sizes with different parametric structures and 100 results were obtained in total. The parameter ranges shown in Table 8 are used.

As in the case study, possible thermal and electrical losses occurring in the distribution network are assigned as a loss

TABLE 5: Comparison of results for different coverage distances.

Dc (hm)	Object. function value	Time (Sn)	Possible best solution	% deviation	Iteration count	Node number	Not covered demands	Opened facility type Co/Tri	Opened facility size
2	11.4750	0.05	11.4750	--	6	0	15	1/4	5 medium 2, 3, 4, 6 small
3	32.1046	0.2	32.1046	--	92	0	5	2/5	1, 2, 6 large 3, 8 medium 4, 7 small
6	38.8437	11.2	38.8437	--	229669	6780	--	3/4	1, 2, 6, 7, 8 large 3, 5 small
7	38.8180	1795	38.8180	--	15939506	584878	--	3/4	1, 2, 6, 7, 8 large 3, 5 small
8	38.8268	7200	38.3907	1.14	56234954	1691231	--	3/4	1, 2, 6, 7, 8 large 3, 5 small
10	38.8187	7200	37.9830	2.2	47832871	1019017	--	3/4	1, 2, 6, 7, 8 large 3, 5 small
12	38.8305	7200	37.8577	2.6	42006374	745401	--	3/4	1, 2, 6, 7, 8 large 3, 4 small

TABLE 6: Assignment results of demand points to energy generation facilities according to requested energy types for  $D_c = 3$  hm (300 m).

Demand points	Candidate facilities								
	1	2	3	4	5	6	7	8	
1				Elec, therm, cool	No assignment				
2				No assignment					
3	Therm		Elec		No assignment		Cool		
4			Elec, therm		No assignment		Cool		
5				No assignment					
6					No assignment		Elec, therm, cool		
7					No assignment	Elec, therm, cool			
8		Elec, therm, cool			No assignment				
9				No assignment					
10				No assignment					
11					No assignment		Elec, therm, cool		
12					No assignment	Elec, cool		Therm	
13	Elec, therm				No assignment	Cool			
14					No assignment	Elec, therm, cool			
15				No assignment					
16		Elec, therm, cool			No assignment				
17					No assignment			Elec, therm, cool	
18		Elec, therm, cool			No assignment				
19					No assignment	Elec, therm, cool			
20		Elec, therm, cool			No assignment				

ratio. This loss ratio is 3% for thermal and cold energy and 0.5% for electricity per one hundred meters according to the distance between customer and generation unit.

6.2. *Test Results.* All runs are limited to 2 hours ( $2 * 60 * 60 = 7200$  seconds). The solution of the model obtained after 7200 seconds of operation is expressed as the lower bound whereas the solver calculates the best integer possible solution for nonoptimal ones defined as the upper bound. Percent deviation is also the percentage of the difference between the lower and upper bound.

From Table 9, it is clear that, as the problem size is increasing, the average solution time is increasing synchronously whereas the number of problems obtaining the optimum solution is decreasing. Another parameter that affects the solution time is the maximum coverage distance added to the

model. This value is chosen randomly between 2 and 12. As the  $D_c$  value is reduced, the number of customers satisfied is reduced and also the solution time is decreasing. However, decrease in number of satisfied customers is not a desirable situation. It is seen that the difference between the lower and upper bounds and the percentage deviation value increase together with the model size.

When the results of the test problems are evaluated, it is seen that not only the problem dimension but also the coverage distance affects both the solution time and the total costs as in Figure 4.

## 7. Conclusions

The authors built a generic MILP model for optimal allocation of centralized district-scale DERs and configuration of

TABLE 7: Assignment results of demand points to energy generation facilities according to requested energy types for  $D_c = 10$  hm (1000 m).

Demand points	Candidate facilities							
	1	2	3	4	5	6	7	8
1				No assignment	Elect, therm, cool			
2	Cool			No assignment			Elect, therm	
3			Elect, therm, cool	No assignment				
4			Cool	No assignment			Elect, therm	
5				No assignment		Elect, therm, cool		
6	Therm			No assignment	Cool		Elect	
7				No assignment		Elect, cool		Therm
8		Elect		No assignment	Therm, cool			
9				No assignment		Elect, cool		Therm
10		Elect, therm		No assignment	Cool			
11			Cool	No assignment			Elect, therm	
12	Therm, cool			No assignment				Elect
13	Elect, therm, cool			No assignment				
14				No assignment		Cool		Elect, therm
15	Elect, cool			No assignment		Therm		
16		Elect, therm		No assignment	Cool			
17	Cool			No assignment				Elect, therm
18		Elect, therm		No assignment	Cool			
19				No assignment		Elect, therm, cool		
20		Elect, therm		No assignment	Cool			

TABLE 8: Parameter ranges for test problems.

Parameters	Ranges
Coverage distance	200–1200 meters
Coordinates $(x, y)$	1–15
Unit transportation costs per hm	0.000001–0.000005 million € (for electricity) 0.00002–0.0006 million € (for thermal and cool energy)
Heat loss	3%
Electricity loss	0.5%

TABLE 9: Average solution results obtained by mathematical model for each problem set.

Pr. Set	No. of demand points	No. of candidate facilities	Avg. solution time (sec)	Avg. value for lower bound	Avg. value for upper bound	Percent deviation	Avg. iteration count	Avg. node number	Avg. Dc	No. of opt. results
1	5	2	0.03	9.15	9.15	0	28	4	7	10
2	10	4	1.10	17.69	17.69	0	27474	1530	6	10
3	15	6	447.11	31.05	31.05	0	5367735	201070	9	10
4	20	8	2853.8	34.62	34.3	0.92	20612336	562825	7	7
5	25	10	3911.5	43.02	42.17	1.66	18302425	281668	7	5
6	30	12	5773.5	59.1	57.14	3.27	28259836	511072	8	3
7	35	14	6647.2	63.68	61.01	4.16	27895929	440645	7	1
8	40	16	5760	68.23	65.27	4.39	24321810	325675	5	2
9	50	20	7200	105.22	97.48	7.99	25325270	472977	8	0
10	100	40	7200	203.79	181.94	12.01	11287922	188854	7	0

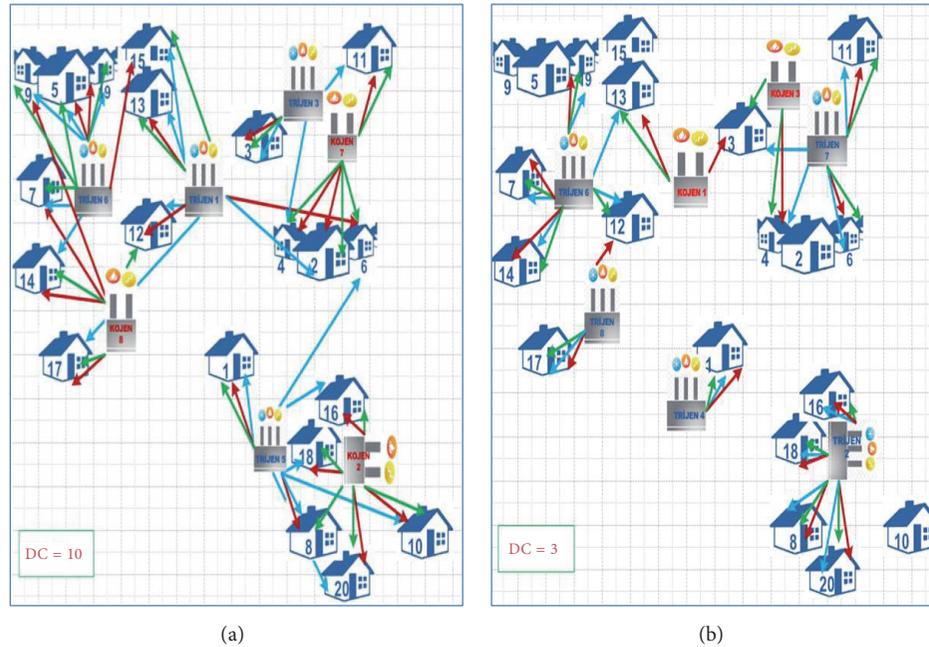


FIGURE 3: Schematic representation of assignment results for coverage distance of (a) 3 hm and (b) 10 hm.

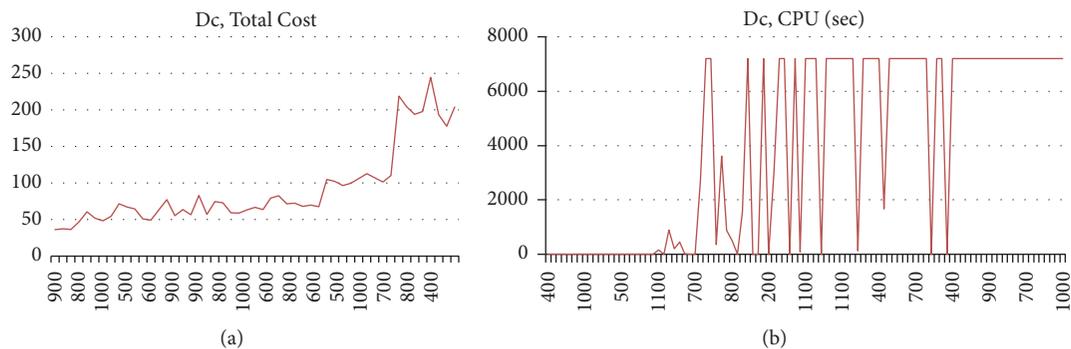


FIGURE 4: The relationship between the maximum coverage distance and both (a) total cost and (b) solution time.

distribution network on the base of annual demand and supply values.

In this paper, the design and assignment of local energy generation systems and distribution network were taken as a problem of facility location selection aimed at meeting all three-type energy demands with high efficiency. When energy demands of end users are met, first the distance between the facility and the user is considered and second the model is established on the basis of the allowable user-facility distance (coverage distance). The proposed MILP model allows the optimal layout of the DHCN with regard to the maximum coverage distance.

The most important issue to be regarded is whether the distance between the facility and the user is exceeding the maximum coverage distance or not. If it exceeds it, the user is assigned to another facility on condition that there is one that is closer. As the installation costs of DHCN constitute almost half of the whole system, the fewer the pipes are, the lower the

costs and losses will be. That is why the maximum coverage distance is added to the proposed model.

With this model, while the assignment of district-scale DER systems has been made with the least initial and transportation costs, the facility type (cogeneration/trigeneration), facility size (production capacity), and structure of district heating and cooling network are also decided, simultaneously. Moreover, both thermal losses and electricity transmission losses are taken into consideration.

The consistent land and energy planning approach is one of the basic principles of sustainable urban planning. The main aim of the general planning is to decide the land use and determine the infrastructure needs. Spatial planning is crucial to implement successful district heating and cooling systems because of the need for optimal infrastructure planning. As distributed generation is spreading, the land use and infrastructure requirements for distributed energy systems should be considered at the creation level of urban plans.

One of the advantages of this model is its capability to be used for general level of spatial urban planning by using annual values. Unlike the previous studies with seasonal demand changes almost implemented over a narrow area [13, 16, 20], this study would be appropriate to form a general framework as a view, especially for energy infrastructure planning, with the inclusion of much wider areas. As a real-world community contains hundreds of buildings, this model can be directly applied to different scale areas ranging from 5 to 100 demand points in any location with different types of buildings such as hospitals, hotels, shopping centers, schools, campuses, public buildings, and satellite towns and energy demands for general level of sustainable urban planning.

The numerical application of the model has been made on a sample with 20 demand points and 8 candidate facilities on an area of 150 hectares with different coverage distances. The developed model has given realistic solutions that can be used as a basis for the design of district-scale DER systems. In any case, the MILP model output has contained information to define the optimal design of the co/trigeneration units and distribution network, taking actual boundary conditions into account.

In the light of the examinations made, the installation costs of the district heating cooling networks have been seen to account for almost half of the regional energy system costs. As fewer pipelines are needed, costs and losses are reduced.

To have a projection, actual values of previously assumed fixed parameters could be used during the detailed system design for real-life applications on different scale areas. In addition, for the future works, the suggested MILP model can be integrated with renewable energy generation technologies, for example, photovoltaic, wind, or biomass systems, and the optimization model can be modified to determine how much distance should be set between the installed renewable energy facilities in order to obtain the minimum energy transportation costs.

The development of appropriate heuristic methods for problem dimensions that could not reach the optimal result is also conceived.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Authors' Contributions

Yeşim Ok constructed the model and designed the experiment case; Yeşim Ok and Mehmet Atak analyzed and evaluated the results together; Yeşim Ok wrote the paper.

## References

- [1] T. Ackermann, G. Andersson, and L. Söder, "Distributed generation: a definition," *Electric Power Systems Research*, vol. 57, no. 3, pp. 195–204, 2001.
- [2] H. Ren and W. Gao, "A MILP model for integrated plan and evaluation of distributed energy systems," *Applied Energy*, vol. 87, no. 3, pp. 1001–1014, 2010.
- [3] R. Priddle, "World energy outlook 2002, International Energy Agency, IEA/OECD, Paris 2002".
- [4] G. Chicco and P. Mancarella, "Matrix modelling of small-scale trigeneration systems and application to operational optimization," *Energy*, vol. 34, no. 3, pp. 261–273, 2009.
- [5] A. Arteconi, C. Brandoni, and F. Polonara, "Distributed generation and trigeneration: Energy saving opportunities in Italian supermarket sector," *Applied Thermal Engineering*, vol. 29, no. 8–9, pp. 1735–1743, 2009.
- [6] J. Wang, Z. J. Zhai, Y. Jing, and C. Zhang, "Optimization design of BCHP system to maximize to save energy and reduce environmental impact," *Energy*, vol. 35, no. 8, pp. 3388–3398, 2010.
- [7] J.-J. Wang, Y.-Y. Jing, and C.-F. Zhang, "Optimization of capacity and operation for CCHP system by genetic algorithm," *Applied Energy*, vol. 87, no. 4, pp. 1325–1335, 2010.
- [8] K. C. Kavvadias, A. P. Tosios, and Z. B. Maroulis, "Design of a combined heating, cooling and power system: Sizing, operation strategy selection and parametric analysis," *Energy Conversion and Management*, vol. 51, no. 4, pp. 833–845, 2010.
- [9] M. Carvalho, M. A. Lozano, L. M. Serra, and V. Wohlgemuth, "Modeling simple trigeneration systems for the distribution of environmental loads," *Environmental Modeling and Software*, vol. 30, pp. 71–80, 2012.
- [10] M. A. Lozano, J. C. Ramos, and L. M. Serra, "Cost optimization of the design of CHCP (combined heat, cooling and power) systems under legal constraints," *Energy*, vol. 35, no. 2, pp. 794–805, 2010.
- [11] C. Weber and N. Shah, "Optimisation based design of a district energy system for an eco-town in the United Kingdom," *Energy*, vol. 36, no. 2, pp. 1292–1308, 2011.
- [12] J. Ortiga, J. C. Bruno, and A. Coronas, "Selection of typical days for the characterisation of energy demand in cogeneration and trigeneration optimisation models for buildings," *Energy Conversion and Management*, vol. 52, no. 4, pp. 1934–1942, 2011.
- [13] E. D. Mehleri, H. Sarimveis, N. C. Markatos, and L. G. Papageorgiou, "A mathematical programming approach for optimal design of distributed energy systems at the neighbourhood level," *Energy*, vol. 44, no. 1, pp. 96–104, 2012.
- [14] E. D. Mehleri, H. Sarimveis, N. C. Markatos, and L. G. Papageorgiou, "Optimal design and operation of distributed energy systems: Application to Greek residential sector," *Journal of Renewable Energy*, vol. 51, pp. 331–342, 2013.
- [15] M. Ameri and Z. Besharati, "Optimal design and operation of district heating and cooling networks with CCHP systems in a residential complex," *Energy and Buildings*, vol. 110, pp. 135–148, 2016.
- [16] A. Omu, R. Choudhary, and A. Boies, "Distributed energy resource system optimisation using mixed integer programming," *Energy Policy*, vol. 61, pp. 249–266, 2013.
- [17] S. Bracco, G. Dentici, and S. Siri, "Economic and environmental optimization model for the design and the operation of a combined heat and power distributed generation system in an urban area," *Energy*, vol. 55, pp. 1014–1024, 2013.
- [18] S. Bracco, G. Dentici, and S. Siri, "DESOD : A mathematical programming tool to optimally design a distributed energy system," *Energy*, vol. 100, pp. 298–309, 2016.
- [19] Y. Yang, S. Zhang, and Y. Xiao, "An MILP (mixed integer linear programming) model for optimal design of district-scale distributed energy resource systems," *Energy*, vol. 90, pp. 1901–1915, 2015.

- [20] Y. Yang, S. Zhang, and Y. Xiao, "Optimal design of distributed energy resource systems coupled with energy distribution networks," *Energy*, vol. 85, pp. 433–448, 2015.
- [21] B. Morvaj, R. Evins, and J. Carmeliet, "Optimising urban energy systems: Simultaneous system sizing, operation and district heating network layout," *Energy*, vol. 116, pp. 619–636, 2016.
- [22] M. Sameti and F. Haghighat, "A two-level multi-objective optimization for simultaneous design and scheduling of a district energy system," *Applied Energy*, vol. 208, pp. 1053–1070, 2017.
- [23] J. F. Marquant, R. Evins, L. A. Bollinger, and J. Carmeliet, "A holarchic approach for multi-scale distributed energy system optimisation," *Applied Energy*, vol. 208, pp. 935–953, 2017.
- [24] X. Zheng, G. Wu, Y. Qiu et al., "A MINLP multi-objective optimization model for operational planning of a case study CCHP system in urban China," *Applied Energy*, vol. 210, pp. 1126–1140, 2018.
- [25] J. Current, M. Daskin, and D. Schilling, "Discrete network location models," *Facility Location: Applications And Theory*, vol. 1, pp. 81–118, 2002.
- [26] M. Sameti and F. Haghighat, "Optimization approaches in district heating and cooling thermal network," *Energy and Buildings*, vol. 140, pp. 121–130, 2017.
- [27] E. Jaraminiene, D. Bieka, E. Jaraminienė, and D. Biekša, *Energy Load Forecasting Model for Integrated Urban Energy Planning*, 2011.
- [28] C. Toregas, R. Swain, C. ReVelle, and L. Bergman, "The location of emergency service facilities," *Operations Research*, vol. 19, no. 6, pp. 1363–1373, 1971.
- [29] M. L. Balinski, "Integer programming: Methods, uses, computation," *Management Science*, vol. 12, pp. 253–313, 1965.

## Research Article

# Patch Based Collaborative Representation with Gabor Feature and Measurement Matrix for Face Recognition

Zhengyuan Xu,<sup>1</sup> Yu Liu,<sup>2</sup> Mingquan Ye,<sup>3</sup> Lei Huang,<sup>1</sup> Hao Yu,<sup>4</sup> and Xun Chen <sup>5</sup>

<sup>1</sup>Department of Medical Engineering, Wannan Medical College, Wuhu 241002, China

<sup>2</sup>Department of Biomedical Engineering, Hefei University of Technology, Hefei 230009, China

<sup>3</sup>School of Medical Information and Research Center of Health Big Data Mining and Applications, Wannan Medical College, Wuhu 241002, China

<sup>4</sup>Anhui Province Key Laboratory of Active Biological Macro-Molecules Research, Central Laboratory of Wannan Medical College, Wuhu 241002, China

<sup>5</sup>Department of Electronic Science and Technology, University of Science and Technology of China, Hefei 230027, China

Correspondence should be addressed to Xun Chen; [xunchen@ece.ubc.ca](mailto:xunchen@ece.ubc.ca)

Received 13 December 2017; Accepted 29 March 2018; Published 10 May 2018

Academic Editor: Nibaldo Rodríguez

Copyright © 2018 Zhengyuan Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, sparse representation based classification (SRC) has emerged as a popular technique in face recognition. Traditional SRC focuses on the role of the  $l_1$ -norm but ignores the impact of collaborative representation (CR), which employs all the training examples over all the classes to represent a test sample. Due to issues like expression, illumination, pose, and small sample size, face recognition still remains as a challenging problem. In this paper, we proposed a patch based collaborative representation method for face recognition via Gabor feature and measurement matrix. Using patch based collaborative representation, this method can solve the problem of the lack of accuracy for the linear representation of the small sample size. Compared with holistic features, the multiscale and multidirection Gabor feature shows more robustness. The usage of measurement matrix can reduce large data volume caused by Gabor feature. The experimental results on several popular face databases including Extended Yale B, CMU\_PIE, and LFW indicated that the proposed method is more competitive in robustness and accuracy than conventional SR and CR based methods.

## 1. Introduction

Face recognition (FR) is one of the most classical and challenging problems in pattern classification, computer vision, and machine learning [1]. Although face recognition technology has made a series of achievements, it still confronts many challenges caused by the variations of illumination, pose, facial expression, and noise in real-world [2, 3]. In real applications, the small sample size problem of FR is also a more difficult issue due to the limitations in availability of training samples.

In terms of classification schemes, several widespread pattern classification methods are used in FR. Generally, there are two types of pattern classification methods [4, 5]: parametric methods and nonparametric methods. Parametric methods such as support vector machine (SVM) [6, 7]

center on how to learn the parameters of a hypothesis classification model from the training samples and then use them to identify the class labels of test samples. In contrast, the nonparametric methods, such as nearest neighbor (NN) [8] and nearest subspace (NS) [9], use the training samples directly to identify the class labels of test samples. Recent works have revealed an advantage by the nonparametric methods over the parametric methods [4, 10, 11]. The distance based classifiers are widely used in nonparametric methods for FR [11], such as the nearest subspace classifier (NSC) [12]. A key issue in distance based nonparametric classifiers is how to represent the test sample [4]. Recently, Wright et al. pioneered by using the SRC for robust FR [13]. First the test sample was sparsely coded by the training samples; then the class labels of test samples were identified by choosing which class yields the smallest coding error. Although sparse

representation related methods [13, 14] achieved a great success in FR, those methods focus on the role of the  $l_1$ -norm while ignoring the role of CR [15], which uses all classes of training samples to represent the target test sample. In this study [16], Zhu et al. argued that both SRC and collaborative representation based classifier (CRC) suffer serious performance degradation when the training sample size is very small, because the test sample cannot be well represented. In order to solve the small sample size problem, they proposed to conduct CRC on the patch and named it the patch based CRC (PCRC).

The PCRC and some related works [17, 18] have demonstrated their effects on small sample size problem of FR; however, some key issues remain to be further optimized. On one hand, all the PCRC related works used the original face feature, but the original feature cannot effectively handle the variations of illumination, pose, facial expression, and noise [19]. On the other hand, the data redundancy existing in these methods leads to poor performance in classification accuracy and computational cost. The face feature problem has been noticed by some recent works in which an efficient and effective image representation has been proposed by using local and holistic features. The Eigenface [20, 21], Randomface [22], and Fisherface [21] are all classical holistic features [23], but some other works argued that those holistic features can be easily affected by variables such as illumination, pose, facial expression, and noise. Therefore, they introduced some local features such as LBP [24] and Gabor filter [25, 26]. Gabor filter has been successfully and widely used in FR [26, 27]. Gabor feature could effectively extract the face local features at multiple scales and multiple directions; however, this may lead to a sharp rise in data volumes [28]. The key to solving the large data volume problem is dimensionality reduction. Numerous dimensionality reduction methods have been put forward to find projections that better separate the classes in low-dimensional spaces, among which linear subspace analysis method has received more and more attention owing to its good properties, including principal component analysis (PCA) [28], linear discriminant analysis (LDA) [29], and independent component correlation algorithm (ICA) [30]. A lot of works showed that PCA has the optimal performance in FR [31–33].

In this paper, we first attempted to alleviate the influence of unreliability environment on small sample size in FR; therefore, we proposed to use Gabor feature and applied it to PCRC. Then to improve the computational efficiency of GPCRC, we proposed to use PCA for dimension reduction and then use the measurement matrix, including Random Gaussian matrices [34], Toeplitz and cyclic matrices [35], and Deterministic sparse Toeplitz matrices ( $\Delta = 4, 8, 16$ ) [36], to reduce the dimension of the transformed signal and used the low-dimensional data to accurately represent the face. The experimental results showed that the GPCRC and its improved methods are effective.

Section 2 briefly reviewed SRC and CRC. Section 3 described the proposed GPCRC and its improved methods. Section 4 illustrated the experiments and the results. And Section 5 concluded the paper.

## 2. Related Works

*2.1. Sparse Representation Based Classification.* Recently, SRC was first reported by Wright et al. [13] for robust FR. In SRC, let  $\mathbf{x}_i \in \mathbf{R}^{m \times n}$  denote the  $i$ -th face dataset, and each column of  $\mathbf{x}_i$  is a sample of the face of the  $i$ -th individual. Assuming there are  $k$  classes of face samples, let  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k]$ . When identifying a target face test sample,  $\mathbf{y} \approx \mathbf{X}\hat{\boldsymbol{\alpha}}$  is used for coding,  $\hat{\boldsymbol{\alpha}} = [\hat{\boldsymbol{\alpha}}_1, \hat{\boldsymbol{\alpha}}_2, \dots, \hat{\boldsymbol{\alpha}}_k]$ , and the coefficient vector  $\hat{\boldsymbol{\alpha}}_i$  is the encoding of the  $i$ -th individual sample. If  $\mathbf{y}$  is from the  $i$ -th class, then  $\mathbf{y} \approx \mathbf{x}_i\hat{\boldsymbol{\alpha}}_i$  is the best reservation, which means that a large number of coefficients are close to zero in  $\hat{\boldsymbol{\alpha}}_k$  ( $k \neq i$ ); only  $\hat{\boldsymbol{\alpha}}_i$  remains intact. Thus, the classification (ID) of the target face test sample  $\mathbf{y}$  can be decoded by the sparse nonzero coefficient in  $\hat{\boldsymbol{\alpha}}$ .

The SRC methods [13] are summarized as follows.

*Step 1.* Given  $k$ -class face training sample  $\mathbf{X}$  and test sample  $\mathbf{y}$ .

*Step 2* (dimension adjustment).  $\mathbf{X}$ ,  $\mathbf{y}$  are projected onto the corresponding low-dimensional feature space  $\tilde{\mathbf{X}}$ ,  $\tilde{\mathbf{y}}$  using the traditional dimensionality (PCA) reduction technique.

*Step 3* ( $l_2$ -norm). Obtain the normalized  $\hat{\tilde{\mathbf{X}}}$  column and  $\hat{\tilde{\mathbf{y}}}$ , respectively.

*Step 4.* Solve the  $l_1$ -minimization problem.  
 $(\hat{\boldsymbol{\alpha}}) = \arg \min \|\boldsymbol{\alpha}\|_1$  satisfies  $\hat{\tilde{\mathbf{X}}}\boldsymbol{\alpha} = \hat{\tilde{\mathbf{y}}}$  or  $\|\hat{\tilde{\mathbf{X}}}\boldsymbol{\alpha} - \hat{\tilde{\mathbf{y}}}\|_2 < \varepsilon$ .

*Step 5.* Compute the residuals to identify the following:  
 $r_i(\mathbf{y}) = \|\mathbf{y} - \mathbf{x}_i\hat{\boldsymbol{\alpha}}_i\|_2$ ,  $\text{ID}(\mathbf{y}) = \arg \min_i [r_i(\mathbf{y})]$ ,  $i = 1, 2, \dots, k$ .

*2.2. Collaborative Representation Based Classification.* Zhang et al. [15] argued that it is the CR but not the  $l_1$ -norm sparsity that makes SRC powerful for face classification. Collaborative representation uses all classes (individuals) of training samples to label the target test samples. In order to reduce the complexity of face detection by  $\mathbf{X}$ -coordinate, a regularized least-squares method is proposed by Zhang et al., which is

$$(\hat{\boldsymbol{\rho}}) = \arg \min_{\boldsymbol{\rho}} \{ \|\mathbf{y} - \mathbf{X} \cdot \boldsymbol{\rho}\|_2^2 + \lambda \|\boldsymbol{\rho}\|_2^2 \}, \quad (1)$$

where  $\lambda$  is a regularization parameter. Equation (1) has dual roles, first of which is stabilizing the least-squares method, and, secondly, it proposes a “sparsity” which is much weaker than  $l_1$ -norm to solve  $\hat{\boldsymbol{\rho}}$ . The CR for the regularized least-squares method of (1) can be solved as follows:

$$\hat{\boldsymbol{\rho}} = (\mathbf{X}^T \mathbf{X} + \lambda \cdot \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}. \quad (2)$$

Let  $\mathbf{P} = (\mathbf{X}^T \mathbf{X} + \lambda \cdot \mathbf{I})^{-1} \mathbf{X}^T$ . Obviously, since  $\mathbf{P}$  and  $\mathbf{y}$  have little relevance, they can be precalculated as a projection matrix. When a target test sample  $\mathbf{y}$  comes in to be identified,  $\mathbf{y}$  is projected onto  $\mathbf{P}$  by  $\mathbf{P}\mathbf{y}$ , thus making CR very fast. The classification by  $\hat{\boldsymbol{\rho}}$  is very similar to the classification by  $\hat{\boldsymbol{\alpha}}$  in the SRC method. In addition to representing the classification residuals  $\|\mathbf{y} - \mathbf{x}_i\hat{\boldsymbol{\rho}}_i\|_2$ ,  $\hat{\boldsymbol{\rho}}_i$  is a coefficient vector

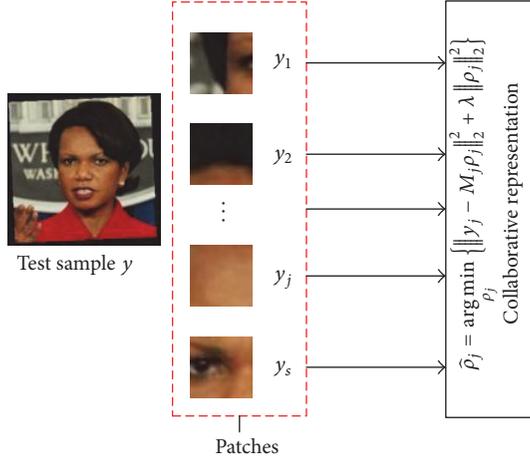


FIGURE 1: Patch based collaborative representation.

associated with class  $i$ ; the  $l_2$ -norm “sparsity”  $\|\hat{\rho}_i\|_2$  also contains abundant information for classification. The CRC method [15] is summarized as follows.

*Step 1.* Give  $k$  classes of face training sample  $\mathbf{X}$  and test sample  $\mathbf{y}$ .

*Step 2* (reduce the dimension). Use PCA to reduce  $\mathbf{X}$  and  $\mathbf{y}$  to low-dimensional feature space and obtain  $\hat{\mathbf{X}}$  and  $\hat{\mathbf{y}}$ .

*Step 3* (normalization). Normalize the columns of  $\hat{\mathbf{X}}$  and  $\hat{\mathbf{y}}$  using the unit  $l_2$ -norm.

*Step 4.* Encode  $\hat{\mathbf{y}}$  on  $\hat{\mathbf{X}}$ :

$$\mathbf{P} = (\mathbf{X}^T \mathbf{X} + \lambda \cdot \mathbf{I})^{-1} \mathbf{X}^T, \hat{\rho} = \mathbf{P} \hat{\mathbf{y}}.$$

*Step 5.* Compute the regularization for identification:

$$r_i(\mathbf{y}) = \|\mathbf{y} - \mathbf{x}_i \hat{\rho}_i\|_2 / \|\hat{\rho}_i\|_2; \text{ID}(\mathbf{y}) = \arg \min_i [r_i(\mathbf{y})], i = 1, 2, \dots, k.$$

**2.3. Patch Based Collaborative Representation.** In the equation of sparse representation and collaborative representation, it can be seen that if the linear system determined by the training dictionary  $\mathbf{X}$  is underdetermined, the linear representation of the target test sample  $\mathbf{y}$  over  $\mathbf{X}$  can be very accurate, but in reality available samples in each target are limited; the sparse representation and the cooperative representation method may fail because the linear representation of the target test sample may not be accurate enough. In order to alleviate this problem, Zhu et al. proposed a PCRC method for FR [16], as shown in Figure 1, the target face test sample  $\mathbf{y}$  is divided into a set of overlapping face image patches  $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_s$  according to the patch size. Each of the divided face image patches  $\mathbf{y}_j$  is collaboratively represented on the local dictionary  $\mathbf{M}_j$  at the corresponding position of the patch  $\mathbf{y}_j$  extracted from  $\mathbf{X}$ . In this case, since the linear system determined by the local dictionary  $\mathbf{M}_j$  is often underdetermined, the patch based representation is more accurate than the overall face image representation.

### 3. Patch Based Collaborative Representation Using Gabor Feature and Measurement Matrix for Face Recognition

**3.1. Gabor Feature.** Gabor feature has been widely used in FR because of its robustness in illumination, expression, and pose compared to holistic feature. Yang and Zhang have applied multiscale and multidirectional Gabor dictionary to the SRC for FR [19], which further improves the robustness of the algorithm. Inspired by previous works, in this paper, we integrate Gabor feature into the PCRC framework to improve its robustness.

A Gabor filter with multidirection  $\mu$  and multiscale  $\nu$  is defined as follows [25, 26]:

$$\Psi_{\mu,\nu}(\mathbf{z}) = \frac{\|\mathbf{k}_{\mu,\nu}\|^2}{\delta^2} e^{(-\|\mathbf{k}_{\mu,\nu}\|^2 \|\mathbf{z}\|^2 / 2\delta^2)} \left[ e^{i\mathbf{k}_{\mu,\nu} \cdot \mathbf{z}} - e^{-\delta^2/2} \right] \quad (3)$$

$$\mathbf{k}_{\mu,\nu} = \mathbf{k}_\nu \cdot e^{i\nu\mu}, \quad \psi_\mu = \frac{\pi\mu}{8}, \quad \mathbf{k}_\nu = \frac{k_{\max}}{f^\nu},$$

where the coordinates of the pixel are  $\mathbf{z} = (x, y)$ ,  $k_{\max}$  is the maximum frequency, and the interval factor of the kernel function distance is denoted as  $f$ . The bandwidth of the filter is determined by  $\delta$ . The convolution of the target image  $\text{Img}$  and the wavelet kernel  $\Psi_{\mu,\nu}$  is expressed as

$$\mathbf{G}_{\mu,\nu} = \text{Img}(\mathbf{z}) * \Psi_{\mu,\nu}(\mathbf{z}) = \mathbf{M}_{\mu,\nu} \cdot \exp(i\theta_{\mu,\nu}(\mathbf{z})), \quad (4)$$

where  $\mathbf{M}_{\mu,\nu}$  ( $\mu = 0, 1, \dots, 7, \nu = 0, 1, \dots, 4$ ) is the amplitude of Gabor,  $\theta_{\mu,\nu}(\mathbf{z})$  is the phase of Gabor, and the local energy change in the image is expressed by amplitude information. Because the Gabor phase changes periodically with the space position and the amplitude is relatively smooth and stable [19, 25, 26], only the magnitude of Gabor was used in this paper, such as Figure 2.

**3.2. Measurement Matrix.** Unfortunately, although Gabor feature can be used to enhance the robustness of face image representation, it brings higher dimensions to the training sets than holistic feature does. In other words, the computation cost and computation time are increased. In order to solve the problem caused by higher dimension of the training sets, a further dimension reduction is necessary. We proposed to use PCA [31–33] for our method. The steps of PCA are as follows: assuming  $N$  sample images  $\mathbf{x}_i \in \mathbf{R}^m$ ,  $i = 1, 2, \dots, N$ . Firstly, normalize each sample (subtract the mean, and then divide the variance), convert vector  $\mathbf{x}_i^*$ , which accord with normal distribution  $(0, 1)$ . Secondly, compute the eigenvectors of covariance matrix  $\mathbf{Q} = \mathbf{X}\mathbf{X}^T$ ,  $\mathbf{X} = [\mathbf{x}_1^*, \mathbf{x}_2^*, \dots, \mathbf{x}_N^*]$ :  $\lambda \mathbf{E} = \mathbf{Q}\mathbf{E}$ ,  $\mathbf{E}$  is the eigenvector corresponding to the eigenvalue  $\lambda$ . Thirdly, the eigenvector is sorted according to the size of the eigenvalues; the first  $p$  eigenvectors are extracted to form a linear transformation matrix  $\mathbf{W}_{\text{PCA}} \in \mathbf{R}^{m \times p}$ . We can use  $\mathbf{y}_i = \mathbf{W}_{\text{PCA}}^T \mathbf{x}_i$  to reduce dimension. But, if  $m \gg N$ , the dimension of  $\mathbf{X}\mathbf{X}^T \in \mathbf{R}^{m \times m}$  would be very large. In order to solve this problem, singular value decomposition is usually used. The eigenvalues

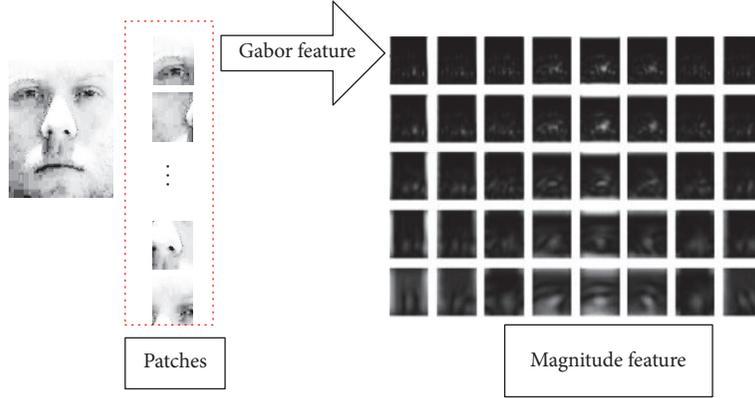


FIGURE 2: Patch based collaborative representation using Gabor feature.

$\lambda$  and eigenvectors  $\mu_i$  of  $XX^T$  are obtained by calculating the eigenvalues  $\delta_i$  and eigenvectors  $v_i$  of  $X^T X$  (the first  $p$ ):  $\mu_i = (1/\sqrt{\delta_i})Xv_i, i = 1, 2, \dots, p$ .

In summary, we found that PCA and its related algorithms have two obvious shortcomings [37, 38]. Firstly, the leading eigenvectors encode mostly illumination and expression, rather than discriminating information. Secondly, in the actual calculation, the amount of calculation is very large and will fail in small samples.

Inspired by the method of using random face for feature extraction illustrated in this literature [38], we used the Random Gaussian matrices  $\Phi \in \mathbf{R}^{M \times N}$  ( $M \ll N$ ) as a measurement matrix to measure face images. The measurement matrix  $\Phi$  is used to measure the redundant dictionary  $\mathbf{X}$  to obtain  $A = \Phi\mathbf{X} \in \mathbf{R}^{M \times S}$ , where  $\mathbf{X} = [x_1, x_2, \dots, x_S] \in \mathbf{R}^{N \times S}$ . In Figure 3(a), for any test image  $x$ , measurements  $y$  were obtained by  $y = \Phi x \in \mathbf{R}^{M \times 1}$ . In essence, utilizing the measurement matrix to reduce the dimension of the image is different from the sparse representation theory. The dimension  $M$  of the measurements  $y$  was measured by the measurement matrix  $\Phi$  and was not limited by the number of training samples.

However, some literatures [35, 39] suggested that the Random Gaussian matrices are uncertain and limit its practical application. And Toeplitz and cyclic matrices were proposed for signal reconstruction. The Toeplitz and cyclic matrices rotate the row vectors to generate all matrices. Usually, the value of the vector in Toeplitz and cyclic matrices is  $\pm 1$ , and each element is independent of the Bernoulli distribution. Therefore, it is easy to implement hardware in practical application. Based on the above analysis, we further used Toeplitz and cyclic matrices and their improved method (namely, the Deterministic sparse Toeplitz matrices [36]) for our method. Some of the relevant measurement matrices used in this paper will be described in detail. The operating mechanism of each measurement matrix is shown in Figure 3.

**3.2.1. Random Gaussian Matrices.** The format of  $m \times n$  Random Gaussian matrices is expressed as follows [34, 38]:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \quad (5)$$

each element  $a_{i,j}$  is independently subject to a Gaussian distribution whose mean is 0 and variance is  $1/m$ .

**3.2.2. Toeplitz and Cyclic Matrices.** The concrete form of  $M \times N$  Toeplitz and cyclic matrices [35, 39] is presented below:

$$\begin{bmatrix} a_N & a_{N-1} & \cdots & a_1 \\ a_{N+1} & a_N & \cdots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{N+M-1} & a_{N+M-2} & \cdots & a_M \end{bmatrix}, \quad (6)$$

$$\begin{bmatrix} a_N & a_{N-1} & \cdots & a_1 \\ a_1 & a_N & \cdots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{M-1} & a_{M-2} & \cdots & a_M \end{bmatrix}. \quad (7)$$

Equation (6) is the Toeplitz matrices; the main diagonal  $a_{i,j} = a_{i+1,j+1}$  are constants. If  $a_i = a_{N+i}$ , the additional condition of (6) is satisfied; then it becomes a cyclic matrices, and its element  $\{a_i\}_{i=1}^{N+M-1}$  follows a certain probability distribution  $P(a)$ .

**3.2.3. Deterministic Sparse Toeplitz Matrices.** The construction of Deterministic sparse Toeplitz matrices [36] is based on the Toeplitz and cyclic matrices, which is illustrated in this paper by the example of the random spacing Toeplitz matrices with an interval of  $\Delta = 2$ . The independent elements in the first row and the first column of (6) constitute the vector  $\mathbf{T}$ :

$$[a_N \ a_{N-1} \ \cdots \ a_1 \ a_{N+1} \ \cdots \ a_{N+M-1}]. \quad (8)$$

Conducting random sparse spacing to (7), one can see that  $T$  contains all the independent elements in (8). Then

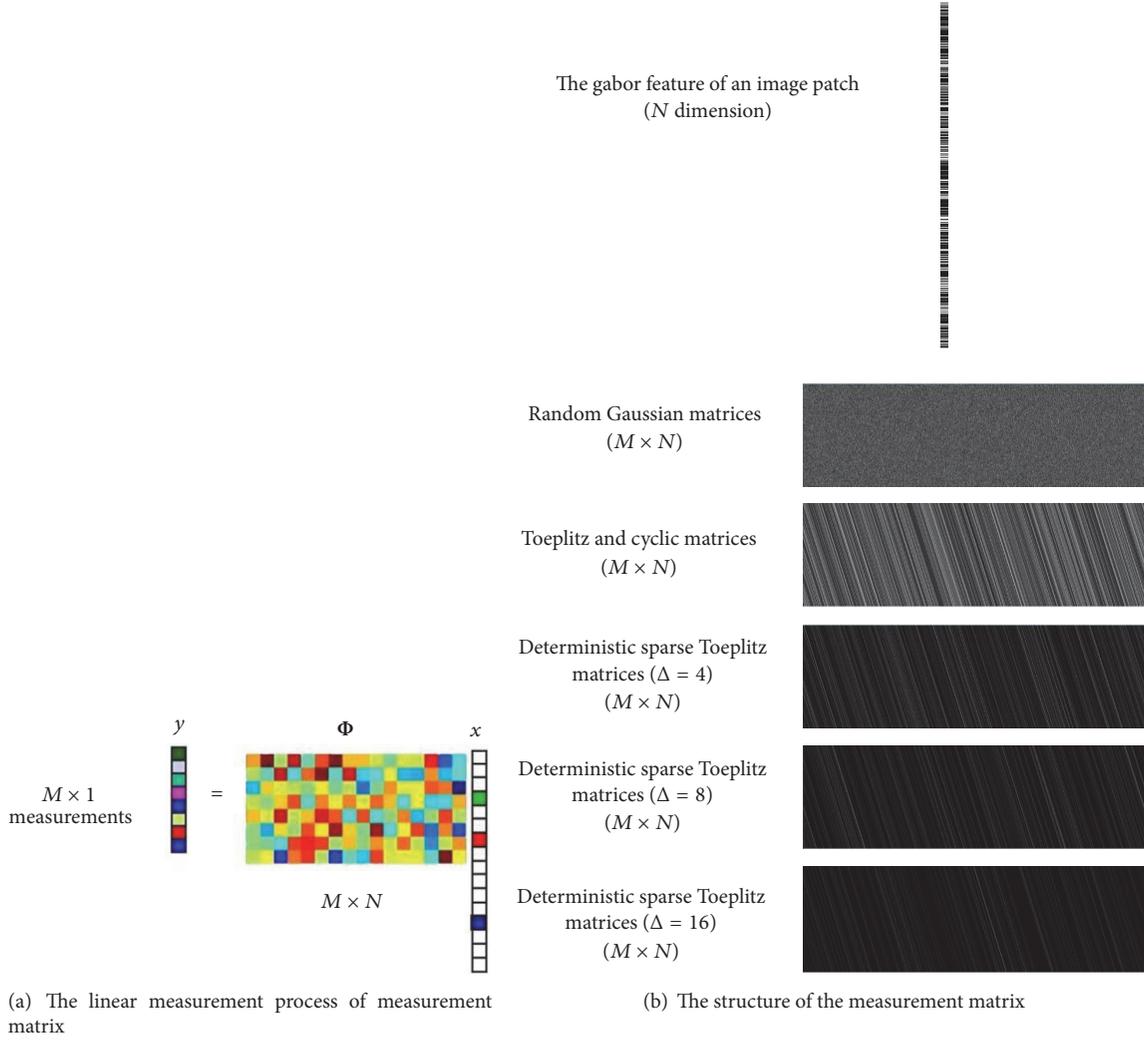


FIGURE 3: The operating mechanism of measurement matrix.

assignment operations are performed on  $T$ , where the elements  $a_i$  ( $i \in \Lambda$ ,  $\Lambda$  is the  $(N + M - 1)/\Delta$  indexes randomly chosen from the index sequence  $1 \rightarrow N + M - 1$ ) obey the independent and identically distributed (i.i.d.) Random Gaussian distribution, while the other elements are 0. Finally, we obtained the Deterministic sparse Toeplitz matrices  $a_{i+1,j+1} = a_{i,j}$  according to the characteristics of construction of the Toeplitz and cyclic matrices.

**3.3. The Proposed Face Recognition Approach.** Although the PCRC can indeed solve the problem of small sample size, this method is still based on the original feature of the patch, and the robustness and accuracy are yet to be improved. Based on the above analysis, Gabor feature and measurement matrix are infused into the PCRC for FR, which not only solves the problem of small sample size but also enhances the robustness and efficiency of the method.

Our proposed method for FR is summarized as follows.

*Step 1* (input). Face training sample  $\mathbf{X}$  and test sample  $\mathbf{y}$ .

*Step 2* (patch). Divide the  $k$  face training samples  $\mathbf{X}$  into  $s$  patches and divide the test sample  $\mathbf{y}$  into  $s$  patches, where  $\mathbf{y}_i$  and  $\mathbf{M}_{ij}$  are the position corresponding to the sample patches:

$$\mathbf{X} = \begin{bmatrix} \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_i \\ \vdots \\ \mathbf{M}_s \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{11} & \cdots & \mathbf{M}_{1j} & \cdots & \mathbf{M}_{1k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{M}_{i1} & \cdots & \mathbf{M}_{ij} & \cdots & \mathbf{M}_{ik} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{M}_{s1} & \cdots & \mathbf{M}_{sj} & \cdots & \mathbf{M}_{sk} \end{bmatrix}, \quad (9)$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_i \\ \vdots \\ \mathbf{y}_s \end{bmatrix}.$$

*Step 3* (extract and measure features). (1) Extract Gabor feature from patches of training samples and test sample, respectively, to obtain

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_1 \\ \vdots \\ \mathbf{G}_i \\ \vdots \\ \mathbf{G}_s \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{11} & \cdots & \mathbf{G}_{1j} & \cdots & \mathbf{G}_{1k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{G}_{i1} & \cdots & \mathbf{G}_{ij} & \cdots & \mathbf{G}_{ik} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{G}_{s1} & \cdots & \mathbf{G}_{sj} & \cdots & \mathbf{G}_{sk} \end{bmatrix}, \quad (10)$$

$$\mathbf{Gy} = \begin{bmatrix} \mathbf{Gy}_1 \\ \vdots \\ \mathbf{Gy}_i \\ \vdots \\ \mathbf{Gy}_s \end{bmatrix}.$$

(2) Carry out the measurement and reduce the dimension of each patch  $\mathbf{G}_{ij}$  ( $i = 1, 2, \dots, s$ ,  $j = 1, 2, \dots, k$ ) using the measurement matrix  $\Phi$ , to obtain measured signal:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_i \\ \vdots \\ \mathbf{A}_s \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{11} & \cdots & \mathbf{a}_{1j} & \cdots & \mathbf{a}_{1k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{a}_{i1} & \cdots & \mathbf{a}_{ij} & \cdots & \mathbf{a}_{ik} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{a}_{s1} & \cdots & \mathbf{a}_{sj} & \cdots & \mathbf{a}_{sk} \end{bmatrix}$$

$$= \begin{bmatrix} \Phi \mathbf{G}_{11} & \cdots & \Phi \mathbf{G}_{1j} & \cdots & \Phi \mathbf{G}_{1k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Phi \mathbf{G}_{i1} & \cdots & \Phi \mathbf{G}_{ij} & \cdots & \Phi \mathbf{G}_{ik} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Phi \mathbf{G}_{s1} & \cdots & \Phi \mathbf{G}_{sj} & \cdots & \Phi \mathbf{G}_{sk} \end{bmatrix}, \quad (11)$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_i \\ \vdots \\ \mathbf{b}_s \end{bmatrix} = \begin{bmatrix} \Phi \mathbf{Gy}_1 \\ \vdots \\ \Phi \mathbf{Gy}_i \\ \vdots \\ \Phi \mathbf{Gy}_s \end{bmatrix}.$$

*Step 4*. Collaborative representation of each patch measurement signal  $\mathbf{b}_i$  of the test sample over the training sample measurement signal local dictionary  $\mathbf{A}_i$ :

$$\hat{\rho}_i = \arg \min_{\rho_i} \left\{ \|\mathbf{b}_i - \mathbf{A}_i \rho_i\|_2^2 + \lambda \|\rho_i\|_2^2 \right\} \quad i = 1, 2, \dots, s. \quad (12)$$

Equation (12) can be easily derived as follows:

$$\hat{\rho}_i = \mathbf{P} \mathbf{b}_i, \quad \mathbf{P} = (\mathbf{A}_i^T \mathbf{A}_i + \lambda \mathbf{I})^{-1} \mathbf{A}_i^T \quad i = 1, 2, \dots, s, \quad (13)$$

where  $\mathbf{I}$  is unit matrix.

*Step 5*. Recognition results of patch  $\mathbf{y}_i$  are

$$r_{ij}(\mathbf{y}_i) = \frac{\|\mathbf{b}_i - \mathbf{a}_{ij} \hat{\rho}_i\|_2}{\|\hat{\rho}_i\|_2} \quad i = 1, 2, \dots, s, \quad j = 1, 2, \dots, k, \quad (14)$$

$$\text{ID}(r_{ij}) = \arg \min_j [r_{ij}(\mathbf{y}_i)].$$

*Step 6*. Use voting to obtain the final recognition result.

## 4. Experimental Analysis

The proposed approaches, GPCRC and its improved methods, were evaluated in three publicly available databases: Extended Yale B [40–42], CMU\_PIE [43, 44], and LFW [45, 46]. To show the effectiveness of the GPCRC, we compared our methods with four classical methods and their improved methods, namely, SRC [13], CRC [15], SSRC [14], and PCRC [16]. Then, we tested our GPCRC using several measurement matrix, namely, Random Gaussian matrices, Toeplitz, and cyclic matrices and Deterministic sparse Toeplitz matrices ( $\Delta = 4, 8, 16$ ) to reduce the dimension of the transformed signal and compare their performance with the conventional PCA methods. In all the following experiments, we ran the MATLAB 2015b on a typical Intel(R) Core(TM) i5-3470 CPU 3.20 GHz, Windows7 x64 PC. In our implementation of Gabor filters, the parameters were set as  $f = \sqrt{2}$ ,  $K_{\max} = \pi/2$ ,  $\sigma = \pi$ ,  $\mu = 0, 1, \dots, 7$ ,  $\nu = 0, 1, \dots, 4$  for all the experiments below. All the programs were run 20 times on each database and the mean and variance of each method were reported. In order to report the best result of each method, the parameter  $\lambda$  used in SRC, SSRC, CRC, and PCRC was set to 0.001 [13], 0.001 [14], 0.005 [15], and 0.001 [16, 17], respectively. The patch sizes were used in PCRC and GPCRC and its improved methods are  $8 \times 8$  [16, 17] and  $16 \times 16$ , respectively.

*4.1. Extended Yale B.* To test the robustness of the proposed method on illumination, we used the classic Extended Yale B database [40–42], because faces from Extended Yale B database were acquired in different illumination conditions. The Extended Yale B database contains 38 human subjects under 9 poses and 64 illumination conditions. For obvious comparison, all frontal-face images marked with P00 were used in our experiment, and the face images size was down-sampled to  $32 \times 32$ . 10, 20 faces from each individual were selected randomly as training sample; 30 others from each individual were selected as test sample. Figure 4 shows some P00 marked samples from the Extended Yale B database. The experimental results of each method are shown in Table 1. In Figure 5 we compared the recognition rates of

TABLE 1: Recognition rate (%) on Extended Yale B database.

Features	Methods	Training sample size	
		10	20
Intensity feature	SRC	76.15 ± 9.60	80.50 ± 4.32
	CRC	78.60 ± 10.58	83.00 ± 6.64
	SSRC	87.53 ± 7.91	90.01 ± 4.38
	PCRC	90.44 ± 1.09	94.25 ± 1.01
LBP feature	SRC	83.56 ± 11.32	89.74 ± 9.52
	CRC	83.02 ± 10.19	90.30 ± 3.27
	SSRC	90.58 ± 6.39	95.89 ± 3.01
	PCRC	79.67 ± 12.57	87.25 ± 5.08
Gabor feature	SRC	85.25 ± 5.36	91.50 ± 3.50
	CRC	86.18 ± 4.43	90.00 ± 2.80
	SSRC	87.06 ± 9.49	92.24 ± 5.06
	PCRC	<b>94.34 ± 2.06</b>	<b>98.83 ± 0.94</b>



FIGURE 4: Some marked samples of the Extended Yale B database.

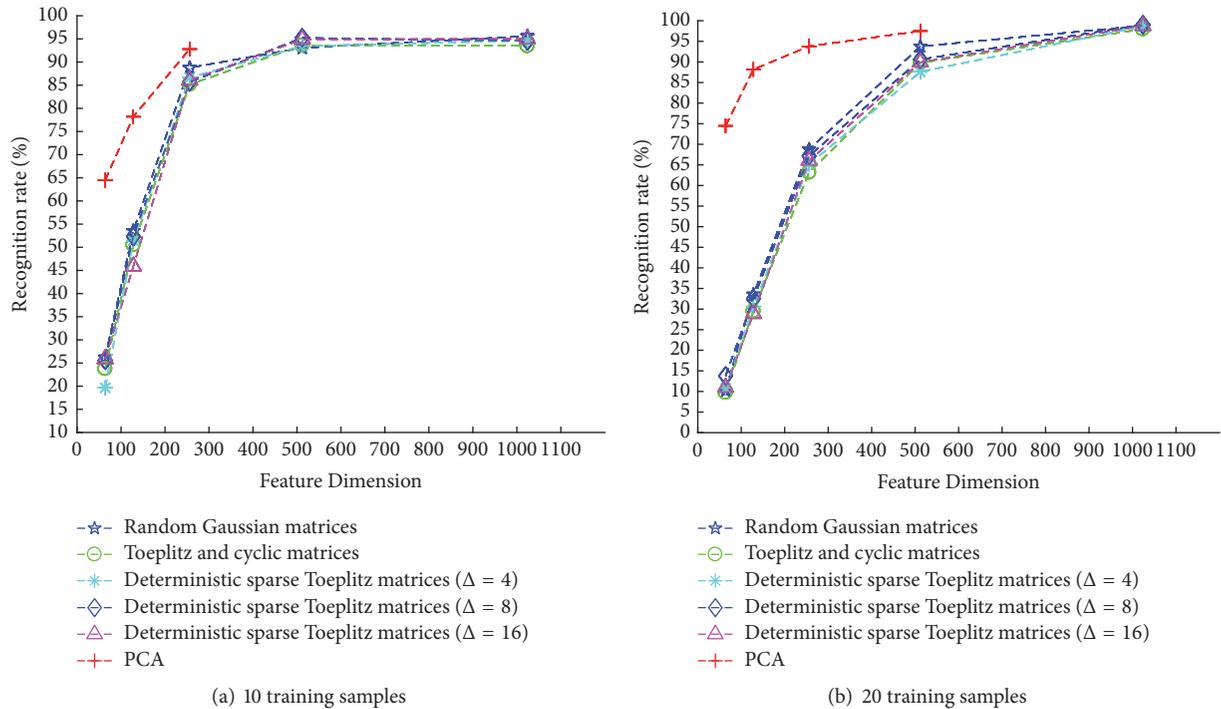


FIGURE 5: The performance of each dimension reduction method for GPCRC on Extended Yale B database.

TABLE 2: The speed of each dimension reduction methods.

Method	Random Gaussian	Toeplitz and cyclic	PCA
Time(s)	1.1131	1.0836	41.7295
Method	Deterministic sparse Toeplitz		
	( $\Delta = 4$ )	( $\Delta = 8$ )	( $\Delta = 16$ )
Time(s)	1.2643	1.2226	1.2678



FIGURE 6: Some marked samples of the CMU\_PIE database.

various methods with the space dimensions 32, 64, 128, 256, 512, and 1024 for each patch in GPCRC, and those numbers correspond to downsampling ratios of 1/320, 1/160, 1/80, 1/40, 1/20, and 1/10 respectively. In Table 1, it can be clearly seen that GPCRC achieves the highest recognition rate performance in all experiments. In Figure 6, we can see the performance of various dimensionality reduction methods for GPCRC. 10 samples of each individual in Figure 5(a) were used as training samples. When the feature dimension is low ( $\leq 256$ ), the best performance of PCA is 92.80% (256 dimension), which is significantly higher than that of the measurement matrix at 256 dimension: Random Gaussian matrices are 88.77%, Toeplitz and cyclic matrices are 85.17%, Deterministic sparse Toeplitz matrices ( $\Delta = 4$ ) are 86.67%, Deterministic sparse Toeplitz matrices ( $\Delta = 8$ ) are 85.69%, and Deterministic sparse Toeplitz matrices ( $\Delta = 16$ ) are 86.05%, but none of these has been as good as the performance of the original dimension (10240 dimension). So the reason why the performance of PCA is significantly better than the measurement matrix at low dimension can be analyzed through their operation mechanism: the PCA transforms the original data into a set of linearly independent representations of each dimension in a linear transformation, which can extract the principal component of the data. These principal components can represent image signals more accurately [31–33]. The theory of compressed sensing [34–36] states that when the signal is sparse or compressible in a transform domain, the measurement matrix which is noncoherent with the transform matrix can be used to project transform coefficients to the low-dimensional vector, and this projection can maintain the information for signal reconstruction. The compressed sensing technology can achieve the reconstruction with high accuracy or high probability using small number of projection data. In the case of extremely low dimension, the reconstruction information is insufficient, and the original signal cannot be reconstructed accurately. So the recognition rate is not high. When the dimension is increased and the reconstructed information can more accurately reconstruct the original signal, the

recognition rate will improve. When the feature dimension is higher ( $\geq 512$ ), PCA cannot work because of the sample size. At 512 dimension, the measurement matrix (Deterministic sparse Toeplitz matrices ( $\Delta = 8$ ) are 95.21%; Deterministic sparse Toeplitz matrices ( $\Delta = 16$ ) are 94.91%) has reached the performance of the original dimension (10240 dimension), and the dimension is only 1/20 of the original dimension. At the 1/10 of the original dimension, the performance of the measurement matrix has basically achieved the performance of the original dimension: Random Gaussian matrices are 95.60%, Deterministic sparse Toeplitz matrices ( $\Delta = 4$ ) are 94.64%, Deterministic sparse Toeplitz matrices ( $\Delta = 8$ ) are 94.56%, and Deterministic sparse Toeplitz matrices ( $\Delta = 16$ ) are 94.98%. In Figure 6(b), 20 samples of each individual were used as training samples. When the feature dimension  $\leq 512$ , the best performance is PCA which is 97.50% (512 dimension). At 1024 dimension, PCA cannot work and the performance of the measurement matrix has basically achieved the performance of the original dimension: Random Gaussian matrices are 98.80%, Deterministic sparse Toeplitz matrices ( $\Delta = 4$ ) are 98.68%, Deterministic sparse Toeplitz matrices ( $\Delta = 8$ ) are 99.01%, and Deterministic sparse Toeplitz matrices ( $\Delta = 16$ ) are 98.77%. In Table 2, we fixed the dimension of each reduction methods at 512 dimension in the case of 20 training samples. In general, the complexity of PCA is  $O(n^3)$  [47], where  $n$  is the number of rows of the covariance matrix. The example we provided consists of 38 individuals, and each individual contains 20 samples. Since the number of Gabor features (10240) for each patch far outweighs the total number of training samples (760), the number of rows of the covariance matrix is determined by the total number of training samples, and the complexity of PCA is approximately  $O(760^3)$ . However, in the proposed measurement matrix algorithm, the measurement matrix is formed by a certain deterministic distribution (e.g., Gaussian distribution) and structure, according to the required dimension and length of the signal, so that the measurement matrix demonstrates low complexity [36, 48, 49]. The actual speed of each dimension reduction methods for one patch of all samples (including all training samples and all testing samples) is listed. Based on the above analysis, we can see that the PCA method has the most outstanding performance, but it is limited by the sample size and it is time-consuming. The measurement matrix does not perform well in low dimension, but is not limited by the sample size. When the dimension reaches a certain value, its performance was the same as that of the original dimension. Thus, the dimension of the data can be reduced without loss of recognition rate.

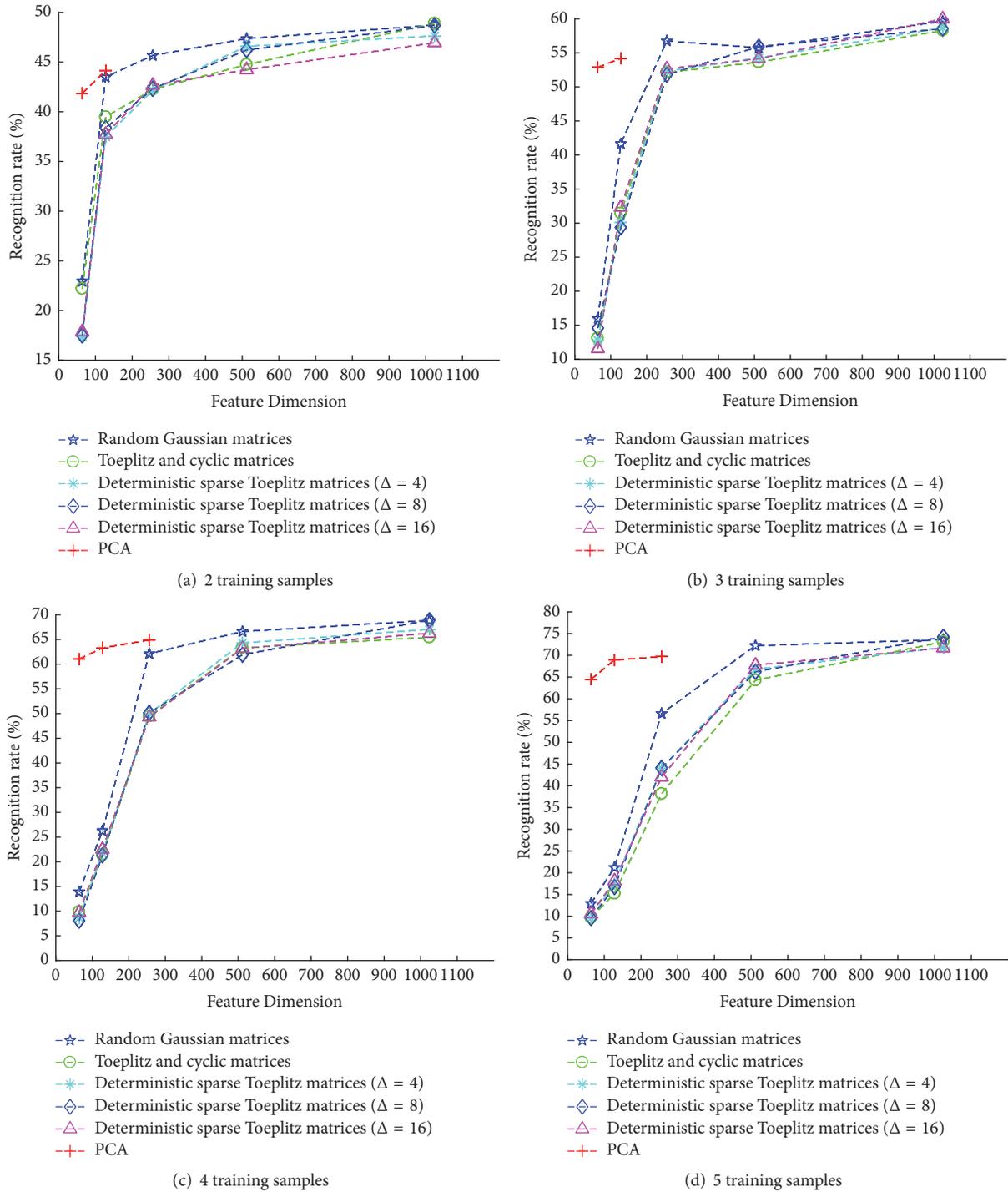


FIGURE 7: The performance of each dimension reduction method for GPCRC on CMU\_PIE database.

4.2. *CMU\_PIE*. In order to further test the robustness in illumination, pose, and expression, we utilized a currently popular database CMU\_PIE [43, 44], a database consisting of 41,368 images of 68 people; for each person, there are 13 different poses, 43 different illumination conditions, and 4 different expressions. To testify the advantage of our method in small sample size condition, we randomly selected 2, 3, 4,

and 5 face samples from each individual as training sample and other 40 face samples from each individual as test samples. The face images size is downsampled to  $32 \times 32$ . Figure 6 shows some marked samples of the CMU\_PIE database. The experimental results of each method are shown in Table 3; GPCRC has the best results. In Figure 7 we compared the recognition rates of various dimension reduction methods

TABLE 3: Recognition rate (%) on CMU\_PIE database.

Features	Method	Training sample size			
		2	3	4	5
Intensity feature	SRC	28.21 ± 5.43	39.36 ± 9.82	44.73 ± 15.29	60.54 ± 6.45
	CRC	30.34 ± 6.15	40.62 ± 9.20	44.35 ± 7.42	61.03 ± 7.82
	SSRC	37.52 ± 8.34	54.74 ± 8.50	57.32 ± 9.82	66.73 ± 8.85
	PCRC	42.32 ± 1.17	56.46 ± 1.88	61.16 ± 2.91	70.47 ± 3.87
LBP feature	SRC	38.42 ± 6.46	53.37 ± 7.10	53.67 ± 5.21	63.31 ± 7.24
	CRC	37.32 ± 7.75	54.83 ± 6.15	54.27 ± 8.31	64.25 ± 6.90
	SSRC	43.17 ± 5.71	56.84 ± 9.21	64.53 ± 4.76	68.25 ± 4.35
	PCRC	41.21 ± 7.10	52.21 ± 11.76	62.54 ± 6.21	70.22 ± 3.07
Gabor feature	SRC	41.43 ± 4.20	53.36 ± 5.40	53.58 ± 7.32	66.79 ± 5.45
	CRC	40.25 ± 5.15	55.35 ± 3.28	56.68 ± 7.42	66.24 ± 6.18
	SSRC	44.55 ± 6.00	55.96 ± 6.20	62.50 ± 3.16	70.41 ± 4.79
	PCRC	<b>48.57 ± 1.29</b>	<b>58.45 ± 1.94</b>	<b>68.82 ± 2.56</b>	<b>73.18 ± 2.44</b>

for GPCRC. Similarly, we can see that PCA cannot achieve the best recognition rate because of the sample size limit, and at 1024 dimension the performance of the measurement matrix has basically achieved the performance of the original dimension. The recognition rate of each measurement matrix is shown below:

- (i) Random Gaussian matrices are **48.73%** (2 training samples), **59.66%** (3 training samples), **68.86%** (4 training samples), and **73.60%** (5 training samples).
- (ii) Toeplitz and cyclic matrices are **48.89%** (2 training samples), 58.30% (3 training samples), 65.44% (4 training samples), and 73.12% (5 training samples).
- (iii) Deterministic sparse Toeplitz matrices ( $\Delta = 4$ ) are 47.63% (2 training samples), **58.68%** (3 training samples), 67.03% (4 training samples), and 71.78% (5 training samples).
- (iv) Deterministic sparse Toeplitz matrices ( $\Delta = 8$ ) are **48.69%** (2 training samples), **58.55%** (3 training samples), **68.91%** (4 training samples), and **74.27%** (5 training samples).
- (v) Deterministic sparse Toeplitz matrices ( $\Delta = 16$ ) are 46.94% (2 training samples), **60.00%** (3 training samples), 66.25% (4 training samples), and 71.69% (5 training samples).

Through the above analysis, we further validated the feasibility of the measurement matrix in the premise of ensuring the recognition rate.

**4.3. LFW.** In practice, the number of training samples is very limited, and only one or two can be obtained from individual identity documents. To simulate the actual situation, we select the LFW database. The LFW database includes frontal images of 5,749 different subjects in unconstrained environment [45]. LFW-a is a version of LFW after alignment using commercial face alignment software [46]. We chose 158 subjects from LFW-a; each subject contains no less than ten samples. For each subject, we randomly choose 1 to 2 samples from each individual for training and another 5 samples from each



FIGURE 8: Some marked samples of the LFW database.

individual for testing. The face images size is downsampled to  $32 \times 32$ . Figure 8 shows some marked samples of the LFW database. The experimental results of each method are shown in Table 4; it can be clearly seen that the GPCRC achieves the highest recognition rate performance in all experiments with the training sample size from 1 to 2. In Figure 9, we can also see the advantages of the measurement matrix in small sample size. At 1024 dimension, the performance of the measurement matrix in the single training sample is as follows: Random Gaussian matrices are **26.18%**, Toeplitz and cyclic matrices are 24.68%, Deterministic sparse Toeplitz matrices ( $\Delta = 4$ ) are 24.43%, Deterministic sparse Toeplitz matrices ( $\Delta = 8$ ) are **25.85%**, and Deterministic sparse Toeplitz matrices ( $\Delta = 16$ ) are 25.28%, and the performances of the measurement matrix in the two training samples are as follows: Random Gaussian matrices are **39.68%**, Toeplitz and cyclic matrices are 37.72%, Deterministic sparse Toeplitz matrices ( $\Delta = 4$ ) are 37.56%, Deterministic sparse Toeplitz matrices ( $\Delta = 8$ ) are **39.91%**, and Deterministic sparse Toeplitz matrices ( $\Delta = 16$ ) are 38.25%.

## 5. Conclusion

In order to alleviate the influence of unreliability environment on small sample size in FR, in this paper we applied improved method for Gabor local features to PCRC; we proposed to use the measurement matrices to reduce the dimension of the transformed signal. Several important observations can be summarized as follows. (1) The proposed GPCPC method can effectively deal with the influence of unreliability environment in small sample FR. (2) The measurement

TABLE 4: Recognition rate (%) on LFW database.

Features	Method	Training sample size	
		1	2
Intensity feature	SRC	10.58 ± 7.20	19.02 ± 8.29
	CRC	9.08 ± 8.15	20.39 ± 7.20
	SSRC	17.03 ± 5.87	28.52 ± 7.50
	PCRC	22.29 ± 3.96	33.29 ± 5.28
LBP feature	SRC	15.52 ± 5.54	25.31 ± 7.42
	CRC	16.23 ± 6.21	26.22 ± 6.57
	SSRC	19.16 ± 3.41	32.10 ± 4.26
	PCRC	21.26 ± 4.73	31.59 ± 6.11
Gabor feature	SRC	18.68 ± 4.23	31.35 ± 6.40
	CRC	17.31 ± 8.15	30.83 ± 5.20
	SSRC	21.30 ± 7.21	33.67 ± 4.33
	PCRC	<b>25.82 ± 3.42</b>	<b>39.60 ± 3.54</b>

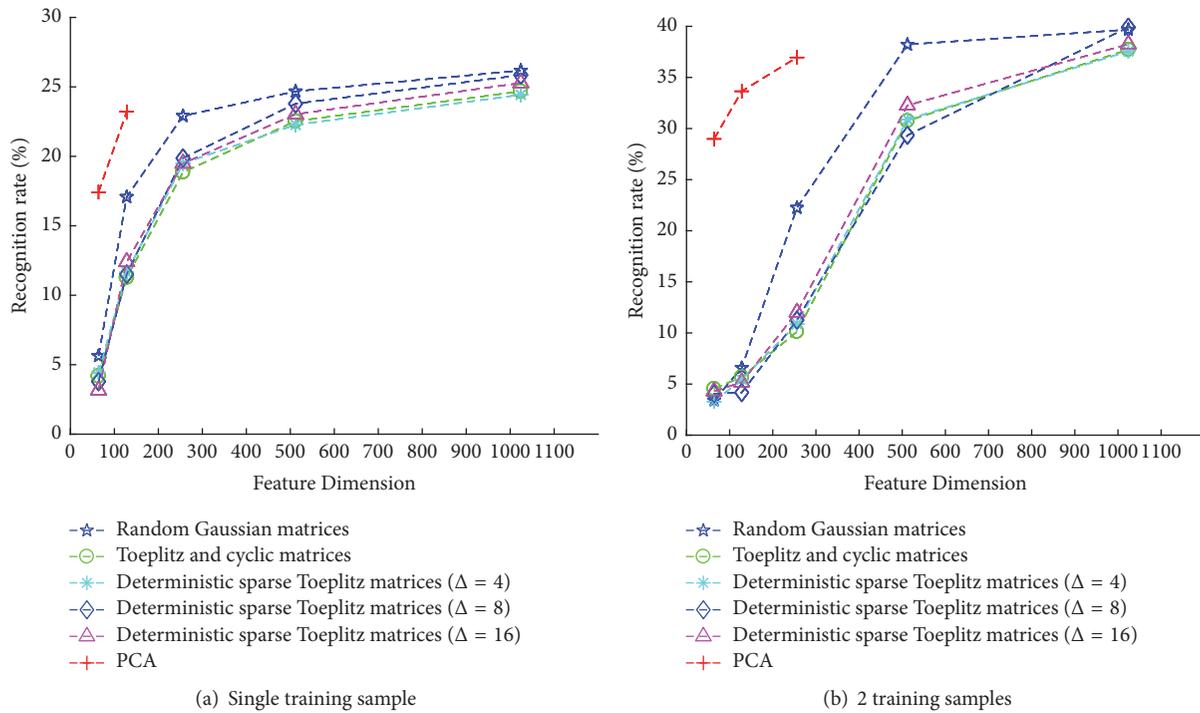


FIGURE 9: The performance of each dimension reduction method for GPCRC on LFW database.

matrix proposed to deal with the high dimension in GPCRC method can effectively improve the computational efficiency and computational speed, and they were able to overcome the limitations of the PCA method.

**Conflicts of Interest**

None of the authors have conflicts of interest.

**Acknowledgments**

This work was supported by the National Natural Science Foundation of China (no. 61672386), Humanities and Social

Sciences Planning Project of Ministry of Education (no. 16YJAZH071), Anhui Provincial Natural Science Foundation of China (no. 1708085MF142), University Natural Science Research Project of Anhui Province (no. KJ2017A259), and Provincial Quality Project of Anhui Province (no. 2016zy131).

**References**

[1] J. Galbally, S. Marcel, and J. Fierrez, “Biometric antispoofing methods: a survey in face recognition,” *IEEE Access*, vol. 2, pp. 1530–1552, 2014.

[2] S. Nagpal, M. Singh, R. Singh, and M. Vatsa, “Regularized deep learning for face recognition with weight variations,” *IEEE Access*, vol. 3, pp. 3010–3018, 2015.

- [3] J. W. Tanaka and D. Simonyi, "The parts and wholes of face recognition: a review of the literature," *The Quarterly Journal of Experimental Psychology*, no. 10, pp. 1–37, 2016.
- [4] W. Yang, Z. Wang, and C. Sun, "A collaborative representation based projections method for feature extraction," *Pattern Recognition*, vol. 48, no. 1, pp. 20–27, 2015.
- [5] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, June 2008.
- [6] K. VenkataNarayana, V. Manoj, and K. K. Swathi, "Enhanced face recognition based on PCA and SVM," *International Journal of Computer Applications*, vol. 117, no. 2, pp. 40–42, 2015.
- [7] E. Owusu and Q. R. Mao, "An svm cadaboost-based face detection system," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 26, no. 4, pp. 477–491, 2014.
- [8] B. V. Dasarathy, "Nearest neighbor (NN) norms: NN pattern classification techniques," *Los Alamitos IEEE Computer Society Press*, vol. 13, no. 100, pp. 21–27, 1991.
- [9] Y. Liu, S. S. Ge, C. Li, and Z. You, " $\kappa$ -NS: a classifier by the distance to the nearest subspace," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 22, no. 8, pp. 1256–1268, 2011.
- [10] J. Hamm and D. D. Lee, "Grassmann discriminant analysis: a unifying view on subspace-based learning," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 376–383, July 2008.
- [11] R. Wang, S. Shan, X. Chen, and W. Gao, "Manifold-manifold distance with application to face recognition based on image set," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, June 2008.
- [12] J. Chien and C. Wu, "Discriminant waveletfaces and nearest feature classifiers for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1644–1649, 2002.
- [13] J. Wright, A. Ganesh, Z. Zhou, A. Wagner, and Y. Ma, "Demo: robust face recognition via sparse representation," in *Proceedings of the 8th IEEE International Conference on Automatic Face & Gesture Recognition (FG '08)*, pp. 1–2, IEEE, Amsterdam, The Netherlands, September 2008.
- [14] W. Deng, J. Hu, and J. Guo, "In defense of sparsity based face recognition," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 399–406, June 2013.
- [15] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 471–478, Barcelona, Spain, November 2011.
- [16] P. Zhu, L. Zhang, Q. Hu, and S. C. K. Shiu, "Multi-scale patch based collaborative representation for face recognition with margin distribution optimization," in *European Conference on Computer Vision*, vol. 7572 of *Lecture Notes in Computer Science*, pp. 822–835, 2012.
- [17] D. Yang, W. Chen, J. Wang, and Y. Xu, "Patch based face recognition via fast collaborative representation based classification and expression insensitive two-stage voting," *International Conference on Computational Science and Its Applications*, vol. 9787, pp. 562–570, 2016.
- [18] J. Lai and X. Jiang, "Classwise sparse and collaborative patch representation for face recognition," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3261–3272, 2016.
- [19] M. Yang and L. Zhang, "Gabor feature based sparse representation for face recognition with gabor occlusion dictionary," in *Proceedings of the 11th European Conference on Computer Vision (ECCV '10)*, pp. 448–461, Springer, Crete, Greece, 2010.
- [20] M. R. D. Rodavia, O. Bernaldez, and M. Ballita, "Web and mobile based facial recognition security system using Eigenfaces algorithm," in *Proceedings of the 2016 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE '16)*, pp. 86–92, Bangkok, Thailand, December 2016.
- [21] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.
- [22] L. P. Yang, W. G. Gong, W. H. Li, and X. H. Gu, "Random sampling subspace locality preserving projection for face recognition," *Optics & Precision Engineering*, vol. 16, no. 8, pp. 1465–1470, 2008.
- [23] X. Sun, M. Chen, and A. Hauptmann, "Action recognition via local descriptors and holistic features," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 58–65, June 2009.
- [24] X. Wang, Y. Zhang, X. Mu, and F. S. Zhang, "The face recognition algorithm based on improved lbp," *International Journal of Optoelectronic Engineering*, vol. 1, no. 4, pp. 76–80, 2012.
- [25] J. Ahmad, U. Ali, and R. J. Qureshi, "Fusion of thermal and visual images for efficient face recognition using gabor filter," in *Proceedings of the IEEE International Conference on Computer Systems and Applications*, pp. 135–139, March 2006.
- [26] V. Štruc, R. Gajšek, and N. Pavešić, "Principal gabor filters for face recognition," in *Proceedings of the IEEE 3rd International Conference on Biometrics: Theory, Applications and Systems (BTAS '09)*, pp. 1–6, Washington, DC, USA, September 2009.
- [27] M. Li, X. Yu, K. H. Ryu, S. Lee, and N. Theera-Umpon, "Face recognition technology development with gabor, PCA and SVM methodology under illumination normalization condition," *Cluster Computing*, no. 3, pp. 1–10, 2017.
- [28] J. Yang, D. Zhang, and A. F. Frangi, "Two-dimensional PCA: a new approach to appearance -based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131–137, 2004.
- [29] H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional data: with application to face recognition," *Pattern Recognition*, vol. 34, no. 10, pp. 2067–2070, 2001.
- [30] J. Kim, J. Choi, J. Yi, and M. Turk, "Effective representation using ICA for face recognition robust to local distortion and partial occlusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1977–1981, 2005.
- [31] C. G. Turhan and H. S. Bilge, "Class-wise two-dimensional PCA method for face recognition," *IET Computer Vision*, vol. 11, no. 4, pp. 286–300, 2017.
- [32] A. Mashhoori and M. Zolghadri Jahromi, "Block-wise two-directional 2DPCA with ensemble learning for face recognition," *Neurocomputing*, vol. 108, pp. 111–117, 2013.
- [33] M. P. Rajath Kumar, R. Keerthi Sravan, and K. M. Aishwarya, "Artificial neural networks for face recognition using PCA and BPNN," in *Proceedings of the 35th IEEE Region 10 Conference (TENCON '15)*, pp. 1–6, IEEE, Macao, China, November 2015.
- [34] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: universal encoding strategies?" *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

- [35] W. U. Bajwa, J. D. Haupt, G. M. Raz, S. J. Wright, and R. D. Nowak, "Toeplitz-structured compressed sensing matrices," in *Proceedings of the IEEE/SP 14th Workshop on Statistical Signal Processing (SSP '07)*, pp. 294–298, IEEE, Madison, WI, USA, August 2007.
- [36] C. Zhang, H.-R. Yang, and S. Wei, "Compressive sensing based on deterministic sparse toeplitz measurement matrices with random pitch," *Acta Automatica Sinica*, vol. 38, no. 8, pp. 1362–1369, 2012.
- [37] E. S. Takemura, M. R. Petraglia, and A. Petraglia, "An image super-resolution algorithm based on wiener filtering and wavelet transform," in *Proceedings of the 2013 IEEE Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE '13)*, pp. 130–134, August 2013.
- [38] M. Yang, *Face Recognition via Sparse Representation*, Wiley Encyclopedia of Electrical and Electronics Engineering, 2008.
- [39] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.
- [40] A. S. Georghiadis, P. N. Belhumeur, and D. J. Kriegman, "From few to many: illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.
- [41] N. McLaughlin, J. Ming, and D. Crookes, "Largest matching areas for illumination and occlusion robust face recognition," *IEEE Transactions on Cybernetics*, vol. 47, no. 3, pp. 796–808, 2017.
- [42] Y. Xu, X. Fang, J. Wu, X. Li, and D. Zhang, "Discriminative transfer subspace learning via low-rank and sparse representation," *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 850–863, 2016.
- [43] H. Roy and D. Bhattacharjee, "Local-gravity-face (lg-face) for illumination-invariant and heterogeneous face recognition," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 7, pp. 1412–1424, 2016.
- [44] Y. Tai, J. Yang, Y. Zhang, L. Luo, J. Qian, and Y. Chen, "Face recognition with pose variations and misalignment via orthogonal procrustes regression," *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, vol. 25, no. 6, pp. 2673–2683, 2016.
- [45] G. B. Huang, M. Narayana, and E. Learned-Miller, "Labeled faces in the wild: a database for studying face recognition in unconstrained environments," *Month*, 2008.
- [46] L. Wolf, T. Hassner, and Y. Taigman, "Similarity scores based on background samples," in *Asian Conference on Computer Vision*, vol. 5995, pp. 88–97, 2009.
- [47] J. Wang, *Geometric Structure of High-Dimensional Data and Dimensionality Reduction*, Springer, 2012.
- [48] Q. Wang, J. Li, and Y. Shen, "A survey on deterministic measurement matrix construction algorithms in compressive sensing," *Acta Electronica Sinica*, vol. 41, no. 10, pp. 2041–2050, 2013.
- [49] W. Yin, S. Morgan, J. Yang, and Y. Zhang, "Practical compressive sensing with toeplitz and circulant matrices," in *Proceedings of the Visual Communications and Image Processing 2010*, vol. 7744, July 2010.

## Research Article

# Global Optimization for Generalized Linear Multiplicative Programming Using Convex Relaxation

Yingfeng Zhao  and Ting Zhao

School of Mathematical Science, Henan Institute of Science and Technology, Xinxiang 453003, China

Correspondence should be addressed to Yingfeng Zhao; zhaoyingfeng6886@163.com

Received 10 February 2018; Accepted 1 April 2018; Published 10 May 2018

Academic Editor: Guillermo Cabrera-Guerrero

Copyright © 2018 Yingfeng Zhao and Ting Zhao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Applications of generalized linear multiplicative programming problems (LMP) can be frequently found in various areas of engineering practice and management science. In this paper, we present a simple global optimization algorithm for solving linear multiplicative programming problem (LMP). The algorithm is developed by a fusion of a new convex relaxation method and the branch and bound scheme with some accelerating techniques. Global convergence and optimality of the algorithm are also presented and extensive computational results are reported on a wide range of problems from recent literature and GLOBALlib. Numerical experiments show that the proposed algorithm with a new convex relaxation method is more efficient than usual branch and bound algorithm that used linear relaxation for solving the LMP.

## 1. Introduction

This paper deals with finding global optimal solutions for generalized linear multiplicative programming problems of the form

(LMP):

$$\left\{ \begin{array}{l} \min \quad f_0(x) = \sum_{j=1}^{p_0} \phi_{0j}(x) \psi_{0j}(x) \\ \text{s.t.} \quad f_i(x) = \sum_{j=1}^{p_i} \phi_{ij}(x) \psi_{ij}(x) \leq 0, \quad i = 1, 2, \dots, M, \\ \quad \quad f_i(x) = \sum_{j=1}^{p_i} \phi_{ij}(x) \psi_{ij}(x) \geq 0, \quad i = M + 1, M + 2, \dots, N, \\ \quad \quad x \in D = \{x \in R^n \mid Ax \leq b, x \geq 0\}, \end{array} \right. \quad (1)$$

where  $\phi_{ij}(x), \psi_{ij}(x)$  in objective and constraints are linear functions with general forms  $\phi_{ij}(x) = \sum_{k=1}^n a_{ijk}x_k + b_{ij}$ ,  $\psi_{ij}(x) = \sum_{k=1}^n c_{ijk}x_k + d_{ij}$ ,  $a_{ijk}, c_{ijk}, b_{ij}$ , and  $d_{ij}$  are all arbitrary real numbers,  $i = 0, 1, 2, \dots, N$ ,  $j = 1, 2, \dots, p_i$ ,  $k = 1, 2, \dots, n$ ,  $A \in R^{m \times n}$  is a matrix,  $b \in R^m$  is a vector, and set  $D \subset R^n$  is nonempty and bounded.

Generally, linear multiplicative programming problem (LMP) is a special case of nonconvex programming problem, known to be NP-hard. Linear multiplicative programming

problem (LMP) has attracted considerable attention in the literature because of its large number of practical applications in various fields of study, including financial optimization in Konno et al. [1], VLSI chip design in Dorneich and Sahinidis [2], data mining and pattern recognition in Bennett and Mangasarian [3], plant layout design in Quesada and Grossmann [4], marketing and service planning in Samadi et al. [5], robust optimization in Mulvey et al. [6], multiple-objective decision in Benson [7], in Keeney and Raika [8], in Geoffrion [9], location-allocation problems in Konno et al. [10], constrained bimatrix games in Mangasarian [11], three-dimensional assignment problems in Frieze [12], certain linear max-min problems in Falk [13], and many problems in engineering design, economic management, and operations research. Another reason why this problem attracts so much attention is that, by utilizing some techniques, many mathematical programs like general quadratic programming, bilinear programming, linear multiplicative programming, and quadratically constrained quadratic programming can be converted into the special form of LMP in Benson [14]; in Tuy [15], the important class of sum of ratios fractional problems can also be encapsulated by LMP; in fact, suppose that  $f_i$  are convex and  $g_i$  are concave and positive. Since each  $1/g_i(x)$  is convex and positive, then the sum of ratios fractional problem

(with the objective function as  $\sum(f_i(x)/g_i(x))$ ) reduces to LMP when  $g_i$  and  $f_i$  are linear. When  $g_i$  and  $f_i$  are negative, one can add a large real member to make negative be positive because of the compactness of the constraint set in Dai et al. [16].

Moreover, from the algorithmic design point of view, sum of products of two affine functions need not be convex (even not be quasi-convex), and hence linear multiplicative programming problem (LMP) may have multiple local solutions, which fail to be global optimal. Due to the facts above, developing practical global optimization algorithms for the LMP has great theoretical and the algorithmic significance.

In the past 20 years, many solution methods have been proposed for globally solving linear multiplicative programming problem (LMP) and its special cases. The methods are mainly classified as parameterization-based methods, outer-approximation methods, kinds of branch-and-bound methods, decomposition method, and cutting plane methods in Konno et al. [17], in Thoai [18], in Shen and Jiao [19, 20], in Wang et al. [21], in Jiao et al. [22], and in Shen et al. [23]. Most of these methods for linear multiplicative programming are designed only for problems in which the constraint functions are all linear or under the assumption that all linear functions that appeared in the multiplicative terms are nonnegative. And most branch-and-bound algorithms for this problem are developed based on two-phase linear relaxation scheme, which will take more computational time in the approximation process. For example, in Zhao and Liu [24] and in Jiao et al. [22], both algorithms utilize two-phase relaxation methods, and a large amount of computational time is consumed in the relaxation process. Compared with these algorithms, the main features of our new algorithm are threefold. (1) The problem investigated in this paper is linear multiplicative constrained multiplicative programming; it has a more general form than those linear multiplicative programming problems with linear constraints. (2) The relaxation programming problem is a convex programming which can be obtained with one-step relaxation; this will greatly improve the efficiency of approximation process. (3) The condition  $\phi_{ij}(x) > 0$ ,  $\psi_{ij}(x) > 0$ ,  $\forall x \in X$ , is a nonnecessitating requirement in our algorithm. (4) Extensive computational numerical examples and comparison from recent literature and GLOBALlib are performed to test our algorithm for LMP.

The rest of this paper is arranged in the following way. In Section 2, the construction process of the convex relaxation problem is detailed, which will provide a reliable lower bound for the optimal value of LMP. In Section 3, some key operations for designing a branch-and-bound algorithm and the global optimization algorithm for LMP are described. Convergence property of the algorithm is also established in this section. Numerical results are reported to show the feasibility and efficiency of our algorithm in Section 4 and some concluding remarks are reported in the last section.

## 2. Convex Relaxation of LMP

As is known to all, constructing a well-performed relaxation problem can bring great convenience for designing

branch-and-bound algorithm of global optimization problems. In this section, we will show how to construct the convex relaxation programming problem (CRP) for LMP. For this, we first compute the initial variable bounds by solving the following linear programming problems:

$$\begin{aligned} x_i^l &= \min_{x \in D} x_i, \\ x_i^u &= \max_{x \in D} x_i, \\ i &= 1, 2, \dots, n; \end{aligned} \quad (2)$$

then an initial hyperrectangle  $X^0 = \{x \in R^n \mid x_i^l \leq x_i \leq x_i^u, i = 1, 2, \dots, n\}$  can be obtained.

For convenience for introduction of relaxation convex programming problem for LMP over subrectangle  $X \subset X^0$ , we further solve the following set of linear programming problems:

$$\begin{aligned} l_{ij} &= \min_{x \in D \cap X} \phi_{ij}(x), \\ u_{ij} &= \max_{x \in D \cap X} \phi_{ij}(x), \\ L_{ij} &= \min_{x \in D \cap X} \psi_{ij}(x), \\ U_{ij} &= \max_{x \in D \cap X} \psi_{ij}(x). \end{aligned} \quad (3)$$

Based on the construction process of  $l_{ij}$  and  $L_{ij}$  and  $u_{ij}$  and  $U_{ij}$ , it is not hard to see that

$$(\phi_{ij}(x) - u_{ij})(\psi_{ij}(x) - U_{ij}) \geq 0, \quad (4)$$

$$(\phi_{ij}(x) - l_{ij})(\psi_{ij}(x) - L_{ij}) \geq 0,$$

$$(\phi_{ij}(x) - l_{ij})(\psi_{ij}(x) - U_{ij}) \leq 0, \quad (5)$$

$$(\phi_{ij}(x) - u_{ij})(\psi_{ij}(x) - L_{ij}) \leq 0;$$

by combining (4) with (5) and performing some equivalent transformation, we can easily obtain a lower and an upper bound of each bilinear term; that is,

$$\begin{aligned} \phi_{ij}(x) \psi_{ij}(x) &\leq \min \{u_{ij}\psi_{ij}(x) + L_{ij}\phi_{ij}(x) \\ &\quad - u_{ij}L_{ij}, l_{ij}\psi_{ij}(x) + U_{ij}\phi_{ij}(x) - U_{ij}l_{ij}\}, \end{aligned} \quad (6)$$

$$\begin{aligned} \phi_{ij}(x) \psi_{ij}(x) &\geq \max \{u_{ij}\psi_{ij}(x) + U_{ij}\phi_{ij}(x) \\ &\quad - U_{ij}u_{ij}, l_{ij}\psi_{ij}(x) + L_{ij}\phi_{ij}(x) - L_{ij}l_{ij}\}. \end{aligned}$$

To facilitate the narrative,  $\forall i = 0, 1, 2, \dots, N$ , by denoting

$$\begin{aligned} \underline{g}_{ij}^1(x) &\triangleq u_{ij}\psi_{ij}(x) + U_{ij}\phi_{ij}(x) - U_{ij}u_{ij}, \\ \underline{g}_{ij}^2(x) &\triangleq l_{ij}\psi_{ij}(x) + L_{ij}\phi_{ij}(x) - L_{ij}l_{ij}, \\ \bar{g}_{ij}^1(x) &\triangleq u_{ij}\psi_{ij}(x) + L_{ij}\phi_{ij}(x) - u_{ij}L_{ij}, \\ \bar{g}_{ij}^2(x) &\triangleq l_{ij}\psi_{ij}(x) + U_{ij}\phi_{ij}(x) - U_{ij}l_{ij}, \end{aligned} \quad (7)$$

we can reformulate conclusion (6) as

$$\begin{aligned} \phi_{ij}(x) \psi_{ij}(x) &\leq \min \{ \bar{g}_{ij}^1(x), \bar{g}_{ij}^2(x) \} \triangleq \bar{g}_{ij}(x), \\ \phi_{ij}(x) \psi_{ij}(x) &\geq \max \{ \underline{g}_{ij}^1(x), \underline{g}_{ij}^2(x) \} \triangleq \underline{g}_{ij}(x), \end{aligned} \quad (8)$$

respectively. With this, we obtain a lower bound function  $\underline{g}_i(x)$  and upper bound function  $\bar{g}_i(x)$  for  $f_i(x)$ , which satisfy  $\underline{g}_i(x) \leq f_i(x) \leq \bar{g}_i(x)$ ,  $i = 0, 1, 2, \dots, N$ , where

$$\begin{aligned} \underline{g}_i(x) &= \sum_{j=1}^{p_i} \underline{g}_{ij}(x), \\ \bar{g}_i(x) &= \sum_{j=1}^{p_i} \bar{g}_{ij}(x), \end{aligned} \quad (9)$$

$$i = 0, 1, 2, \dots, N.$$

So far, based on the above discussion, it is not too hard to obtain the relaxation programming problem over  $X$  for the LMP which we formulated as follows:

(CRP):

$$\left\{ \begin{array}{l} \min \quad \underline{g}_0(x) = \sum_{j=1}^{p_0} \underline{g}_{0j}(x) \\ \text{s.t.} \quad \underline{g}_i(x) = \sum_{j=1}^{p_i} \underline{g}_{ij}(x) \leq 0, \quad i = 1, 2, \dots, M, \\ \quad \quad \bar{g}_i(x) = \sum_{j=1}^{p_i} \bar{g}_{ij}(x) \geq 0, \quad i = M+1, M+2, \dots, N, \\ \quad \quad x \in D \cap X = \{x \in X \mid Ax \leq b, x \geq 0\}. \end{array} \right. \quad (10)$$

*Remark 1.* As one can easily confirm, program (CRP) is a convex program which can be effectively solved by some convex optimization tool-boxes such as CVX; that is, all functions that appeared in the objective and constraints of CRP are convex.

*Remark 2.* Both the lower and upper bound functions  $\underline{g}_i(x)$  and  $\bar{g}_i(x)$  will approximate function  $f_i(x)$ , as the diameter of rectangle converges to zero.

### 3. Branch-and-Bound Algorithm and Its Convergence

As we have known, branch-and-bound algorithms utilize tree search strategies to implicitly enumerate all possible solutions to a given problem, applying pruning techniques to eliminate regions of the search space that cannot yield a better solution. There are three algorithmic components in branch-and-bound scheme which can be specified to fine-tune the performance of the algorithm. These components are the search strategy, the branching strategy, and the pruning rules. This section presents a description of these components and the proposed algorithm for solving LMP.

*3.1. Key Operation.* There are two important phases of any branch-and-bound algorithm: search phase and verification phase.

The choice of search strategy primarily impacts the search phase and has potentially significant consequences for the amount of computational time required and memory used. In this paper, we will choose the depth-first search (DFS) strategy with node ranking techniques, which can reduce a lot of storage space.

The choice of branching strategy determines how children nodes are generated from a subproblem. It has significant impacts on both the search phase and the verification phase. By branching appropriately at subproblems, the strategy can guide the algorithm towards optimal solutions. In this paper, to develop the proposed algorithm for LMP, we adopt a standard range bisection approach, which is adequate to insure global convergence of the proposed algorithm. Detailed process is described as follows.

For any region  $X = [x^l, x^u] \subset X^0$ , let  $r \in \operatorname{argmax}\{x_i^u - x_i^l \mid i = 1, 2, \dots, n\}$  and  $x_{\text{mid}} = (x_r^l + x_r^u)/2$ , and then the current region  $X$  can be divided into the two following subregions:

$$\begin{aligned} \bar{X} &= \{x \in R^n \mid x_i^l \leq x_i \leq x_i^u, \quad i \neq r, \quad x_r^l \leq x_r \leq x_{\text{mid}}\}, \\ \underline{X} &= \{x \in R^n \mid x_i^l \leq x_i \leq x_i^u, \quad i \neq r, \quad x_{\text{mid}} \leq x_r \leq x_r^u\}. \end{aligned} \quad (11)$$

Another critical aspect of branch-and-bound algorithm is the choice of pruning rules used to exclude regions of the search space from exploration. The most common way to prune is to produce a lower and (or) upper bound on the objective function value at each subproblem and use this to prune subproblems whose bound is no better than the incumbents solution value. For each partition subset  $X$  generated by the above branching operation, the bounding operation is mainly concentrated on estimating a lower bound  $\text{LB}(X)$  and an upper bound  $\text{UB}(X)$  for the optimal value of linear multiplicative programming problem (LMP).  $\text{LB}(X)$  can be obtained by solving the following convex relaxation programming problems:

(CRP(X)):

$$\left\{ \begin{array}{l} \min \quad \underline{g}_0(x) = \sum_{j=1}^{p_0} \underline{g}_{0j}(x) \\ \text{s.t.} \quad \underline{g}_i(x) = \sum_{j=1}^{p_i} \underline{g}_{ij}(x) \leq 0, \quad i = 1, 2, \dots, M, \\ \quad \quad \bar{g}_i(x) = \sum_{j=1}^{p_i} \bar{g}_{ij}(x) \geq 0, \quad i = M+1, M+2, \dots, N, \\ \quad \quad x \in D \cap X = \{x \in X \mid Ax \leq b, x \geq 0\}. \end{array} \right. \quad (12)$$

Moreover, since any feasible solution of linear multiplicative programming problem (LMP) will provide a valid upper bound of the optimal value, we can evaluate the initial objective value of the optimal solution of CRP to determine an upper bound (if possible) for the optimal value of linear multiplicative programming problem (LMP) over  $X$ .

*3.2. Branch-and-Bound Algorithm.* Based on the former discussion, the presented algorithm for globally solving the LMP can be summarized as follows.

*Step 0 (initialization).*

*Step 0.1.* Set iteration counter  $k := 0$  and the initial partition set as  $\Omega_0 = X^0$ . The set of active nodes  $\bar{X}^0 = \{X^0\}$ ; the upper bound  $UB = \infty$ ; the set of feasible points  $F = \emptyset$ ; the convergence tolerance  $\epsilon = 1 \times 10^{-8}$ .

*Step 0.2.* Solve the initial convex relaxation problem (CRP) over region  $X^0$ ; if the CRP is not feasible, then there is no feasible solution for the initial problem. Otherwise, denote the optimal value and solution as  $f_{\text{opt}}^0$  and  $x_{\text{opt}}^0$ , respectively. If  $x_{\text{opt}}^0$  is feasible to the LMP, we can obtain an initial upper bound and lower bound of the optimal value for linear multiplicative programming problem (LMP); that is,  $UB := f_0(x_{\text{opt}}^0)$ , and  $LB := f_{\text{opt}}^0$ . And then, if  $UB - LB < \epsilon$ , the algorithm can stop, and  $x_{\text{opt}}^0$  is the optimal solution of the LMP; otherwise proceed to Step 1.

*Step 1 (branching).* Partition  $X^k$  into two new subrectangles according to the partition rule described in Section 3.1. Delete  $X^k$  and add the new nodes into the active nodes set  $\bar{X}^k$ ; still denote the set of new partitioned sets as  $\bar{X}^k$ .

*Step 2 (bounding).* For each subregion still of interest  $X^{k\mu} \subseteq X^0$ ,  $\mu = 1, 2$ , obtain the optimal solution and value for convex relaxation problem (CRP) by solving the convex relaxation programming problem over  $X^{k\mu}$ ; if  $LB(X^{k\mu}) > UB$ , delete  $X^{k\mu}$  from  $\bar{X}^k$ . Otherwise, we can update the lower and upper bounds:  $LB = \min\{LB(X^{k\mu}) \mid \mu = 1, 2\}$  and  $UB = \min\{UB, f(x^{k\mu}) \mid \mu = 1, 2\}$ .

*Step 3 (termination).* If  $UB - LB \leq \epsilon$ , the algorithm can be stopped;  $UB$  is the global optimal value for LMP. Otherwise, set  $k := k + 1$ , and select the node with the smallest optimal value as the current active node, and return to Step 1.

**3.3. Global Convergence of the Algorithm.** The global convergence properties of the above algorithm for solving linear multiplicative programming problem (LMP) can be given in the sense of the following theorem.

**Theorem 3.** *The proposed algorithm will terminate within finite iterations or it will generate an infinite sequence  $\{x^k\}$ , any accumulation point of which is a global optimal solution for the LMP.*

*Proof.* If the proposed algorithm terminates finitely, assume that it terminates at the  $k_{\text{th}}$  iteration:  $k \geq 0$ . By the termination criteria, we know that

$$UB - LB \leq \epsilon. \quad (13)$$

Based on the upper bounding technique described in Step 2, it is implied that

$$f(x^k) - LB \leq \epsilon. \quad (14)$$

Let  $v_{\text{opt}}$  be the optimal value of linear multiplicative programming problem (LMP); then, by Section 3.1 and Step 2 above, we know that

$$UB = f(x^k) \geq v_{\text{opt}} \geq LB. \quad (15)$$

Hence, taking (14) and (15) together, it is implied that

$$v_{\text{opt}} + \epsilon \geq LB + \epsilon \geq f(x^k) \geq v_{\text{opt}}, \quad (16)$$

and thus the proof of the first part is completed.

If the algorithm is infinite, then it generates an infinite feasible solution sequence  $\{x^k\}$  for the LMP via solving the CRP. Since the feasible region of LMP is compact, the sequence  $\{x^k\}$  must have a convergent subsequence. For definiteness and without loss of generality, assume  $\lim_{k \rightarrow \infty} x^k = x^*$ , and then we have

$$\lim_{k \rightarrow \infty} \phi_{ij}(x^k) = \phi_{ij}(x^*), \quad (17)$$

$$\lim_{k \rightarrow \infty} \psi_{ij}(x^k) = \psi_{ij}(x^*).$$

By the definitions of  $l_{ij}$  and  $L_{ij}$  and  $u_{ij}$  and  $U_{ij}$ , we know that

$$\lim_{k \rightarrow \infty} l_{ij}(x^k) = \phi_{ij}(x^*), \quad (18)$$

$$\lim_{k \rightarrow \infty} L_{ij}(x^k) = \psi_{ij}(x^*).$$

Moreover, we have

$$UB_k = \lim_{k \rightarrow \infty} f(x^k) = f(x^*), \quad (19)$$

$$LB_k = \lim_{k \rightarrow \infty} \sum_{j=1}^{p_0} g_{0j}(x) = f(x^*).$$

Therefore, we have  $\lim_{k \rightarrow \infty} (UB_k - LB_k) = 0$ . This implies that  $x^*$  is a global optimal solution for linear multiplicative programming problem (LMP).  $\square$

## 4. Numerical Experiments

To test the proposed algorithm in efficiency and solution quality, we performed some computational examples on a personal computer containing an Intel Core i5 processor of 2.40 GHz and 4 GB of RAM. The code base is written in Matlab 2014a and all subproblems are solved by using CVX.

We consider some instances of linear multiplicative programming problem (LMP) from some recent literature in Wang and Liang [25], in Jiao [26], in Shen et al. [23], in Shen and Jiao [19, 20], and in Jiao et al. [22] and GLOBALlib at <http://www.gamsworld.org/global/globallib.htm>. Among them, Examples 1–6 are taken from some recent literature. Examples 7–11 are taken from GLOBALlib, a collection of nonlinear programming models. The last example is a nonlinear nonconvex mathematical programming problem with randomized linear multiplicative objective and constraint functions.

*Example 1* (references in Shen et al. [23]).

$$\begin{aligned}
 \min \quad & -4x_1^2 - 5x_2^2 + x_1x_2 + 2x_1 \\
 \text{s.t.} \quad & x_1 - x_2 \geq 0, \\
 & \frac{1}{3}x_1^2 - \frac{1}{3}x_2^2 \leq 1, \\
 & \frac{1}{2}x_1x_2 \leq 1, \\
 & 0 \leq x_1 \leq 3, \\
 & x_2 \geq 0.
 \end{aligned} \tag{20}$$

*Example 2* (references in Wang and Liang [25] and in Jiao [26]).

$$\begin{aligned}
 \min \quad & x_1^2 + x_2^2 \\
 \text{s.t.} \quad & 0.3x_1x_2 \geq 1, \\
 & 2 \leq x_1 \leq 5, \\
 & 1 \leq x_2 \leq 3.
 \end{aligned} \tag{21}$$

*Example 3* (references in Jiao [26]).

$$\begin{aligned}
 \min \quad & x_1^2 + x_2^2 - x_3^2 \\
 \text{s.t.} \quad & 0.3x_1x_2 + 0.3x_2x_3 + 0.6x_1x_3 \geq 4, \\
 & 2 \leq x_1 \leq 5, \\
 & 1 \leq x_2 \leq 3, \\
 & 1 \leq x_3 \leq 3.
 \end{aligned} \tag{22}$$

*Example 4* (references in Shen and Jiao [19, 20] and in Jiao et al. [22]).

$$\begin{aligned}
 \min \quad & x_1x_2 - 2x_1 + x_2 + 1 \\
 \text{s.t.} \quad & 8x_2^2 - 6x_1 - 16x_2 \leq -11, \\
 & -x_2^2 + 3x_1 + 2x_2 \leq 7, \\
 & 1 \leq x_1 \leq 2.5, \\
 & 1 \leq x_2 \leq 2.225.
 \end{aligned} \tag{23}$$

*Example 5* (references in Shen and Jiao [19, 20] and in Jiao et al. [22]).

$$\begin{aligned}
 \min \quad & x_1 \\
 \text{s.t.} \quad & \frac{1}{4}x_1 + \frac{1}{2}x_2 - \frac{1}{16}x_1^2 - \frac{1}{16}x_2^2 \leq 1, \\
 & \frac{1}{14}x_1^2 + \frac{1}{14}x_2^2 - \frac{3}{7}x_1 - \frac{3}{7}x_2 \leq -1, \\
 & 1 \leq x_1 \leq 5.5, \\
 & 1 \leq x_2 \leq 5.5.
 \end{aligned} \tag{24}$$

*Example 6* (references in Jiao et al. [22]).

$$\begin{aligned}
 \min \quad & x_1 + (2x_1 - 3x_2 + 13)(x_1 + x_2 - 1) \\
 \text{s.t.} \quad & -x_1 + 2x_2 \leq 8, \\
 & -x_2 \leq -3, \\
 & x_1 + 2x_2 \leq 12, \\
 & x_1 - 2x_2 \leq -5, \\
 & x_1 \geq 0, \\
 & x_2 \geq 0.
 \end{aligned} \tag{25}$$

Examples 1–6 are taken from some literature, where they are solved by branch-and-bound algorithm with linear relaxation techniques. Numerical experiment demonstrates that the above method is more efficient than other methods in the sense that our algorithm requires rather less iterations and CPU time for solving the same problems. Specific results of numerical Examples 1–6 are listed in Table 1, where the notations used in the headline have the following meanings: Exam.: serial number of numerical examples in this paper; Ref.: serial number of numerical examples in the references; Iter.: iterative times; Time: CPU time in seconds; Prec.: precision used in the algorithm; Opt. val. and Opt. sol. denote the optimal value and solution of the problem, respectively.

*Example 7* (st-qpkl).

$$\begin{aligned}
 \min \quad & 2x_1 - 2x_1^2 + 2x_1x_2 + 3x_2 - 2x_2^2 \\
 \text{s.t.} \quad & -x_1 + x_2 \leq 1, \\
 & x_1 - x_2 \leq 1, \\
 & -x_1 + 2x_2 \leq 3, \\
 & 2x_1 - x_2 \leq 3, \\
 & 0 \leq x_1, x_2.
 \end{aligned} \tag{26}$$

*Example 8* (st-z).

$$\begin{aligned}
 \min \quad & -x_1^2 - x_2^2 - x_3^2 + 2x_3 \\
 \text{s.t.} \quad & x_1 + x_2 - x_3 \leq 0, \\
 & -x_1 + x_2 - x_3 \leq 0, \\
 & 12x_1 + 5x_2 + 12x_3 \leq 22.8, \\
 & 12x_1 + 12x_2 + 7x_3 \leq 17.1, \\
 & -6x_1 + x_2 + x_3 \leq 1.9, \\
 & x_2 \geq 0.
 \end{aligned} \tag{27}$$

*Example 9* (ex5-4-2).

$$\begin{aligned}
 \min \quad & x_1 + x_2 + x_3 \\
 \text{s.t.} \quad & x_4 + x_6 \leq 400, \\
 & -x_4 + x_5 + x_7 \leq 300,
 \end{aligned}$$

TABLE 1: Results of the numerical contrast experiments 1–6.

Exam.	Ref.	Opt. val.	Opt. sol.	Iter.	Time (s)	Prec.
1	in Shen et al. [23]	-15.000	(2.0, 1)	1657	120.58	$10^{-6}$
	Ours	-15.0000	(2.0000, 1.0000)	31	0.34449	$10^{-8}$
2	In Wang and Liang [25]	6.7780	(2.00003, 1.66665)	44	0.18	$10^{-4}$
	In Jiao [26]	6.77778	(2.0, 1.666667)	58	<1	$10^{-8}$
	Ours	6.77778	(2.0000, 1.6667)	1	0.0137	$10^{-8}$
3	In Jiao [26]	-4.0	(2.0, 1.0, 3.0)	43	-	$10^{-8}$
	Ours	-4.0	(2.0000, 1.0000, 3.0000)	1	0.0214	$10^{-8}$
4	In Shen and Jiao [19, 20]	0.0000	(2.00, 1.00)	24	-	$10^{-3}$
	In Jiao et al. [22]	0.00000003	(2.0000061, 1.0)	16	0.018	$10^{-8}$
	Ours	0.0000	(2.0000, 1.0000)	1	0.0351	$10^{-8}$
5	In Shen and Jiao [19, 20]	1.1771	(1.17709, 2.1772)	434	1	$10^{-3}$
	In Jiao et al. [22]	1.17708	(1.17709, 2.1772)	189	0.226	$10^{-6}$
	Ours	1.1770	(1.177088, 2.17718)	3	0.1534	$10^{-8}$
6	In Jiao et al. [22]	3.0000	(0.0000, 4.0000)	25	0.750	$10^{-8}$
	Ours	3.0000	(0.0000, 4.0000)	1	0.01436	$10^{-8}$

$$\begin{aligned}
& -x_5 + x_8 \leq 100, \\
& x_1 - x_1 x_6 + 833.3333333333333 x_4 \\
& \leq 83333.33333333333, \\
& x_2 x_4 - x_2 x_7 - 1250 x_4 + 1250 x_5 \leq 0, \\
& x_3 x_5 - x_3 x_8 - 2500 x_5 \leq -1250000 \\
& 100 \leq x_1 \leq 10000, \\
& 1000 \leq x_2 \leq 10000, \\
& 1000 \leq x_3 \leq 10000, \\
& 10 \leq x_4 \leq 1000, \\
& 10 \leq x_5 \leq 1000, \\
& 10 \leq x_6 \leq 1000, \\
& 10 \leq x_7 \leq 1000, \\
& 10 \leq x_8 \leq 1000.
\end{aligned} \tag{28}$$

$$\begin{aligned}
& \text{s.t. } x_1 + x_2 + 2x_3 + x_4 + x_5 \geq 10, \\
& 2x_1 + 3x_2 + x_5 \geq 8, \\
& x_2 + 4x_3 - x_4 + 2x_5 \geq 12, \\
& 8x_1 - x_2 - x_3 + 6x_4 \geq 20, \\
& 2x_1 + x_2 + 3x_3 + x_4 + x_5 \leq 30, \\
& x_1, x_2, x_3, x_4, x_5 \geq 0.
\end{aligned} \tag{29}$$

Example 11 (st-e26).

$$\begin{aligned}
& \min \quad -3x_1^2 - 5x_1 - 3x_2^2 - 5x_2 \\
& \text{s.t. } 0.7x_1 + x_2 \leq 6.3, \\
& 0.5x_1 + 0.8333x_2 \leq 6, \\
& x_1 + 0.6x_2 \leq 7.08, \\
& 0.1x_1 + 0.25x_2 \leq 1.35, \\
& 0 \leq x_1 \leq 10, \\
& 0 \leq x_2 \leq 30.
\end{aligned} \tag{30}$$

Example 10 (st-qpc-m1).

$$\begin{aligned}
& \min \quad 10x_1 - 0.34x_1x_1 - 0.28x_1x_2 + 10x_2 \\
& \quad - 0.22x_1x_3 + 10x_3 \\
& \quad - 0.24x_1x_4 + 10x_4 - 0.51x_1x_5 + 10x_5 \\
& \quad - 0.28x_2x_1 - 0.34x_2x_2 - 0.23x_2x_3 \\
& \quad - 0.24x_2x_4 - 0.45x_2x_5 - 0.22x_3x_1 - 0.23x_3x_2 \\
& \quad - 0.35x_3x_3 - 0.22x_3x_4 \\
& \quad - 0.34x_3x_5 - 0.24x_4x_1 - 0.24x_4x_2 - 0.22x_4x_3 \\
& \quad - 0.2x_4x_4 - 0.38x_4x_5 \\
& \quad - 0.51x_5x_1 - 0.45x_5x_2 - 0.34x_5x_3 - 0.38x_5x_4 \\
& \quad - 0.99x_5x_5
\end{aligned}$$

These five examples are taken from GLOBALlib; all of them are generalized linear multiplicative programs with different types of constraints. Computational results and some known results are listed in Table 2, where the notations used in the headline have the following meanings: Exam. denotes the serial number of the example tested in this paper; Best sol. and Best val. represent the best optimal solution and optimal value currently known; Our sol. and Our val. are the optimal solution and optimal value obtained by our algorithm described in this paper. From the results summarised in the table, we can see that our algorithm can effectively solve the LMP.

TABLE 2: Results of numerical experiments 7–11.

Exam.	Best sol.	Our sol.	Best val.	Our val.
7	Unknown	(1, 0)	Unknown	0
8	Unknown	(0.9, 0, 0.9)	Unknown	-1
9	$x_{\text{glob}}$	$x_{\text{our}}$	7512.2301449	7512.2301446
10	Unknown	(0, 0, 0, 3.333, 26.6667)	Unknown	-473.778
11	Unknown	(7.08, 0)	Unknown	-185.779196

TABLE 3: Numerical results of Example 12.

$p$	$m$	$n$	Avr. iter.	Std. dev.	Avr. time (s)
5	10	10	5.7	28.2	0.5157
5	10	20	11.4	13.4	1.315
5	10	30	19.0	14.7	5.528
10	10	20	31.4	18.4	41.2140
10	20	40	35.7	17.7	43.541
20	30	40	49.6	19.5	75.208
20	30	50	56.1	22.3	88.233
30	50	80	117.0	14.8	168.213
50	80	100	327.0	24.5	216.209
50	100	150	530.0	58.5	436.139

One has

$$\begin{aligned}
 x_{\text{glob}} &= (1026.9484, 1000, 5485, 265.0597, 280.58873, 134.9403, 284.47098, 380.5887) \\
 x_{\text{our}} &= (1026.9484, 1000, 5485.282, 265.06, 280.589, 134.94, 284.471, 380.589) .
 \end{aligned}
 \tag{31}$$

Example 12 (random test).

$$\begin{aligned}
 \min \quad & f(x) = \sum_{i=1}^p \left( (a^{0i})^T x + b_{0i} \right) \left( (c^{0i})^T x + d_{0i} \right) \\
 \text{s.t.} \quad & \sum_{i=1}^p \left( (a^{1i})^T x + b_{1i} \right) \left( (c^{1i})^T x + d_{1i} \right) \leq 0, \\
 & \sum_{i=1}^p \left( (a^{2i})^T x + b_{2i} \right) \left( (c^{2i})^T x + d_{2i} \right) \geq 0, \\
 & x \in D = \{x \in R^n \mid Ax \leq b\},
 \end{aligned}
 \tag{32}$$

where the real numbers  $b_{0j}$  and  $d_{0j}$  are randomly generated in the range  $[-1, 1]$ ,  $b_{1j}$  is randomly generated in interval  $[-1, 0]$ ,  $d_{1j}$ ,  $b_{2j}$ ,  $d_{2j}$  are randomly generated in the range  $[0, 1]$ , and the real elements of  $a_{ij}$ ,  $c_{ij}$ ,  $A$ , and  $b$  are randomly generated in the range  $[0, 1]$ .

For this problem, we tested 8 groups of instances with different dimension. For each group, we performed 10 instances for a total of 80 instances. The computational results are listed in Table 3, where the notations used in the headline have the

following meanings: Avr. iter.: average numbers of iterations in the algorithm; Std. dev.: standard deviation; Avr. time: average CPU time in seconds;  $m$  and  $n$  denote the numbers of linear constraints and variables, respectively.

### 5. Concluding Remarks

This paper presents a new relaxation method for designing branch-and-bound algorithm for generalized linear multiplicative programming problem with linear multiplicative constraints. The relaxation problem is a convex programming which can be easily obtained with one-step relaxation; it has better approximation effect than usual two-phase linear relaxation method. The presented algorithm can efficiently work without nonnegative restriction to linear function in multiplicative terms, while this restriction is a necessary condition to most branch and bound algorithms described in a lot of literatures. Extensive results of numerical experiments from recent literature show that our method is feasible and effective for this kind of problems.

### Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Acknowledgments

This paper is supported by the Science and Technology Key Project of Education Department of Henan Province (14A110024 and 15A110023), the National Natural Science Foundation of Henan Province (152300410097), the Science and Technology Projects of Henan Province (182102310941), the Cultivation Plan of Young Key Teachers in Colleges and Universities of Henan Province (2016GGJS-107), the Higher School Key Scientific Research Projects of Henan Province (18A110019 and 17A110021), and the Major Scientific Research Projects of Henan Institute of Science and Technology (2015ZD07).

## References

- [1] H. Konno, H. Shirakawa, and H. Yamazaki, "A mean-absolute deviation-skewness portfolio optimization model," *Annals of Operations Research*, vol. 45, no. 1-4, pp. 205–220, 1993.
- [2] M. C. Dorneich and N. V. Sahinidis, "Global optimization algorithms for chip layout and compaction," *Engineering Optimization*, vol. 25, no. 2, pp. 131–154, 1995.
- [3] K. P. Bennett and O. L. Mangasarian, "Bilinear separation of two sets in nspace," *Computational optimization and applications*, vol. 2, no. 3, pp. 207–227, 1993.
- [4] I. Quesada and I. E. Grossmann, "Alternative bounding approximations for the global optimization of various engineering design problems," in *Global Optimization in Engineering Design, Nonconvex Optimization and Its Applications*, I. E. Grossmann, Ed., vol. 9, Kluwer Academic Publishers, Norwell, MA, 1996.
- [5] F. Samadi, A. Mirzazadeh, and M. . Pedram, "Fuzzy pricing, marketing and service planning in a fuzzy inventory model: a geometric programming approach," *Applied Mathematical Modelling: Simulation and Computation for Engineering and Environmental Systems*, vol. 37, no. 10-11, pp. 6683–6694, 2013.
- [6] J. M. Mulvey, R. J. Vanderbei, and S. A. Zenios, "Robust optimization of large-scale systems," *Operations Research*, vol. 43, no. 2, pp. 264–281, 1995.
- [7] H. P. Benson, "Vector maximization with two objective functions," *Journal of Optimization Theory and Applications*, vol. 28, no. 2, pp. 253–257, 1979.
- [8] R. L. Keeney and H. Raika, *Decisions with Multiple Objective*, Cambridge University Press, Cambridge, 1993.
- [9] A. M. Geoffrion, "Solving bicriterion mathematical programs," *Operations Research*, vol. 15, pp. 39–54, 1967.
- [10] H. Konno, P. T. Thach, and H. Tuy, *Optimization on Low Rank Nonconvex Structures*, vol. 15 of *Nonconvex Optimization and its Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.
- [11] O. L. Mangasarian, "Equilibrium points of bimatrix games," *Journal of the Society for Industrial and Applied Mathematics*, vol. 12, pp. 778–780, 1964.
- [12] A. M. Frieze, "A bilinear programming formulation of the 3-dimensional assignment problem," *Mathematical Programming*, vol. 7, no. 1, pp. 376–379, 1979.
- [13] J. E. Falk, "A linear max-min problem," *Mathematical Programming*, vol. 5, pp. 169–188, 1973.
- [14] H. P. Benson, "Global maximization of a generalized concave multiplicative function," *Journal of Optimization Theory and Applications*, vol. 137, no. 1, pp. 105–120, 2008.
- [15] H. Tuy, *Convex Analysis and Global Optimization*, Springer International Publishing AG, Switzerland, 2nd edition, 2016.
- [16] Y. Dai, J. Shi, and S. Wang, "Conical partition algorithm for maximizing the sum of dc ratios," *Journal of Global Optimization*, vol. 31, no. 2, pp. 253–270, 2005.
- [17] H. Konno, Y. Yajima, and T. Matsui, "Parametric simplex algorithms for solving a special class of nonconvex minimization problems," *Journal of Global Optimization*, vol. 1, no. 1, pp. 65–81, 1991.
- [18] N. V. Thoai, "A global optimization approach for solving the convex multiplicative programming problem," *Journal of Global Optimization*, vol. 1, no. 4, pp. 341–357, 1991.
- [19] P. Shen and H. Jiao, "A new rectangle branch-and-pruning approach for generalized geometric programming," *Applied Mathematics and Computation*, vol. 183, no. 2, pp. 1027–1038, 2006.
- [20] P. Shen and H. Jiao, "Linearization method for a class of multiplicative programming with exponent," *Applied Mathematics and Computation*, vol. 183, no. 1, pp. 328–336, 2006.
- [21] C.-F. Wang, S.-Y. Liu, and P.-P. Shen, "Global minimization of a generalized linear multiplicative programming," *Applied Mathematical Modelling: Simulation and Computation for Engineering and Environmental Systems*, vol. 36, no. 6, pp. 2446–2451, 2012.
- [22] H.-W. Jiao, S.-Y. Liu, and Y.-F. Zhao, "Effective algorithm for solving the generalized linear multiplicative problem with generalized polynomial constraints," *Applied Mathematical Modelling: Simulation and Computation for Engineering and Environmental Systems*, vol. 39, no. 23-24, pp. 7568–7582, 2015.
- [23] P. Shen, Y. Duan, and Y. Ma, "A robust solution approach for nonconvex quadratic programs with additional multiplicative constraints," *Applied Mathematics and Computation*, vol. 201, no. 1-2, pp. 514–526, 2008.
- [24] Y. Zhao and S. Liu, "An efficient method for generalized linear multiplicative programming problem with multiplicative constraints," *SpringerPlus*, vol. 5, no. 1, article no. 1302, 2016.
- [25] Y. Wang and Z. Liang, "A deterministic global optimization algorithm for generalized geometric programming," *Applied Mathematics and Computation*, vol. 168, no. 1, pp. 722–737, 2005.
- [26] H. Jiao, "A branch and bound algorithm for globally solving a class of nonconvex programming problems," *Nonlinear Analysis. Theory, Methods & Applications. An International Multidisciplinary Journal*, vol. 70, no. 2, pp. 1113–1123, 2009.

## Research Article

# Local Search Algorithms for the Beam Angles' Selection Problem in Radiotherapy

Guillermo Cabrera-Guerrero <sup>1</sup>, Nivaldo Rodriguez,<sup>1</sup> Carolina Lagos <sup>1</sup>,  
Enrique Cabrera,<sup>2</sup> and Franklin Johnson<sup>3</sup>

<sup>1</sup>Escuela de Ingeniería Informática, Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile

<sup>2</sup>CIMEAV, Universidad de Valparaíso, Valparaíso, Chile

<sup>3</sup>Departamento de Computación e Informática, Universidad de Playa Ancha, Valparaíso, Chile

Correspondence should be addressed to Guillermo Cabrera-Guerrero; [guillermo.cabrera@pucv.cl](mailto:guillermo.cabrera@pucv.cl)

Received 7 December 2017; Accepted 28 March 2018; Published 6 May 2018

Academic Editor: Josefa Mula

Copyright © 2018 Guillermo Cabrera-Guerrero et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

One important problem in radiation therapy for cancer treatment is the selection of the set of beam angles radiation will be delivered from. A primary goal in this problem is to find a beam angle configuration (BAC) that leads to a clinically acceptable treatment plan. Further, this process must be done within clinically acceptable times. Since the problem of selecting beam angles in radiation therapy is known to be extremely hard to solve as well as time-consuming, both exact algorithms and population-based heuristics might not be suitable to solve this problem. In this paper, we compare two matheuristic methods based on local search algorithms, to approximately solve the beam angle optimisation problem (BAO). Although the steepest descent algorithm is able to find locally optimal BACs for the BAO problem, it takes too long before convergence, which is not acceptable in clinical practice. Thus, we propose to use a next descent algorithm that converges quickly to good quality solutions although no (local) optimality guarantee is given. We apply our two matheuristic methods on a prostate case which considers two organs at risk, namely, the rectum and the bladder. Results show that the matheuristic algorithm based on the next descent local search is able to quickly find solutions as good as the ones found by the steepest descent algorithm.

## 1. Introduction

Radiation is one of the most common therapies used to treat patients suffering from cancer. The purpose of radiation therapy is to deliver a dose of radiation to a tumour in order to sterilize all cancer cells and to minimize the collateral effects on the surrounding healthy organs and tissues. Intensity modulated radiation therapy (IMRT) is the most common technique within radiation therapy.

We can separate IMRT problem into three sequential optimisation (sub-)problems: the beam angle optimisation (BAO) problem, the fluence map optimisation (FMO) problem, and the multileaf collimator sequencing problem, Ehr Gott et al. [1]. In the BAO problem, we determine the number and directions of the beam angles we shall use to produce a treatment plan. The set of beams used to treat a patient is

called beam angle configuration (BAC). Then, in the FMO problem, we determine the radiation intensities that will be delivered from each beam angle. The solution to this problem is a vector of intensities that is called fluence map. Finally, a sequence of movements of a physical device called multileaf collimator is computed in order to efficiently deliver the fluence map computed during the previous phases. It is clear from here that the selection of beam angles in the BAO phase has a big impact on the quality of the fluence map computed in the FMO phase. That is, a good combination of beam angles will lead to a good quality fluence map and, consequently, will produce a good quality treatment plan. In this paper the problem of selecting a good quality BAC is addressed.

To measure the quality of a BAC we need to solve the associated FMO, that is, we solve the FMO problem for each evaluated BAC. Computing the optimal fluence map for a

BAC is a time-consuming process. Further, one practical constraint when solving the BAO problem is that we need to obtain results within clinically acceptable times. In our experience, “clinically acceptable times” are around 12 hours (i.e., running algorithms overnight) so treatment planners can decide among alternative treatment plans during the day after patients images have been obtained. Thus, we need to find efficient strategies that produce good quality treatment plans within these time limits; that is, not too many BACs can be evaluated during the optimisation process. For this reason, sophisticated (meta-)heuristic algorithms such as population-based algorithms might not be suitable for solving this problem.

The aim of this paper is to compare two different local search strategies to capture the trade-off between solutions quality and the required time before convergence. Using the next descent algorithm, we aim to accelerate the search without any major impairment to treatment plans’ quality.

The remainder of this paper is organised as follows. In Section 2 we introduce the BAO problem we aim to solve in this paper and a brief review on different approaches dealing with the BAO problem is presented. In Section 3 both the steepest descent and the next descent algorithms are presented and their main differences are highlighted. In Section 4 the set of instances considered in our experiments are presented. Obtained results are also discussed in this section. Finally, in Section 5 some conclusions are drawn and the future work is outlined.

## 2. Intensity Modulated Radiation Therapy: An Overview

During the last three decades or so, many researchers have worked on the problem of finding efficient treatment plans for radiation therapy for cancer treatment. Most of their efforts have been focused on the problem of finding the optimal fluence map that can be delivered to the patients given a predefined BAC (see, e.g., [2–5]). Unfortunately, significantly less attention has been paid to the problem of finding the best BAC. This is, in part, because the BAO problem is, from a mathematical point of view, much harder to solve than the FMO problem. Further, treatment planners rely on their experience to choose BACs that are good enough to produce clinically acceptable treatment plans. Figure 1 shows the traditional try-and-error process followed by treatment planners in clinical practice.

As we can see, treatment planners try an initial BAC based on their experience and judgement. If no clinically acceptable treatment plan can be produced using the initial BAC, then the expert proposes a new BAC to be tested. This process is repeated until a clinically acceptable treatment plan is produced. Depending on the particular case and the treatment planner experience, this process can take very long.

The first attempt to study the BAO problem from a theoretical point of view is presented in Bortfeld and Schlegel [6]. Since then, both exact and heuristic methods have been proposed to solve this problem. For instance, Rocha et al. [7] propose a guided pattern search method to solve the BAO problem. Genetic algorithms have also been used to explore

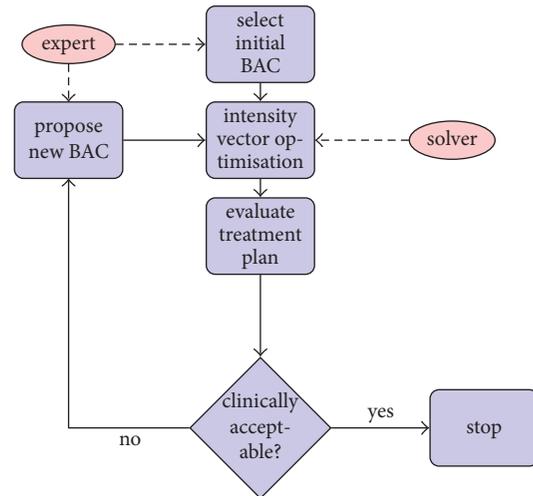


FIGURE 1: Sequential approach to find a good quality BAC.

the beam angles space in Dias et al. [8]; Lei and Li [9]; and Li et al. [10]. In Li et al. [11], the authors propose an ant colony optimisation algorithm that uses a similar representation to Li et al. [10]. In Li and Yao [3], authors propose a hybrid algorithm that combines ant colony optimisation and genetic algorithms. The authors claim that their approach is faster than previously implemented genetic algorithms. A particle swarm optimisation algorithm has also been proposed in Li et al. [12]. In Aleman et al. [13], the authors propose a surface response to find high-quality BACs. We need to highlight at this point that the population-based algorithms as well as the evolutionary algorithms cited above address the BAO problem with only linear or quadratic objective functions and constraints. Thus, they can evaluate a large number of BACs within time limits. This is not the case for our study, as the objective function and constraints we consider here are highly complex nonlinear functions. We will explain this at the end of this section.

Local search approaches have also been proposed. For instance, in Aleman et al. [14] local search algorithms, namely, simulated annealing and add/drop, are implemented. Authors highlight the fact that good quality BACs can be generated using this kind of strategies. Further, they also propose a neighbourhood structure that allows for faster convergence, reducing the time required to obtain a good quality plan. Simulated annealing is also used in Mohammadi et al. [15] and Dias et al. [16].

Methods that combine heuristic and mathematical programming have also been proposed. For instance, a novel method that combines simulated annealing with gradient-based search is presented in Bertsimas et al. [17]. Similarly, in Craft [18], authors combine a local search algorithm with linear programming and gradient search. In Lin et al. [19], authors propose a two-stage heuristic method where, in the first stage, Benders decomposition is considered. Obtained solutions are then deputed during the second stage using local branching.

In Cabrera et al. [20] the authors propose a two-phase approach to solve the multiobjective version of the BAO

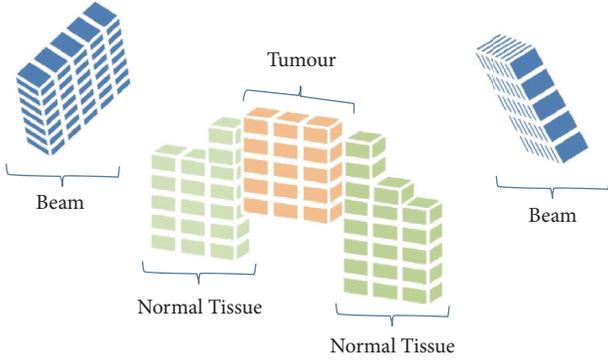


FIGURE 2: Graphic representation of three different regions discretised into voxels and two beam angles discretised into beamlets.

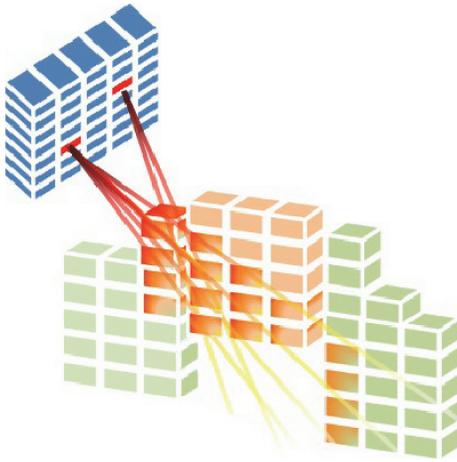


FIGURE 3: Illustration of the radiation coming from two beamlets. Radiation hits not only voxels on the tumour but also some voxels on healthy tissue.

problem. One distinctive feature of their approach is that they use a single objective local search during the first phase to produce efficient treatment plans.

In all the approaches above, the quality of a BAC is determined by the best fluence map such a BAC is able to produce. Thus, for each BAC we want to evaluate, we must solve its associated FMO problem. To mathematically model the BAO problem, each beam angle is discretised into a set of beamlets that radiation is delivered from. Similarly, organs at risk (OARs) and the tumour are discretised into a set of voxels. Figure 2 shows a representation of the problem considering two beam angles (labelled as *beam*), one tumour (labelled as *Tumour*), and two organs at risk (labelled as *Normal Tissue*). As we can see, each beam is discretised forming a “grid” of beamlets (in blue), while the tumour and OARs are divided into voxels (in orange and green, resp.)

Although all beam angles around the patient body are pointing to the centre of the tumour, we cannot avoid the fact that the radiation delivered from beamlets hits not only the tumour but also some OARs (see Figure 3). Thus, finding fluence maps that generate uniform doses on the tumour and

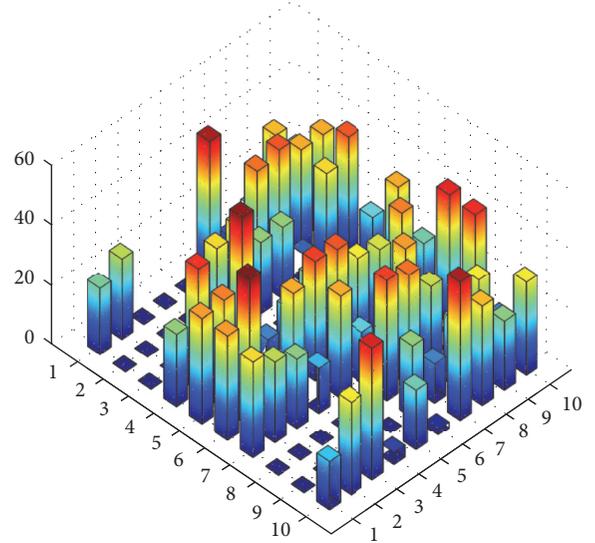


FIGURE 4: Example of a fluence map corresponding to one beam angle with  $10 \times 10$  beamlets.

that avoid as much as possible overirradiating the OARs is a key task during the treatment plan generation process.

A fluence map specifies the total time each beamlet is allowed to irradiate the tumour. Figure 4 shows an example of a fluence map for a beam angle. In this illustration, the beam angle is divided into a grid of  $10 \times 10$  beamlets. Using a multileaf collimator (a physical device consisting of a set of leaves that can block the radiation coming from the linear accelerator), radiation can be modulated so we can determine how much radiation is delivered from each beamlet. Radiation delivered from each beamlet is represented as a bar in Figure 4, where higher bars mean more radiation. While some beamlets will deliver no radiation during the treatment plan, others will irradiate not only the tumour but also some OARs. Discretizing beam angles into beamlets and organs into voxels allows modelling the problem as an optimisation problem. We first let  $K = \{k\pi/36 : k = 0, 1, 2, \dots, 71\}$  be the set of beam angles that radiation can be delivered from. Although in this paper we only consider coplanar angles, our algorithms can be easily extended to noncoplanar angles. Angular resolutions other than  $\pi/36$  can also be used. Let  $\mathcal{A} \in \mathcal{P}^N(K)$  be a BAC where  $\mathcal{P}^N(K)$  is the set of all  $N$ -element subsets of  $K$ , with  $N > 0$  being the number of beam angles we shall consider within a BAC. We denote the  $i$ th angle of BAC  $\mathcal{A}$  by  $\mathcal{A}_i$ , for  $i = 1, \dots, N$ .

As mentioned before, in the BAO problem we need to solve, for each evaluated BAC, the associated FMO problem. Thus, we first introduce the mathematical model of the FMO problem we consider in this work for a fixed BAC  $\mathcal{A} \in \mathcal{P}^N(K)$ :

$$f(\mathcal{A}) = \min_{x \in \mathcal{X}(\mathcal{A})} z(x), \quad (1)$$

with  $z : \mathbb{R}_{\geq 0}^n \rightarrow \mathbb{R}_{\geq 0}$ , where  $z(x)$  is an objective function and the fluence map  $x \in \mathcal{X}(\mathcal{A}) \subseteq \mathbb{R}^n$  is a nonnegative vector with each component  $x_i$ , called *fluence*, representing the length of

time that a patient is exposed to beamlet  $i$  and where  $n$  is the total number of beamlets summed over all  $|K|$  possible beam angles. The set  $\mathcal{X}(\mathcal{A})$  is the set of all feasible solutions of the FMO problem when BAC  $\mathcal{A}$  is considered. Only beamlets  $x_i$  that belong to a beam angle in  $\mathcal{A}$  are allowed to have a value greater than zero.

Different objective functions have been proposed in the literature to solve the FMO problem. The vast majority of this functions need to compute the total radiation dose deposited into each voxel  $j$  of the tumour and OARs by fluence map  $x$ . This dose delivered on region  $r$  by a fluence map  $x$  is represented by a vector we call  $d^r(x) \in \mathbb{R}^{m^r}$ , where  $m^r$  is the total number of voxels in region  $r$  and each of its elements  $d_j^r$  is calculated as in (2) (see [1]).

$$d_j^r(x) = \sum_{i=1}^n A_{ji}^r x_i \quad \forall j = 1, 2, \dots, m^r, \quad (2)$$

where  $r \in R = \{O_1, \dots, O_Q, T\}$  is an element of the index set of regions  $R$  where  $r = T$  refers to the tumour and  $r = O_q$  refers to the  $q$ -th OAR with  $q \in \{1, \dots, Q\}$ . Matrix  $A^r \in \mathbb{R}^{m^r \times n}$ , called dose deposition matrix, is a given matrix where its elements  $A_{ji}^r \geq 0$  determine how much radiation is delivered for each beamlet  $i$  to each voxel  $j$  per time unit at region  $r$ .

In this paper we consider an objective function originally proposed by Wu et al. [4] and called the *logistic function*. This function is based on a well-known measure used in radiation therapy called the *generalised equivalent uniform dose* (gEUD) which was proposed in the work by Niemierko [21]. The gEUD is defined in the work by Niemierko [21] as “the equivalent dose, which if distributed uniformly across the tumour, would lead to the same level of cell killing as the actual dose distribution of interest.” Mathematically, the gEUD is as follows:

$$\text{gEUD}^r(x) = \left( \frac{1}{m^r} \sum_{j=1}^{m^r} (d_j^r(x))^{a^r} \right)^{1/a^r}, \quad (3)$$

where  $a^r$  is a region dependant parameter defined by treatment planners.

The logistic function originally proposed in Wu et al. [4] to compute the optimal fluence map for an associated BAC is

$$\begin{aligned} \min_{x \in \mathcal{X}(\mathcal{A})} z(x) & \\ &= -\ln L(x; T, \nu_T, \text{eud}_0^T) \\ &\quad - \sum_{q=1}^Q \ln U(x; O_q, \nu_{O_q}, \text{eud}_0^{O_q}), \end{aligned} \quad (4)$$

where

$$L(x; T, \nu_T, \text{eud}_0^T) = \left( 1 + \left( \frac{\text{eud}_0^T}{\text{gEUD}^T(x)} \right)^{\nu_T} \right)^{-1}, \quad (5)$$

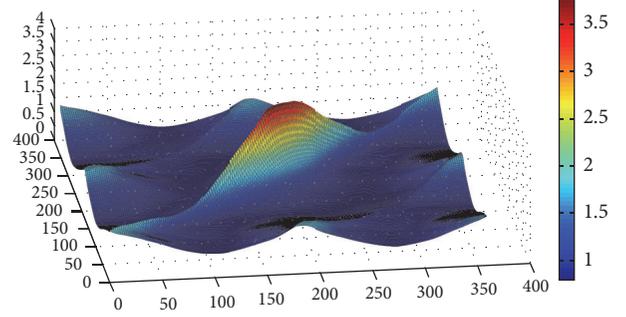


FIGURE 5: Surface of the logistic function in (7) for all possible BACs that consider only two beam angles.

$$\begin{aligned} U(x; O_q, \nu_{O_q}, \text{eud}_0^{O_q}) & \\ &= \left( 1 + \left( \frac{\text{gEUD}^{O_q}(x)}{\text{eud}_0^{O_q}} \right)^{\nu_{O_q}} \right)^{-1}. \end{aligned} \quad (6)$$

Parameters  $\text{eud}_0^T$  and  $\text{eud}_0^{O_q}$  are the prescribed gEUD value for tumour and the maximum gEUD value allowed for OARs, respectively, and  $\nu_T, \nu_{O_q} > 0$  are user-defined parameters that indicate the importance of the tumour and the  $q$ -th OAR, respectively. gEUD-based objective functions as the ones in (4) have been largely used to solve the FMO problem (e.g., [4, 5, 20, 22]).

In this paper we slightly modify the logistic function in (4) by removing (5) from the objective function and adding a constraint to the model setting gEUD of the tumour to  $t$ , as proposed in Cabrera et al. [20]. Thus, objective function  $z$  of the FMO problem we consider in this paper is as follows:

$$\begin{aligned} \min_{x \in \mathcal{X}(\mathcal{A})} z(x) &= - \sum_{q=1}^Q \ln U(x; O_q, \nu_{O_q}, \text{eud}_0^{O_q}) \\ \text{s.t.} \quad &\text{gEUD}^T(x) \geq \text{eud}_0^T. \end{aligned} \quad (7)$$

As discussed in the work by Cabrera et al. [20], the  $\text{gEUD}^T(x)$  is a convex function of  $x$  and  $-\ln(U)$  is also convex in  $\text{gEUD}^T(x)$  for the region of interest. Thus, problem (7) is convex and, therefore, optimal solutions can be found by exact algorithms that can deal with nonlinear problems.

Having defined the FMO problem, we can now introduce the mathematical model for the BAO problem we aim to solve in this study.

$$\begin{aligned} \min_{\mathcal{A} \in \mathcal{A}} f(\mathcal{A}) &= \min_{\mathcal{A} \in \mathcal{A}} \min_{x \in \mathcal{X}(\mathcal{A})} z(x) \\ \text{s.t.} \quad &\text{gEUD}^T(x) \geq \text{eud}_0^T. \end{aligned} \quad (8)$$

The BAO problem has been shown to be nonconvex and highly nonlinear. As an example of this, Figure 5 shows a surface plot for the logistic function in (7) for all possible BACs that consider two beam angles. As we can see, the surface presents several combinations of beams where locally

```

Input:  $N$  (Number of angles in a BAC)
Output:  $\mathcal{A}$  (Locally optimal BAC)
begin
   $\mathcal{A} = \text{initialRandomSolution}(\mathcal{P}^N(K));$ 
  localOptimum = false;
  repeat
     $\mathcal{N} = \text{generateNeighbourhood}(\mathcal{A});$ 
     $\mathcal{A}' = \text{argmin}_{\mathcal{B} \in \mathcal{N}} f(\mathcal{B});$ 
    if  $f(\mathcal{A}') < f(\mathcal{A})$  then
       $\mathcal{A} = \mathcal{A}'$ 
    else
      localOptimum = true;
  until localOptimum;
  return ( $\mathcal{A}$ );

```

ALGORITHM 1: Steepest descent local search algorithm.

optimal solutions can be found. Clearly, as more beam angles are added, the problem becomes much harder to solve. Also, we can note that worst values are obtained when both angles within the BAC are the same (or too close to each other). Opposite angles, that is, with a difference around  $180^\circ$ , also obtain high values. We will discuss this point in the conclusions of this study.

### 3. Local Search-Based Heuristic Algorithms for BAO

As mentioned before, many heuristic algorithms have been proposed in the literature to solve the BAO problem. In this paper we propose to solve this problem by using local search algorithms. First, we implement a simple yet efficient local search algorithm called steepest descent, which was previously used in Cabrera et al. [20] to solve the multiobjective version of the BAO problem. Algorithm 1 shows the pseudocode of this method.

The steepest descent algorithm starts with an initial BAC, which can be either provided by the treatment planner or randomly generated. This initial solution is labelled as the current solution. For this current solution, a set of neighbours is generated by applying the neighbourhood move  $\mathcal{N}()$  (as explained in the paragraphs below) on the current solution. Once the entire neighbourhood defined by  $\mathcal{N}()$  has been generated, the best neighbourhood with respect to its objective function is selected. If the best neighbour solution is better than the current one, then the best neighbour is labelled as the new current solution. If the best neighbour is not better than the current solution the algorithm stops and the current solution is returned as the locally optimal solution the algorithm found. As before, the steepest descent algorithm is very simple to implement and it converges to good quality solutions. However, since it evaluates the entire neighbourhood of the current solution at each iteration, sometimes it can take too long to converge. Since one mandatory requirement in clinical practice in radiation therapy is to provide good quality treatment plans within

```

Input:  $N$  (Number of angles in a BAC)
Output:  $\mathcal{A}$  (Locally optimal BAC)
begin
   $\mathcal{A} = \text{initialRandomSolution}(\mathcal{P}^N(K));$ 
  localOptimum = false;
  repeat
     $\mathcal{N} = \text{generateNeighbourhood}(\mathcal{A});$ 
    foreach  $\mathcal{B} \in \mathcal{N}$  do
      if  $f(\mathcal{B}) < f(\mathcal{A})$  then
         $\mathcal{A} = \mathcal{B};$ 
        break;
       $\mathcal{N} = \mathcal{N} \setminus \mathcal{B};$ 
    if  $\mathcal{N} = \emptyset$  then
      localOptimum = true;
  until localOptimum;
  return ( $\mathcal{A}$ );

```

ALGORITHM 2: Next descent local search algorithm.

clinically acceptable times, algorithms such as the steepest descent might not be suitable. Thus, in this paper, we propose an algorithm that is able to produce treatment plans faster than the steepest descent without any major impact on the quality of the obtained treatment plans. Algorithm 2 shows this algorithm called the next descent algorithm. Just as in the steepest descent algorithm, the next descent algorithm starts with an initial BAC which is also labelled as the current solution. Then, for each element within the neighbourhood of the current solution the FMO problem is solved and the quality of the resulting treatment plan is compared to the current solution. If the neighbour is better than the current solution, then the neighbour is labelled as the new current solution and its neighbourhood is generated.

If the current solution is better than the neighbour solution, then the neighbour solution is removed from the neighbourhood and the next neighbour is evaluated. The algorithm stops once the neighbourhood is empty; that is, no neighbour solution is better than the current solution. Since the next descent algorithm does not explore the entire neighbourhood of the current solution at each iteration, it needs fewer objective function evaluations, leading to a faster convergence.

Both the steepest descent and the next descent algorithms need a neighbourhood to be defined. In this paper we consider the following neighbourhood:

$$\mathcal{N}(\mathcal{A}) = \left\{ \mathcal{B} \in \mathcal{P}^N(K) : \mathcal{A}_j = \mathcal{B}_j \pm \frac{k\pi}{180} \text{ for some } j \in \{1, \dots, N\}, \mathcal{A}_i = \mathcal{B}_i, \forall i = 1, \dots, N, \text{ where } i \neq j \right\}, \quad (9)$$

where, again,  $k = \{0, 1, 2, \dots, 71\}$ . Given this neighbourhood definition, we know that  $2 \times N$  neighbours will be generated. Further, in some cases, we can have that  $\mathcal{B}_i = \mathcal{B}_j$  with  $i \neq j$ . We allow neighbours to repeat a beam angle although it is well

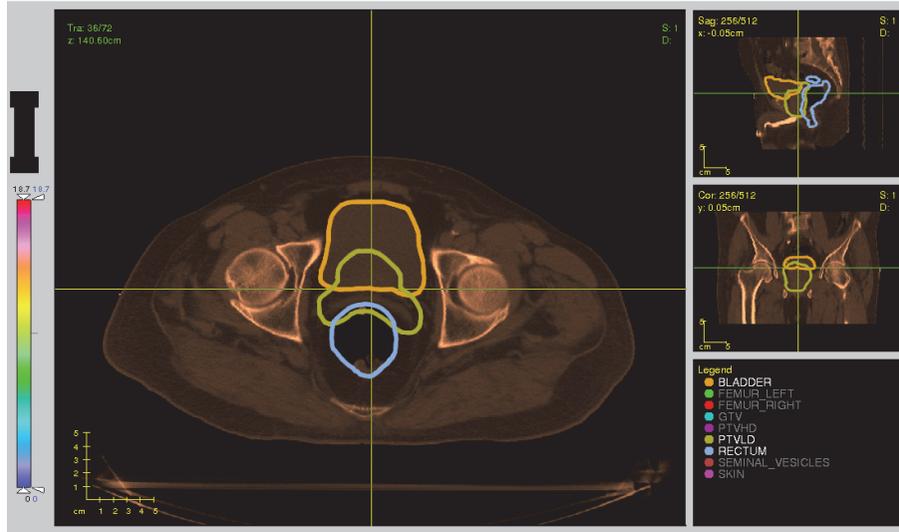


FIGURE 6: Prostate case from CERR. Two OARs (bladder and rectum) are considered.

known that this situation impacts the neighbour quality. We need to point out that other neighbourhood definitions that support noncoplanar beam angles can be found in Lim and Cao [23]; Mišić et al. [24]; Yarmand and Craft [25].

#### 4. Computational Experiments

We present in this section the computational experiments carried out in this work. We use an Intel i7 processor and 32 GB of RAM to run our experiments. Linux 16.04 was the operating system. Both the steepest descent and the next descent algorithms were coded in JAVA 8 language using NetBeans IDE. Experiments are performed on a prostate obtained from the Computational Environment for Radiotherapy Research (CERR, <http://www.cerr.info/>). The tumour is located at the prostate while OARs considered in this study are the bladder and the rectum. We need to point out that prostate cases are, in general, very difficult as the tumour and OARs are, as in our case, overlapped (see Figure 6).

In this prostate case, the tumour has more than 7,000 voxels, the rectum about 5,500, and the bladder around 9,500. Although the same multileaf collimator is used for all beam angles the number of beamlets considered in the optimisation process depends on each beam angle. This is because those beamlets that do not irradiate at least one voxel of the tumour are not considered as decision variables in the optimisation process. Thus, the number of decision variables (beamlets) depends on the BAC and ranges between 160 and 220. The number of beam angles  $N$  considered in a BAC is equal to 5. As we mentioned before, we use IPOPT Wächter and Biegler [26] as the solver for all nonlinear optimisation problems.

Table 1 shows the value of the parameters of the logistic function used for our experiments. These values are the same as in Cabrera et al. [20].

Three sets of initial BACs are generated for our experiments. The first set ( $\mathcal{A}$ ) is a set of 14 BACS for which beam angles are equidistant. The second set ( $\mathcal{B}$ ) is a set of initial BACs for which beam angles have been selected randomly but

TABLE 1: Parameters of the logistic function and gEUD.

$r$	$a^r$	$\nu^r$	$\text{eud}_0^r$
Tumour	-10	12	75
Rectum	8	8	50
Bladder	2	5	50

considering some geometrical constraints such as avoiding opposite beam angles and angles that are too close to each other. The third set of initial BACs ( $\mathcal{C}$ ) consists of BACs for which beam angles have been selected absolutely randomly. All BACs evaluated in our experiments consist of 5 beam angles. We need to point out that for the SD algorithm only one run per initial BAC is needed, as it is a deterministic algorithm. For the ND, 10 independent runs are performed for each initial BAC. Reported results in the next section correspond to the average of these 10 runs for each initial BAC.

**4.1. Results.** A summary of the obtained results is presented in this section. Tables 2, 3, and 4 show the results obtained by both the steepest descent (SD) and the next descent (ND) algorithms for instance sets  $\mathcal{A}$ ,  $\mathcal{B}$ , and  $\mathcal{C}$ , respectively. As we can see, not only is the next descent algorithm faster than the steepest descent algorithm in all but three instances in total but also it performs slightly better, in average, than the steepest descent algorithm in sets  $\mathcal{A}$  and  $\mathcal{C}$ . For set  $\mathcal{B}$ , the steepest descent performs slightly better than the next descent algorithm. These results confirm that we can use faster algorithms without any major impairment in the treatment plan quality.

Further, using faster algorithms such as the next descent algorithm will allow us to, for instance, explore different parts of the beam angle space by restarting the algorithm from different initial BACs. Moreover, this type of algorithms might also be used within more complex frameworks that combine different optimisation algorithms.

TABLE 2: A comparison of the results obtained by both the SD and ND algorithms for the instance set  $\mathcal{A}$ , with respect to their objective function values and number of function evaluations.

Instance	$f(\mathcal{A})_{SD}$	$f(\mathcal{A})_{ND}$	Savings on $f$	$\#f_{SD}$	$\#f_{ND}$	Savings on $\#f$
$\mathcal{A}_1$	0.723	0.713	1.41%	81	69	14.81%
$\mathcal{A}_2$	0.723	0.723	0.00%	70	58	17.14%
$\mathcal{A}_3$	0.723	0.724	-0.06%	81	54	33.33%
$\mathcal{A}_4$	0.725	0.725	0.00%	71	38	46.48%
$\mathcal{A}_5$	0.725	0.725	0.07%	50	35	30.00%
$\mathcal{A}_6$	0.719	0.725	-0.88%	91	59	35.16%
$\mathcal{A}_7$	0.718	0.718	0.01%	130	92	29.23%
$\mathcal{A}_8$	0.718	0.719	-0.10%	157	95	39.49%
$\mathcal{A}_9$	0.718	0.718	0.00%	156	100	35.90%
$\mathcal{A}_{10}$	0.733	0.718	2.10%	81	129	-59.26%
$\mathcal{A}_{11}$	0.713	0.732	-2.69%	141	46	67.38%
$\mathcal{A}_{12}$	0.714	0.713	0.06%	120	82	31.67%
$\mathcal{A}_{13}$	0.714	0.714	0.14%	100	61	39.00%
$\mathcal{A}_{14}$	0.715	0.714	0.10%	80	48	40.00%
$\bar{x}$	<b>0.720</b>	<b>0.720</b>	<b>0.01%</b>	<b>101</b>	<b>69</b>	<b>28.60%</b>

TABLE 3: A comparison of the results obtained by both the SD and ND algorithms for the instance set  $\mathcal{B}$ , with respect to their objective function values and number of function evaluations.

Instance	$f(\mathcal{A})_{SD}$	$f(\mathcal{A})_{ND}$	Savings on $f$	$\#f_{SD}$	$\#f_{ND}$	Savings on $\#f$
$\mathcal{B}_1$	0.729	0.719	1.38%	91	71	21.98%
$\mathcal{B}_2$	0.734	0.738	-0.53%	141	91	35.46%
$\mathcal{B}_3$	0.713	0.713	0.00%	201	127	36.82%
$\mathcal{B}_4$	0.738	0.733	0.66%	201	91	54.73%
$\mathcal{B}_5$	0.712	0.719	-0.92%	141	54	61.70%
$\mathcal{B}_6$	0.723	0.722	0.17%	111	59	46.85%
$\mathcal{B}_7$	0.723	0.723	0.00%	231	172	25.54%
$\mathcal{B}_8$	0.715	0.715	0.00%	161	97	39.75%
$\mathcal{B}_9$	0.721	0.718	0.46%	281	126	55.16%
$\mathcal{B}_{10}$	0.714	0.718	-0.56%	220	137	37.73%
$\mathcal{B}_{11}$	0.712	0.718	-0.79%	111	46	58.56%
$\mathcal{B}_{12}$	0.714	0.733	-2.75%	200	46	77.00%
$\mathcal{B}_{13}$	0.713	0.730	-2.35%	228	69	69.74%
$\mathcal{B}_{14}$	0.726	0.726	0.00%	181	162	10.50%
$\mathcal{B}_{15}$	0.719	0.725	-0.81%	141	106	24.82%
$\bar{x}$	<b>0.720</b>	<b>0.723</b>	<b>-0.40%</b>	<b>176</b>	<b>97</b>	<b>43.76%</b>

Figure 7 shows the % of improvement (savings) of the next descent algorithm compared to the steepest descent algorithm for the all 44 instances evaluated in this study. Positive values mean next descent performs better. Negative values mean steepest descent performs better.

Similar to Figure 7, Figure 8 shows the variations in the number of iterations each algorithm needs to perform before convergence. Again, positive values mean that next descent algorithm was faster. It is interesting to note that, for some few instances ( $\mathcal{A}_{10}$ ,  $\mathcal{C}_3$ , and  $\mathcal{C}_{12}$ ), the next descent algorithm took longer (4,897; 1,280; and 6,483 seconds, resp.) than the steepest descent. Although possible, this is very unusual.

As a summary, results show that the next descent algorithm we implement in this paper is able to obtain treatment plans that are quite competitive with respect to the ones obtained by the steepest descent algorithm. The next descent algorithm takes a fraction of the time that the steepest descent takes, though.

## 5. Conclusions

In this paper we have implemented two local search methods, namely, the steepest descent and the next descent algorithms. While the former needs to explore the entire neighbourhood

TABLE 4: A comparison of the results obtained by both the SD and ND algorithms for the instance set  $\mathcal{C}$ , with respect to their objective function values and number of function evaluations.

Instance	$f(\mathcal{A})_{SD}$	$f(\mathcal{A})_{ND}$	Savings on $f$	$\#f_{SD}$	$\#f_{ND}$	Savings on $\#f$
$\mathcal{C}_1$	0.746	0.746	0.00%	111	53	52.25%
$\mathcal{C}_2$	0.709	0.709	0.00%	121	61	49.59%
$\mathcal{C}_3$	0.741	0.715	3.56%	151	167	-10.60%
$\mathcal{C}_4$	0.723	0.721	0.30%	191	64	66.49%
$\mathcal{C}_5$	0.725	0.725	0.00%	171	103	39.77%
$\mathcal{C}_6$	0.729	0.728	0.07%	201	115	42.79%
$\mathcal{C}_7$	0.710	0.710	0.00%	131	66	49.62%
$\mathcal{C}_8$	0.718	0.718	-0.01%	101	76	24.75%
$\mathcal{C}_9$	0.724	0.724	0.00%	91	59	35.16%
$\mathcal{C}_{10}$	0.720	0.729	-1.22%	151	50	66.89%
$\mathcal{C}_{11}$	0.731	0.738	-0.95%	181	103	43.09%
$\mathcal{C}_{12}$	0.746	0.730	2.13%	110	190	-72.73%
$\mathcal{C}_{13}$	0.708	0.708	0.00%	231	104	54.98%
$\mathcal{C}_{14}$	0.727	0.723	0.49%	161	90	44.10%
$\mathcal{C}_{15}$	0.742	0.742	0.00%	171	79	53.80%
Average	0.727	0.724	0.29%	152	92	36.00%

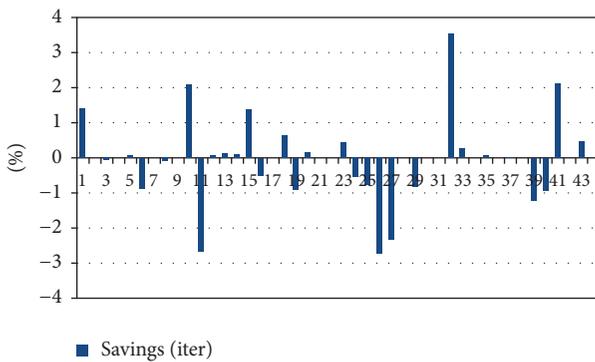


FIGURE 7: Savings (%) in the objective function value for the entire set of instances.

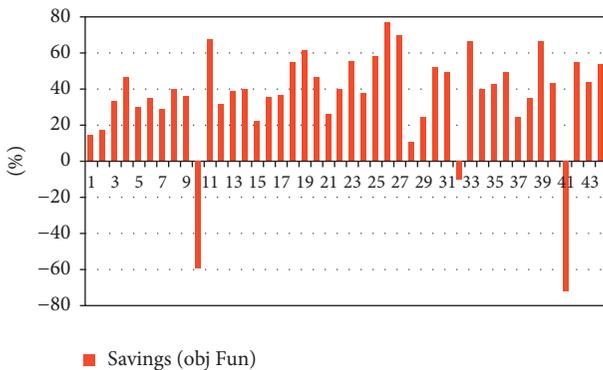


FIGURE 8: Savings (%) in the number of iterations each algorithm needs before convergence.

at each iteration, the latter does not. As a consequence, the next descent algorithm converges much faster than the next descent algorithm, as it explores a smaller part of the solution space.

We try these two local search methods on a complex problem arising in radiation therapy for cancer treatment called the beam angle optimisation problem. Both algorithms are able to produce clinically acceptable treatment plans. However, the next descent algorithm converges to locally optimal solutions much faster than the steepest descent. Results show that the quality of the treatment plans obtained by the steepest descent is, in average, slightly better than the ones obtained by the next descent algorithm. However, for almost half of the instances we test in this paper, the next descent algorithm demonstrates to perform better than the steepest descent.

As a future work, other local search algorithms such as Tabu Search and Variable Neighbourhood Search must be tried with the aim of obtaining high-quality treatment plans within clinically acceptable times. Moreover, the multiobjective version of the next descent algorithm can be tested on the MO-BAO problem.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

Guillermo Cabrera-Guerrero wishes to acknowledge both DI 039.416/2017 Project and CONICYT/FONDECYT/INICIACION/11170456 Project for partially supporting this research.

## References

- [1] M. Ehrgott, C. Güler, H. W. Hamacher, and L. Shao, "Mathematical optimization in intensity modulated radiation therapy," *Annals of Operations Research*, vol. 175, pp. 309–365, 2009.
- [2] M. Ehrgott, A. Holder, and J. Reese, "Beam selection in radiotherapy design," *Linear Algebra and its Applications*, vol. 428, no. 5-6, pp. 1272–1312, 2008.

- [3] Y. Li and D. Yao, "Accelerating the radiotherapy planning with a hybrid method of genetic algorithm and ant colony system," in *Advances in Natural Computation*, vol. 4222 of *Lecture Notes in Computer Science*, pp. 340–349, Springer, Berlin, Germany, 2006.
- [4] Q. Wu, R. Mohan, and A. Niemierko, "IMRT optimization based on the generalized equivalent uniform dose (EUD)," in *Proceedings of the 22nd Annual International Conference of the IEEE*, J. Enderle, Ed., vol. 1, pp. 710–713, 2000.
- [5] Q. Wu, D. Djajaputra, Y. Wu, J. Zhou, H. H. Liu, and R. Mohan, "Intensity-modulated radiotherapy optimization with gEUD-guided dose-volume objectives," *Physics in Medicine and Biology*, vol. 48, no. 3, pp. 279–291, 2003.
- [6] T. Bortfeld and W. Schlegel, "Optimization of beam orientations in radiation therapy: some theoretical considerations," *Physics in Medicine and Biology*, vol. 38, no. 2, article 006, pp. 291–304, 1993.
- [7] H. Rocha, J. M. Dias, B. C. Ferreira, and M. C. Lopes, "Beam angle optimization for intensity-modulated radiation therapy using a guided pattern search method," *Physics in Medicine and Biology*, vol. 58, no. 9, pp. 2939–2953, 2013.
- [8] J. Dias, H. Rocha, B. Ferreira, and M. Lopes, "A genetic algorithm with neural network fitness function evaluation for IMRT beam angle optimization," *Central European Journal of Operations Research*, vol. 22, no. 3, pp. 431–455, 2014.
- [9] J. Lei and Y. Li, "An approaching genetic algorithm for automatic beam angle selection in IMRT planning," *Computer Methods and Programs in Biomedicine*, vol. 93, no. 3, pp. 257–265, 2009.
- [10] Y. Li, J. Yao, and D. Yao, "Automatic beam angle selection in IMRT planning using genetic algorithm," *Physics in Medicine and Biology*, vol. 49, no. 10, pp. 1915–1932, 2004.
- [11] Y. Li, D. Yao, W. Chen, J. Zheng, and J. Yao, "Ant colony system for the beam angle optimization problem in radiotherapy planning: a preliminary study," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, D. Corne, Ed., volume 2, pp. 1532–1538, IEEE, Edinburgh, Scotland, UK, September 2005.
- [12] Y. Li, D. Yao, J. Yao, and W. Chen, "A particle swarm optimization algorithm for beam angle selection in intensity-modulated radiotherapy planning," *Physics in Medicine and Biology*, vol. 50, no. 15, pp. 3491–3514, 2005.
- [13] D. M. Aleman, H. E. Romeijn, and J. F. Dempsey, "A response surface approach to beam orientation optimization in intensity-modulated radiation therapy treatment planning," *INFORMS Journal on Computing*, vol. 21, no. 1, pp. 62–76, 2009.
- [14] D. M. Aleman, A. Kumar, R. K. Ahuja, H. E. Romeijn, and J. F. Dempsey, "Neighborhood search approaches to beam orientation optimization in intensity modulated radiation therapy treatment planning," *Journal of Global Optimization*, vol. 42, no. 4, pp. 587–607, 2008.
- [15] S. Mohammadi, C. Shang, Z. Ouhib, T. Leventouri, and G. Kalantzis, "A computational study on different penalty approaches for constrained optimization in radiation therapy treatment planning with a simulated annealing algorithm," in *Proceedings of the 16th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD '15)*, June 2015.
- [16] J. Dias, H. Rocha, B. Ferreira, and M. D. C. Lopes, "Simulated annealing applied to IMRT beam angle optimization: a computational study," *Physica Medica*, vol. 31, no. 7, pp. 747–756, 2015.
- [17] D. Bertsimas, V. Cacchiani, D. Craft, and O. Nohadani, "A hybrid approach to beam angle optimization in intensity-modulated radiation therapy," *Computers & Operations Research*, vol. 40, no. 9, pp. 2187–2197, 2013.
- [18] D. Craft, "Local beam angle optimization with linear programming and gradient search," *Physics in Medicine and Biology*, vol. 52, no. 7, article no. N02, pp. N127–N135, 2007.
- [19] S. Lin, G. J. Lim, and J. F. Bard, "Benders decomposition and an IP-based heuristic for selecting IMRT treatment beam angles," *European Journal of Operational Research*, vol. 251, no. 3, pp. 715–726, 2016.
- [20] G. Cabrera G., M. Ehrgott, A. J. Mason, and A. Raith, "A mathematical approach to solve the multiobjective beam angle optimization problem in intensity-modulated radiation therapy," *International Transactions in Operational Research*, vol. 25, no. 1, pp. 243–268, 2018.
- [21] A. Niemierko, "Reporting and analyzing dose distributions: a concept of equivalent uniform dose," *Medical Physics*, vol. 24, no. 1, pp. 103–110, 1997.
- [22] S. Das, T. Cullip, G. Tracton et al., "Beam orientation selection for intensity-modulated radiation therapy based on target equivalent uniform dose maximization," *International Journal of Radiation Oncology Biology Physics*, vol. 55, no. 1, pp. 215–224, 2003.
- [23] G. J. Lim and W. Cao, "A two-phase method for selecting IMRT treatment beam angles: branch-and-prune and local neighborhood search," *European Journal of Operational Research*, vol. 217, no. 3, pp. 609–618, 2012.
- [24] V. V. Mišić, D. M. Aleman, and M. B. Sharpe, "Neighborhood search approaches to non-coplanar beam orientation optimization for total marrow irradiation using IMRT," *European Journal of Operational Research*, vol. 205, no. 3, pp. 522–527, 2010.
- [25] H. Yarmand and D. Craft, "Two effective heuristics for beam angle optimization in radiation therapy," *Medical Physics*, 2013, <https://arxiv.org/abs/1305.4959>.
- [26] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

## Research Article

# Optimum Assembly Sequence Planning System Using Discrete Artificial Bee Colony Algorithm

Özkan Özmen <sup>1</sup>, Turgay Batbat <sup>2</sup>, Tolgan Özen <sup>3</sup>,  
Cem Sinanoğlu <sup>1</sup> and Ayşegül Güven <sup>2</sup>

<sup>1</sup>Department of Industrial Design Engineering, Erciyes University, Kayseri 38030, Turkey

<sup>2</sup>Department of Biomedical Engineering, Erciyes University, Kayseri 38030, Turkey

<sup>3</sup>Software's Functional Management Branch Office, Turkish Air Force, Ankara 06580, Turkey

Correspondence should be addressed to Turgay Batbat; [turgaybatbat@erciyes.edu.tr](mailto:turgaybatbat@erciyes.edu.tr)

Received 20 November 2017; Revised 13 February 2018; Accepted 6 March 2018; Published 12 April 2018

Academic Editor: Abhishek K. Gupta

Copyright © 2018 Özkan Özmen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Assembly refers both to the process of combining parts to create a structure and to the product resulting therefrom. The complexity of this process increases with the number of pieces in the assembly. This paper presents the assembly planning system design (APSD) program, a computer program developed based on a matrix-based approach and the discrete artificial bee colony (DABC) algorithm, which determines the optimum assembly sequence among numerous feasible assembly sequences (FAS). Specifically, the assembly sequences of three-dimensional (3D) parts prepared in the computer-aided design (CAD) software AutoCAD are first coded using the matrix-based methodology and the resulting FAS are assessed and the optimum assembly sequence is selected according to the assembly time optimisation criterion using DABC. The results of comparison of the performance of the proposed method with other methods proposed in the literature verify its superiority in finding the sequence with the lowest overall time. Further, examination of the results of application of APSD to assemblies consisting of parts in different numbers and shapes shows that it can select the optimum sequence from among hundreds of FAS.

## 1. Introduction

With increasing international competition, efficient methods for manufacturing and producing goods at low cost are essential. This is particularly important in the manufacturing sector, where products are designed, assembled, and simulated with the aid of computers before they are produced. Assembly, the process of combining parts to create a structure or the result of this operation, is also the name given to the product. Assembly sequence planning (ASP) is a very complicated procedure which becomes more difficult as the number of parts increases, as this results in more complexities. These complexities in assembly processes necessitate that assembly sequences be well planned and thoroughly scrutinized. Hence, research is actively being conducted in this field with the objective of developing an ideal method for ASP [1–3].

In this paper, assembly planning system design (APSD), a computer program developed based on a previously

proposed matrix-based approach and the discrete artificial bee colony (DABC) algorithm, is proposed for automatically generating assembly sequences and determining the optimum sequence without user intervention. The program runs on the computer-aided design (CAD) software AutoCAD using the visual basics for applications (VBA) module and a CAD database. APSD can also work with data from other 3D CAD software programs by first converting it from the 3D CAD data of those software into dwg or ACIS data format. With the capability of counting, identifying, and labelling the components of a 3D model, APSD can automatically determine the relations between parts in terms of contact and translational functions. In the software, these relations are represented in matrix form and used to evaluate assembly sequences for automatic sequence generation and determination of whether connected parts can form a feasible assembly. Then, according to the assembly time criterion, an optimum assembly sequence is selected from the feasible assembly

sequences (FAS). DABC algorithm is used for optimisation and compared with the literature to evaluate the performance.

## 2. Literature Review

Initially, products sequencing planning in assembly lines was performed manually. However, with the development of related software and solid modelling programs, the first solid models of assembly products were developed in CAD programs. Subsequently, by means of various theorems, their assembly sequencing was determined. Finally, the optimum ordering was investigated.

The idea of combining assembly modelling with the 3D model of a product using a feature-based method was first introduced by Eng et al. [4]. Their primary approach involved using disassembly to determine the assembly. More specifically, they used the degree of freedom between two features of connected parts to distinguish kinematic conditions, boundary box control to determine collisions, and user interaction to set the restrictions on precedence relations. In another feature-based study, Zha and Du [5] used STEP standards to show modelling information and to manage and convert that information into assembly data. They proposed an assembly editor that could differentiate all feature connections and generate assembly sequences based on feature identification techniques. However, in their system, information on the location and liaison relations between parts must be detected for the automatic generation of the assembly sequences. This was for the purpose of moving parts along the axes of the Cartesian coordinate system in a CAD environment in order to diagnose the intersections between the parts and generate the assembly sequences. This idea was also utilised by Gottipolu and Ghosh [6] to detect the relational information between parts based on a geometric model of the assembly. In their approach, for every couple, the contact relations and disassembly orientations between parts along the coordinate axes are described in terms of two functions: contact and translational. The feasibility of the assemblies are then examined under two compulsory conditions with the implementation of logical operations: connectivity and precedence constraints. Then, after assessing all the numerous feasible sequences, the optimum sequences are shown as a table of assembly states and assembly tasks in a hierarchical fashion, starting from individual parts in the unassembled state to the completed assembly. This assembly sequence table is also supplied with revised features to use strategic constraints under several quantitative and qualitative criteria for analysis of suitable sequences and to determine the final sequence [7].

Lee and Kumara [8] proposed an approach for analysis and efficient production of the assembly sequencing of a complex assembly design before its schematic design stage. In their approach, every part is disassembled and recorded in a sweeping table. Then, assembly and disassembly sequences are produced using matrices and sweeping tables. Dini and Santochi [9] proposed a method based on a mathematical model of the product obtained by determination of three matrices (interaction, contact, and connection). For each

subassembly and product, all the possible assembly sequences are produced and the matrix numbers reduced depending on specified optimisation criteria. Zhang et al. [10] developed a procedure for automatic production of all appropriate assembly sequences for the assembly of the body of a car. Their developed procedure is based on the mathematical model of the car body obtained through its connecting and matching matrices, which represent the precedence constraints between the components and the subassembly. Possible subassemblies are then automatically determined on the basis of whether they satisfy specified mathematical conditions. Cizak [11] proposed a new concept based on graph theory and heuristic multipurpose optimisation that utilises two kinds of matrices (collision and assembly) to generate assembly sequences. Wu et al. [12] proposed an approach that employs assembly knowledge to ASP problems and presented an appropriate method for articulating geometric information and nongeometric knowledge. In their proposed approach, an assembly connection graph is built based on knowledge in the engineering, design, and manufacturing areas. Complicated and low-performance calculations in the assembly design process are bypassed in their approach, which they demonstrated via an assembly planning example.

Depending on the geometry and number of parts in an assembly, hundreds or even thousands of FAS can exist. Determining the optimum assembly sequence from among these FAS is a major issue. Various researchers are currently actively searching for solutions to this problem using artificial intelligence algorithms. For example, Guo et al. [13] proposed an approach based on the shuffled frog leaping algorithm to optimise the ASP of maintenance activities in radioactive environments. In their proposed approach, the geometrical feasibility of the assembly sequences is tested with the help of interference matrices along six axes. Further, to evaluate the fitness function, assembly directional and gripper changes are made. Experimental results indicated that their proposed method yielded better results than approaches based on classical genetic algorithm and particle swarm optimisation. Motavalli and Islam [14] proposed a multicriteria algorithm to find the best assembly sequence among all FAS based on simulated annealing (SA). Their proposed multicriteria algorithm considers both the assembly time and reorientation to find the optimum sequence. Subsequently, Choi et al. [15] applied genetic algorithm (GA) to the problem and obtained better results than using SA. Further, Karthik and Deb [16] compared GA and an hybrid cuckoo-search genetic algorithm (CS-GA) and found that they resulted in the same assembly time. In addition, Mukred et al. [17] applied the particle swarm algorithm (PSA) and the binary particle swarm algorithm (BPSA) to the same problem and found that PSA produced the optimised solution with the shorter assembly time.

## 3. Mathematical Model

In this section, our proposed mathematical model is explained from three aspects: (1) minimisation of assembly time, (2) reorientation, and (3) matrix-based assembly

sequencing [15, 18]. Minimum assembly time and cost are achieved by considering factors that play a major role, such as the setup time, transfer time, number of tool changes, and proper fixture selection. Therefore, to evaluate the performance of DABC, the following two criteria are discussed: (1) minimisation of assembly time, including setup and actual assembly; (2) minimisation of the number of reorientations to satisfy geometrical constraints, as the parts or subassemblies to be assembled could differ in terms of geometry. When the DABC algorithm is considered in APSD, only assembly time is considered.

**3.1. Minimisation of Assembly Time.** Assembly time consists of setup time and actual assembly time, which is assumed to be always constant regardless of the sequence. Every component of an assembly demands proper setup. The setup time depends on the geometry of the component itself and the components assembled previously. The following function can be used to predict the setup time for a component:

$$T_{\text{Setup}}(i) = p_{i0} + \sum_{j=1}^n p_{ij}x_{ij}, \quad (1)$$

where  $i$  is component to be assembled,  $p_{i0}$  is setup time for product  $i$  being the first component in the assembly,  $p_{ij}$  is

contribution to the setup time due to the presence of part  $j$  when entering part  $i$ ,  $x_{ij} = \{1, \text{ if component } j \text{ has already assembled, for } i = 1, \dots, n; 0, \text{ otherwise, for } i = 1, \dots, n\}$

The total assembly time is the summation of the setup time and actual assembly time:

$$\min T_{\text{Assembly}} = \sum_{i=1}^n (T_{\text{Setup}}(i) + A_i), \quad (2)$$

where  $A_i$  is the assembly time for component  $i$ . The objective function for minimising the assembly time is as follows:

$$Z_1 = \min T_{\text{Assembly}}. \quad (3)$$

**3.2. Reorientations and Combined Objective Function.** There are certain geometric relations between each part in an assemble. Some intermediate assembly may need reorientation in order to get the parts assembled in a particular sequence. The target here is to minimise the number of reorientations. The total number of reorientations necessary during the assembly of a part is a function of the geometry of the part and the subassembly. Reorientation is defined as follows:

$$R_i = \begin{cases} 1, & \text{reorientation is needed when entering component } i, \text{ for } i = 1, \dots, n \\ 0, & \text{otherwise, for } i = 1, \dots, n. \end{cases} \quad (4)$$

The total number of reorientations is given by

$$R = \sum_{j=1}^n R_j. \quad (5)$$

Thus,  $Z_2 = \min R$ .

A multicriteria utility function is used to describe two combined objective functions ( $Z_1$  and  $Z_2$ ). Thus, the combined objective function (COF) is as follows:

$$\min Z = w_1 \times Z_1 + w_2 \times Z_2, \quad (6)$$

where  $Z$  is the combined objective function,  $w_i$  is the weight of the individual functions, and  $i$  is 1 or 2. The weight of the function depends on the strategy followed and expert opinion. In this study,  $w_1 = 0.9$  and  $w_2 = 0.1$  were used to evaluate the performance of the DABC algorithm. Further, only the  $Z_1$  function was considered in the APSD program.

**3.3. The Matrix-Based ASP Method.** The proposed approach for the determination of assembly sequences begins with creation of a CAD assembly model. The assembled parts generated in the CAD medium are sufficient for geometric knowledge but not for assembly planning. The data pertaining to the parts created in the CAD environment are used

as input to the assembly planning system. Then, a matrix-based mathematical model is formed and, finally, its assembly sequence is determined by means of Boolean algebra [6, 18]. The theorem is explained below based on the wheel of a shopping cart, as exemplified in Figure 1.

The contacts between two components,  $a$  and  $b$ , are represented by the  $1 \times 6$  binary function  $C_{ab} = (C_1, C_2, C_3, C_4, C_5, C_6)$ . Six elements are used to represent the  $+X, +Y, +Z, -X, -Y$ , and  $-Z$  directions of the Cartesian coordinates. A matrix called the contact function matrix is then defined as follows:  $C_i = 1$  signifies contact in the direction  $i$ ;  $C_i = 0$  signifies no contact between  $a$  and  $b$ . The translational motion between parts  $a$  and  $b$  is represented by a  $1 \times 6$  binary function called the function of translation, which is defined by  $T_{ab} = (T_1, T_2, T_3, T_4, T_5, T_6)$ . In this function,  $T_i = 1$  signifies that part  $b$  can move freely in the direction of  $i$  without collision with part  $a$ ;  $T_i = 0$  signifies either that part  $b$  hits part  $a$  in the direction of  $i$  or part  $a$  prevents part  $b$  from moving freely.

As can be seen in Figure 1, all the values in the  $P_1P_2$  line of the contact matrix have a value of zero, as there is no contact in the six axes between parts  $P_1$  and  $P_2$ . The values for the contact and translational functions of the wheel of a shopping cart comprising four parts are presented in Table 1. Because parts  $P_1P_2$  do not prevent each other's movement, a value

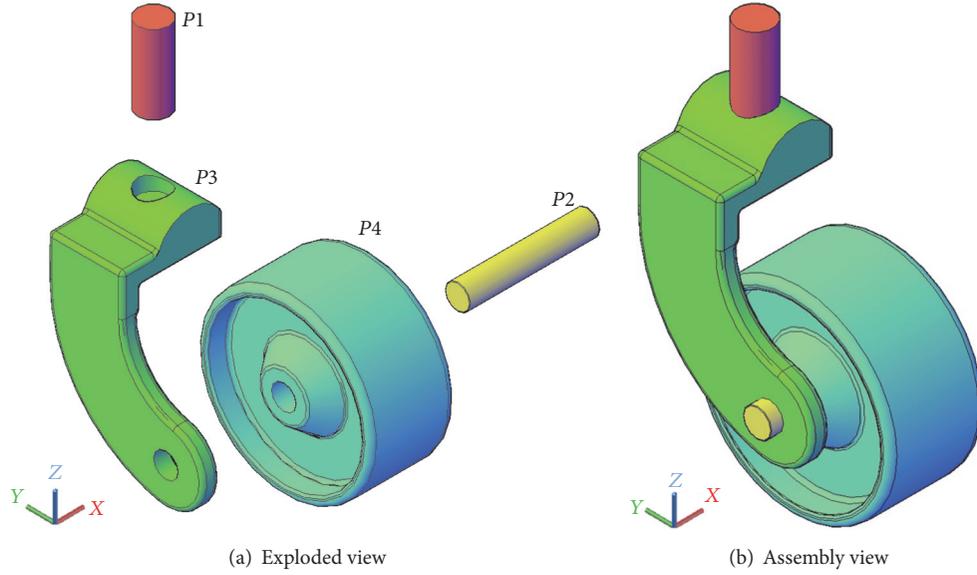


FIGURE 1: Three-dimensional view of shopping cart wheels.

TABLE 1: The contact and translational function of the wheel of a shopping cart.

Pair	$C_{+x}$	$C_{+y}$	$C_{+z}$	$C_{-x}$	$C_{-y}$	$C_{-z}$	$T_{+x}$	$T_{+y}$	$T_{+z}$	$T_{-x}$	$T_{-y}$	$T_{-z}$
$P_1P_2$	0	0	0	0	0	0	1	1	1	1	1	1
$P_1P_3$	1	1	0	1	1	1	0	0	1	0	0	0
$P_1P_4$	0	0	0	0	0	0	1	1	1	1	1	0
$P_2P_1$	0	0	0	0	0	0	1	1	1	1	1	1
$P_2P_3$	0	1	1	0	1	1	1	0	0	1	0	0
$P_2P_4$	0	1	1	0	1	1	1	0	0	1	0	0
$P_3P_1$	1	1	1	1	1	0	0	0	0	0	0	1
$P_3P_2$	0	1	1	0	1	1	1	0	0	1	0	0
$P_3P_4$	0	0	0	0	0	0	0	1	1	1	1	0
$P_4P_1$	0	0	0	0	0	0	1	1	0	1	1	1
$P_4P_2$	0	1	1	0	1	1	1	0	0	1	0	0
$P_4P_3$	0	0	0	0	0	0	1	1	0	0	1	1

TABLE 2: For the assembly  $P_1P_3P_2$ , contact and translational functions relation of  $P_1P_2$  and  $P_1P_3$  pairs.

Pair	$C_{+x}$	$C_{+y}$	$C_{+z}$	$C_{-x}$	$C_{-y}$	$C_{-z}$	Pair	$T_{+x}$	$T_{+y}$	$T_{+z}$	$T_{-x}$	$T_{-y}$	$T_{-z}$
$P_1P_2$	0	0	0	0	0	0	$P_1P_2$	1	1	1	1	1	1
$P_3P_2$	0	1	1	0	1	1	$P_3P_2$	1	0	0	1	0	0
TC	0	1	1	0	1	1	TT	1	0	0	1	0	0

of “1” is assigned to all the variables in the first line of the matrix.

After determining the contact ( $C$ ) and the translational ( $T$ ) function, to determine the feasibility of the assembly of this couple, the total contact (TC), total translation (TT), total contact result (TCR), and total translation result (TTR) values are calculated. Two types of limiters (connection and priority) have to be considered following procurement of the contact and translational functions for the FAS of the parts to be assembled. These parts depend on the other parts for assembly to be learned from the connection limiter. The

priority limiter states the parts that have to be assembled in advance. Not assembling these parts in the correct sequence will prevent assembly of subsequent parts. These limitations are verified through application of the principles of Boolean algebra to the matrix elements of  $C$  and  $T$  functions (Table 2).

$TC_i = C(P_1P_2) \vee C(P_3P_2)$ , where  $i$  denotes that the relevant columns in each of axes 1 to 6 are subject to the “OR” operation. Finally, TCR is obtained.

This does not include

$$TCR = TC_1 \vee TC_2 \vee TC_3 \vee TC_4 \vee TC_5 \vee TC_6. \quad (7)$$

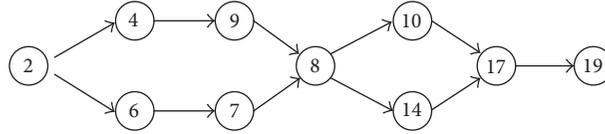


FIGURE 2: The assembly precedence diagram for performance of DABC [15].

TABLE 3: Feasible assembly sequences of wheel of a shopping cart.

Feasible assembly sequences	
$P_1-P_3-P_2-P_4$	$P_3-P_1-P_2-P_4$
$P_2-P_3-P_1-P_4$	$P_3-P_2-P_1-P_4$
$P_2-P_3-P_4-P_1$	$P_3-P_2-P_4-P_1$
$P_2-P_4-P_3-P_1$	$P_4-P_2-P_3-P_1$

A value of  $TCR = 1$  signifies that connections exist between the parts. In the example given  $TCR = 1$ , which means that part  $P_2$  is in contact with the set of subassembly parts  $P_1P_3$ . Contact is necessary for the parts to be assembled, but contact alone is not sufficient. In order to obtain priority limiters, an accuracy table is elicited from the motion prevention functions as in the connection limitations. Next, priority limiters are obtained by subjecting the parts to a logical AND ( $\wedge$ ) operation. The results are then subjected to a logical OR ( $\vee$ ) operation to obtain TTR.

$$TTR = TT_1 \vee TT_2 \vee TT_3 \vee TT_4 \vee TT_5 \vee TT_6. \quad (8)$$

A value of  $TTR = 1$  that there is freedom of movement between the parts in at least one axis, whereas  $TTR = 0$  signifies lack of freedom of movement in all axes. In order for one part to be assembled with another set of parts, it must meet the suitability criteria. The feasibility of assembly (FA) is obtained by subjecting the total movement prevention outcome and the total contact outcomes to a logical AND operation. If the result of this operation is one, then assembly of the part is possible; otherwise, it is impossible to assemble. In the example assembly presented,

$$FA = TCR \wedge TTR = 1 \wedge 1 = 1. \quad (9)$$

This means that it is possible to assemble part  $P_2$  and subset parts  $P_1P_3$ . In conclusion, suitable  $P_1P_3P_2$  subset assembly parts can be obtained. After this stage, the same theorem can be used to determine whether part  $P_4$  can be assembled with the subassembly of  $P_1P_3P_2$ . Thus, FAS can be determined by applying the theorem to all the dual parts. FAS for the given parts are presented in Table 3.

#### 4. Artificial Bee Colony (ABC) Algorithm for ASP

Artificial intelligence (AI) is often used to solve problems that do not precisely match mathematical models or to decrease industrial costs. Even low-level computers are capable of solving difficult problems in a relatively short time with the

help of AI. Due to the necessity of rapid manufacturing of products in the internationally competitive environment, the assembly sequence is very important in terms of cost [1]. Therefore, it is only natural that a whole new field that focuses especially on ASP optimisation has been developed.

One of the most effective AI models is the ABC algorithm, proposed by Karaboga [19], which models the behaviour of bees in search of food. As the values in the ASP optimisation problem are not continuous, discrete optimisation has to be used to solve the problem. The neighbouring solution creation strategy necessitates new approaches when the problem is recognised as discrete optimisation. Seeking solutions for discrete optimisation problems with the help of the ABC algorithm is an area of interest that is constantly increasing. One such discrete optimisation problem is the workplace flow scheduling problem, for which models for dynamic clustering processes in different types of datasets have been presented [20–25]. In addition, next-generation problems such as cloud service composition, DNA sequencing, and DNA three-dimensional structure prediction problems are present in the literature [26–28].

*4.1. Description and Initialisation of the ABC Algorithm for ASP.* In the ABC algorithm, bees' gathering nectar correspond to the feasible solution of the problem, while the quality of the nectar in the source corresponds to the fitness value. In our study, the feasible solutions of the problem correspond to the sequence of the parts to be assembled. Further, the fitness values are in the form of the values scaled with (10), calculated in (6) on the basis of the assembly time and knowledge of the reorientation, if any.

$$\text{fitness}_i = \begin{cases} \frac{1}{1 + f_i}, & f_i \geq 0 \\ 1 + |(f_i)|, & f_i < 0. \end{cases} \quad (10)$$

Half of the initially formed population is regarded as employed and the other half as onlooker bees. While initial solutions are being formulated, a random source is determined, as defined in (11), in the whole solution range for each employed bee.  $\text{Rand}(0, 1)$  in the equation denotes a value varying between zero and one. The fitness value of the source visited by the employed bees can be obtained by means of a problem-related function replacing it with  $f_i$  in (10). To assess the performance of DABC, the priority graph (Figure 2) composed of 19 parts in the reference article and the priority matrix obtained from them are used for random sequences initial solutions, and FAS values (which will be

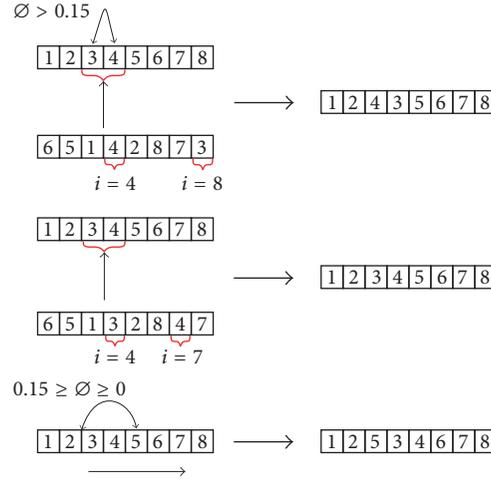


FIGURE 3: Proposed neighbouring solutions generation mechanism.

explained in the next section) are used for initial solutions in the assessment of APSD.

$$x_{ij} = x_j^{\min} + \text{rand}(0, 1)(x_j^{\max} - x_j^{\min}). \quad (11)$$

$i = 1 \cdots SN$  source,  $j = 1 \cdots D$ , the employed bee assigned to turn to the source,  $SN$  is total number of sources,  $D$  is total number of parameters,  $x_j^{\max}$  is upper limit of parameter  $j$ , and  $x_j^{\min}$  is lower limit of parameter  $j$

**4.2. Employed Bee Phase.** The bees assigned to make honey are divided into three different categories in the algorithm formulated: scout bees assigned to locate utterly new sources; employed bees bringing nectar to the hive from the newly discovered source and nearby sources as well as describing the location and quality of the source to other bees waiting in the hive; and onlooker bees waiting in the hive to go to the sources they determine according to the descriptions they have received of the location of the source. Honey bees associate the location of sources with the position of the sun and impart this to the other bees, together with the knowledge about the nectar, by means of their dances in the hive. During these activities, employed bees continue their search of other sources nearby. As is seen in (12), on their way to a newly discovered source, they make use of either the source they are in or one of the sources from which other employed bees collect nectar.

$$v_{ij} = x_{ij} + \emptyset_{ij}(x_{ij} - x_{kj}). \quad (12)$$

The new source is derived by changing the parameter from the source, where  $\vec{x}_i$  denotes the change from  $\vec{v}_i$  and  $\emptyset_{ij}$  is a randomly selected number in the range  $[-1, 1]$ . However, because the values found are discrete, the newly obtained solution will recur in all stages in the manner presented below. In this random process, the source  $k$ , which is one of the sources selected randomly from among other sources, is used. Thus, as the sources approach each other, the size of the

steps will decrease, resulting in a more intensive search. The solutions exceeding the limits are fixed to the limits, as in

$$v_{ij} = \begin{cases} x_j^{\min}, & v_{ij} < x_j^{\min} \\ v_{ij}, & x_j^{\min} \leq v_{ij} \leq x_j^{\max} \\ x_j^{\max}, & v_{ij} > x_j^{\max}. \end{cases} \quad (13)$$

**4.3. Generation of Neighbouring Solutions.** As the values are discrete, new solutions cannot be realised with numerical changes. Equation (11) is not used directly to produce neighbouring solutions. However, as was noted previously, it is necessary to take advantage of the algorithm in the process of obtaining neighbouring solutions, sticking to the foundations of the algorithm. Inspired by the discrete optimisation approaches in the literature, the method shown in Figure 3 was devised.

In this strategy, the  $\emptyset_{ij}$  component used by the ABC algorithm represents the random selection of the function which yields the new solution from the former one taken suitably for its purpose. This process is achieved as follows:

- (i) A randomly selected different solution ( $\vec{x}_k$ ) is used for 85% of the neighbouring solutions. The components determined in terms of random numbers not exceeding 30% of the solution dimension are selected from  $\vec{x}_i$ , beginning from the random position. The neighbouring solution is generated by sequencing the parts in this section and arranging their location in other solutions. Thus, the tendency to turn to a different solution is realised.
- (ii) In the remaining solutions, a randomly selected solution is brought to its previous location to reduce the rate of assimilation, and the components between the replaced component and its own location are shifted to the next location. This process is called insert.

Thus, while new solutions are being obtained by means of different solutions, early convergence is prevented with

random solutions. Addressing only the solutions appropriate for the existing conditions when generating neighbouring solutions would give rise to early convergence and result in the algorithm becoming stuck in local minima. Penalising the fitness value of unfit solutions with certain coefficients not only raises the probability of unfit solutions, if any, but also eliminates the possibility of getting stuck in local minima.

**4.4. Onlooker Bee Phase.** Onlooker bees waiting in the hive act on the information supplied by employed bees. Equipped with this information, they choose one of the sources and go to that source to collect nectar. Which source the onlooker bees will go to is determined in accordance with the feasibility value obtained with the ratio of the fitness value of that source to the total value of all the sources from which all the employed bees collect nectar. The feasibility value is calculated using (14). The onlooker bees going to the randomly selected source in view of this feasibility value enable the local search to intensify around the sources with greater quality, and they take nectar from the neighbouring sources located with the neighbouring solutions generation mechanism.

$$P_i = \frac{\text{fitness}_i}{\sum_{j=1}^{SN} \text{fitness}_j}. \quad (14)$$

**4.5. Scout Bee Phase.** In the ABC algorithm, sources with depleted nectar are abandoned by the honey bees, thereby preventing cessation of the algorithm. In this process, employed bees resume searching for new sources while onlooker bees return to the hive. The failure counter, which determines when each source is deemed depleted, checks whether the limit parameter defined before each iteration is exceeded. Failure<sub>*i*</sub> starts as zero when initial solutions are generated. In both the employed and onlooker bee phases, if the former value of the source is replaced with the value produced in the phase, it is reset to zero. Otherwise, the value on the counter is increased by one. In each generation, only one source can remain. Usually, scout bees determine a source using the same method as their initial solutions. However, in our study, a different method is employed as the development of discrete values incurs significant computation time. This method is realised, without comparing fitness value, with the insert operation explained in the generating neighbouring solutions section above, and is repeated up to three times until similarity is sufficiently reduced. Each time these three phases are completed, the favourable solution in the population is compared with the previous favourable solution and the superior one is selected. The general DABC algorithm proposed is presented in Figure 4.

## 5. Assembly Planning System Design (APSD)

As more than one assembly sequence of the products is usually used in industry, and only one of these sequences is chosen by various criteria for the realisation of assembly following acquisition of all the assembly sequences, these

criteria are determined by an expert assembly sequence planner for any product, and only one of them is sufficient to complete the assembly planning.

In this study, the software APSD was devised in AutoCAD using the algorithms explained above and the VBA module. APSD consists of four main tabs: (1) Parts List, (2) *C-T* Functions, (3) Feasible Assembly Sequences, and (4) Assembly Tree. These tabs help users carry out the four main procedures shown in Figure 5, specifically, identification of parts, determination of *C* and *T* functions, generation of all feasible assemblies, and displaying of the assembly tree.

In the software, FAS are first found and then the optimisation algorithms are executed. First, the contact and translational functions are described as arrays. A procedure called the assembly test function (ATF) was coded to achieve these calculations and to test the feasibility of the sequences using the matrix-based assembly sequence methodology explained in the previous section. With the implementation of recursive programming, a procedure called the proper part finding procedure (PPFP) was coded to accomplish part selection from the parts list before the feasibility of the assembly is checked. The flowchart of this algorithm is shown in Figure 6 [29].

In this section, determination and optimisation of the FAS of the connection frame assembly system shown in Figure 7 are realised with the software developed.

When the “Identification of Parts” button is pressed, the program automatically names the parts (*P1-P2-P3-P4-P5-P6-P7-P8-P9*). Figure 5 shows the interface of the program. In addition, the names assigned by AutoCAD are shown next to the part names. As seen in the figure, there are nine parts on the connection frame assembly. To determine all the feasible sequences of the parts, first contact and translational matrices have to be found. This is achieved by pressing the “Calculate the *C-T* Functions” button in the interface of the program. While moving each part along the six axes to determine its interaction with the other parts, the program prints the matrix in a section of the *C-T* Function window. As there are nine parts, as is seen in Figure 8, in the connection frame (9!) 362880 sequences are possible. However, not all of these sequences are significant for assembly. To obtain the feasible sequences in the assembly it is necessary to press the “Feasible Assembly Sequences” button. This returns all the assembly sequences obtained using the matrix-based assembly sequence theorem presented in the previous section. Thus, 4530 FAS are presented, as shown in Figure 8.

As shown in the figure, 4530 FAS are obtained in the assembly system. However, the designer cannot possibly assess all these sequences one by one. Therefore, the found FAS are transferred to a text file by means of “Write Results to Text File” function and input into MATLAB. Unlike the assessment of the performance of the DABC algorithm, FAS are taken as entry in place of the priority matrix. Next, optimisation is realised from FAS in view of the assembly time, for which Table 5 is used according to the number of components. For example, for a nine-component assembly, the 9 × 9 section of the table is used.

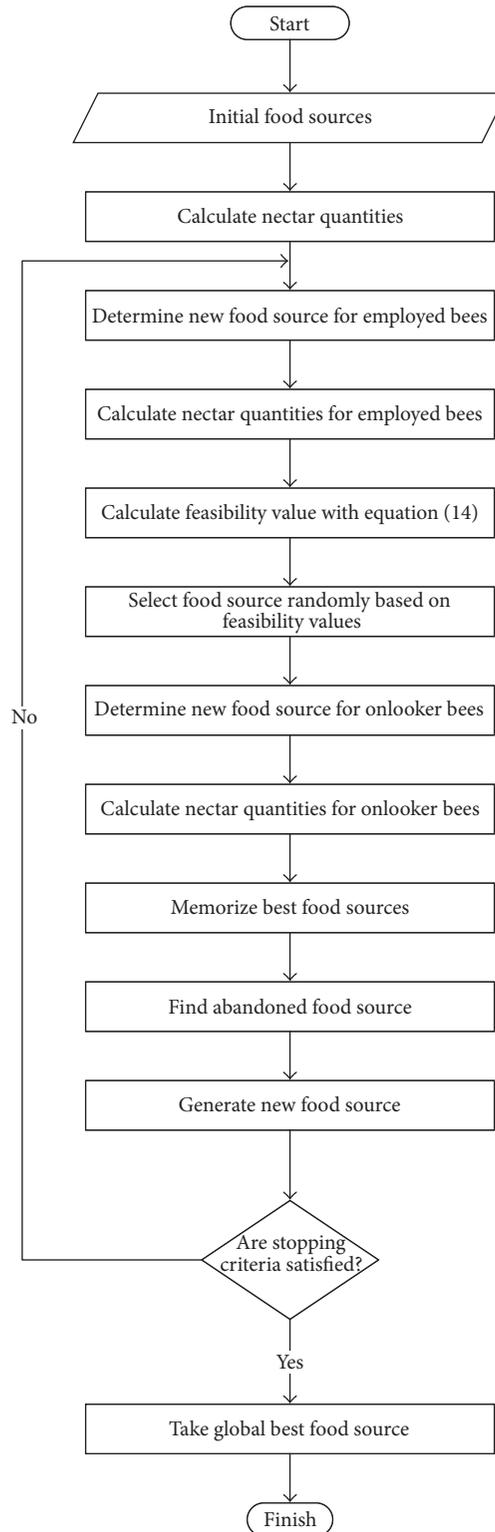


FIGURE 4: Flowchart of the proposed discrete artificial bee colony (DABC) algorithm.

## 6. Results and Discussion

The program code of the DABC algorithm was developed in MATLAB and executed on a desktop computer with an i7-4790 3.60 GHz CPU, 16 GB RAM, and 1 TB memory. The

parameters used for the DABC algorithm for the purpose of comparison with other studies were as follows: limit value, 100; colony size (NP) values, 20, 40, 100; and number of generations, 100. For the solutions applied in APSD, the size of the population was doubled; the population was 10 times

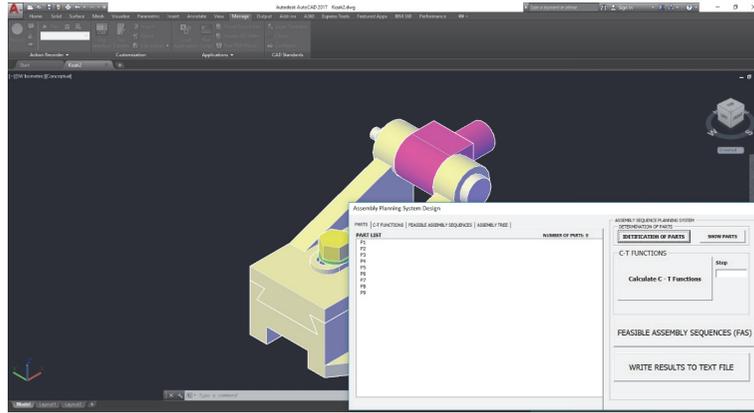


FIGURE 5: Program interface is shown in AutoCAD visual basic for application module.

TABLE 4: Parameters of optimisation techniques.

	DABC	SA	CS-GA	GA
Number of iteration	100	500	1000	1000
Colony size	20, 40, 100	-	15	15
Limit	100	-	-	-
Run	10	-	100	100
Cross over rate	-	-	0.8	0.8
Mutation rate	-	-	0.2	0.2
Initial temperature	-	100	-	-
Cooling rate	-	0.95	-	-

the number of features; and the limit value was one-half the size of the population. Thus, efforts were made to enable the system to respond to assemblies involving different features. The selected parameters yielded values far below those in the literature. In addition, a penalty value of five was used for each parameter in the compared systems, but, for APSD, the number 50 was used only once to make it conform to the rules and optimum results obtained. The parameters in the article taken as reference to assess the performance of DABC are presented in Table 4. The assembly duration and reorientation of the parts used in the studies taken as reference and in the present study are presented in Tables 5 and 6, respectively. Further, Table 7 shows the assembly duration of the sequences in the reference articles according to (6), compared with the assembly duration obtained from DABC.

As the population size suggested in the DABC algorithm has been run 10 times and the minimum value of results has been taken into consideration. However, in each of the ten operations, the minimum value was obtained at least once (Figure 9).

The assembly with nine components in Figure 7 was tested in APSD and 4530 feasible solutions were obtained. Following optimisation in DABC, a total assembly time of 173.800 and the assembly sequence 1-9-8-5-4-3-6-2-7 were obtained.

## 7. Conclusion

In this study, a program for finding the optimum sequence among the FAS of an assembly system designed with computer assistance as a 3D solid object based on assembly time using DABC was developed. The program automatically realises the background within the framework of algorithms determined at every step by minimising utilisation and interaction. Examination of the results obtained reveals that the optimum sequence was selected from among hundreds of available assemble sequences, indicating that the program is effective. The program has been tested on assembly systems consisting of parts of different numbers and shapes. In addition, the performance of the DABC algorithm was compared with the results of other methods proposed in the literature with superior results being obtained.

All the possible assembly sequences modelled as 3D solid in AutoCAD were generated automatically. Further, the developed program produced consistent results over numerous runs. Hundreds of FAS can be generated even in cases in which the number of parts in an assembly is small. In the developed software, selection is made from among the available sequences according to predetermined criteria. Despite these results, the method we have proposed has one constraint: the efficacy of the method suggested to obtain assembly sequence is predicated on 3D CAD data

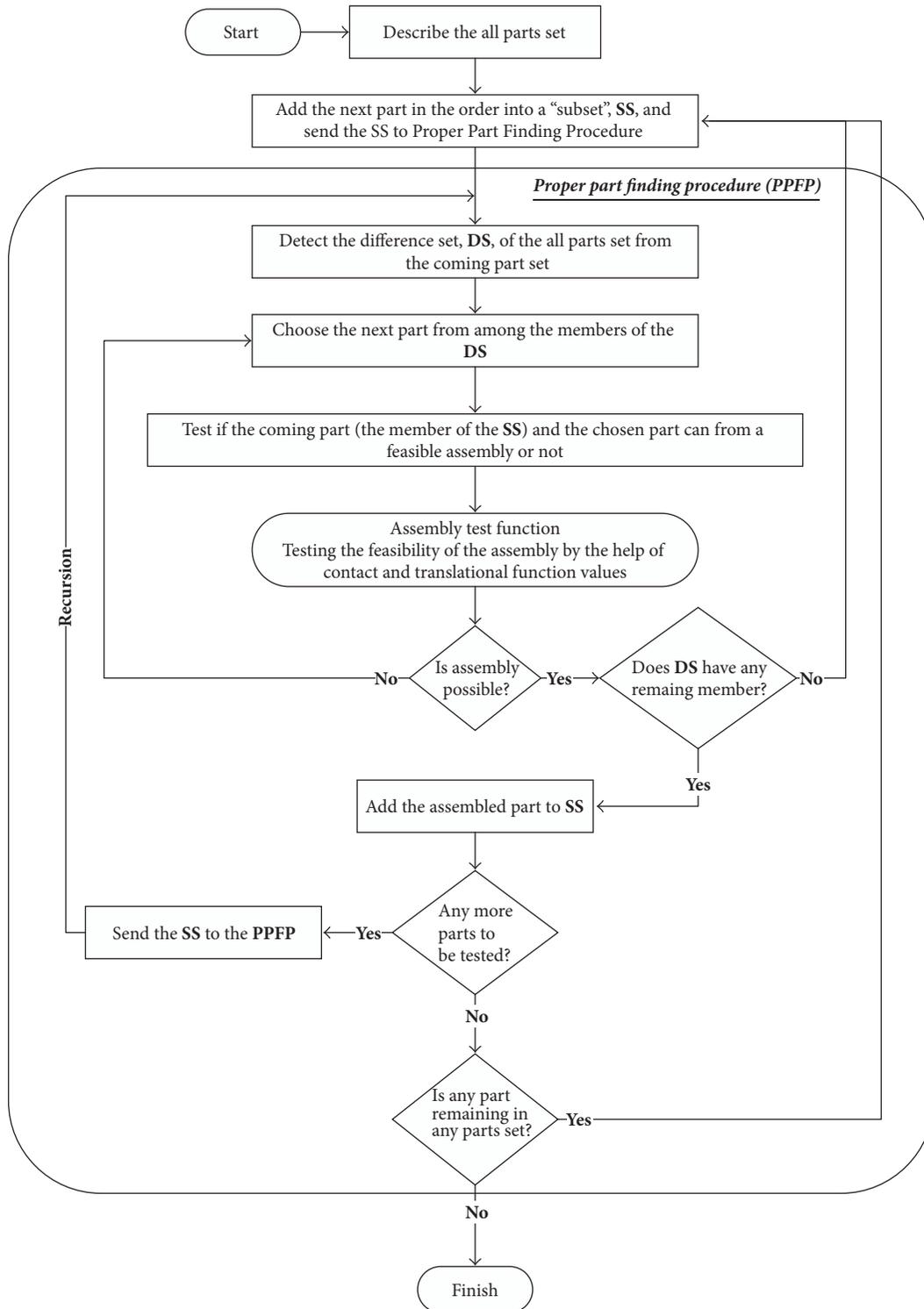


FIGURE 6: Recursive procedure for testing the feasibility of the parts to be assembled.

fitting in with Cartesian coordinates. On the other hand, the development of algorithm suggested for curved components is one of the topics of future studies we plan to do to assess existing assembly systems by considering both mechanical

criteria such as weight and degree of subassembly freedom and by using real assembly durations together with expert assembly sequence planners and enlarging the scope of optimisation with the proposed algorithm.

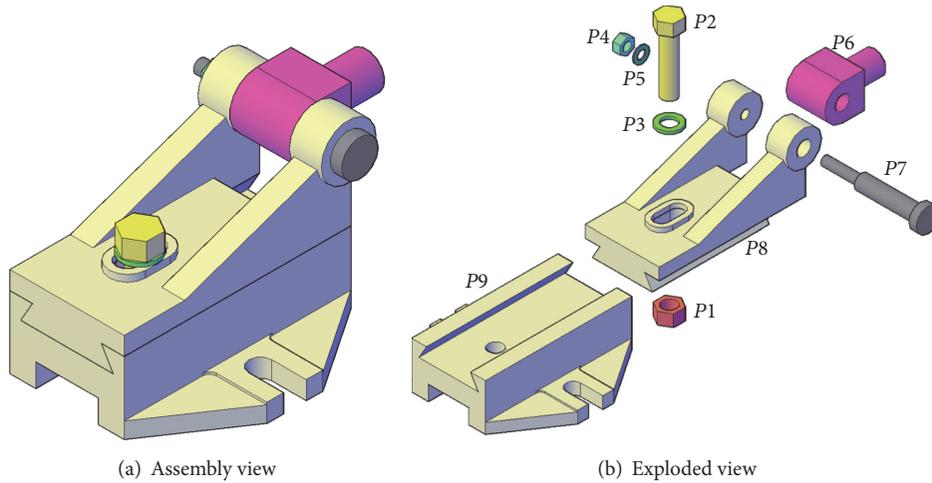


FIGURE 7: Three-dimensional connection frames in AutoCAD.

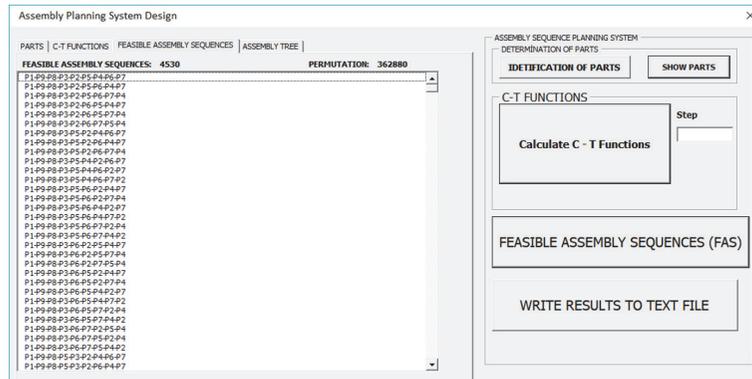


FIGURE 8: The program interface of the tab of FAS.

**Nomenclature**

- DABC: Discrete artificial bee colony
- 3D: 3-dimensional
- CAD: Computer-aided design
- APSD: Assembly planning system design
- FAS: Feasible assembly sequences
- ASP: Assembly sequence planning
- VBA: Visual basic for application module
- SA: Simulated annealing
- GA: Genetic algorithm
- CS-GA: Cuckoo-search genetic algorithm
- BPSA: Binary particle swarm algorithm
- PSA: Particle swarm algorithm
- COF: Combined objective function
- C: Contact
- T: Translational
- TC: Total contact
- TT: Total translation
- TCR: Total contact result

- TTR: Total translation result
- FA: Feasibility of assembling
- AI: Artificial intelligence
- ABC: Artificial bee colony
- ATF: Assembly test function
- PPFP: Proper part finding procedure
- SS: Subset
- DS: Difference set.

**Conflicts of Interest**

The authors declare that there are no conflicts of interest regarding the publication of this paper.

**Acknowledgments**

The author would like to acknowledge Editage for English language editing. Also this work was supported by Erciyes University Scientific Research Projects Coordination Unit with Project no. FBY-12-3987.

TABLE 5: Coefficient of various components in the assembly.

Component to be assembled	Component <i>I</i>																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	10	1	2	3	4	5	6	7	8	9	3.2	4.3	7	6.1	1.2	3.4	0	0	7.4
2	1.5	10	2	2	2	2	2	2	2	2	0	3.1	6	4.3	2.7	4.8	0	3	0.5
3	1	2.3	10	0	4	5	0	4	4	2.3	9.8	2.4	5	1.2	3.4	4.5	5.6	3.4	3.1
4	0	2	3.4	10	4.5	0	4	0	0	8	3.4	5.6	5	0	0	3.4	0	0	9.8
5	1.2	1	2	3	10	7.9	8.9	0	1.2	2	2.3	0	3	0	3.6	0	2.8	9.8	0
6	9.8	4.5	0	1.2	3.6	10	3.4	4	0	2.3	4.6	5.6	0	4	3	2	0	0.4	3.2
7	0.5	1.4	2.3	0.5	1.9	1	10	13.4	1.2	4	2.3	0	3	5.7	8.30	2	0.1	0	0.5
8	0	0	0	0	0	1.8	9.8	10	2.3	3	8.9	2.3	0	0	2.3	0.5	9.8	0	2.3
9	1	3	4.5	2.3	4.6	9.8	7.5	6.8	10	6	2.3	3.4	5	12.3	3.4	5.61	1	0	0
10	2.3	4.5	2.3	0	2.3	0	2.1	0	4.5	10	1.1	2.3	2	0	0	2.1	1.2	5.4	9.2
11	1	1	2	3	4	5	6	7	8	9	10	4.5	3	6.1	1.2	3.4	0.3	0	1.3
12	1.5	0	2	2	2	2	2	1	2	2	11.2	10	6	4.3	2.7	4.8	0	3	0.5
13	1	2.3	0	0	4	5	0	4	2.3	4.3	9.8	2.4	10	1.2	2.4	4.5	1.6	2.4	3.1
14	0	2	3.4	0	4.5	0	4	0	8	0	3.4	5.6	5	10	2.1	1.4	1	0	2.8
15	1.2	1	2	3	0	7.9	8.9	0	1.2	2	1.3	4	3	1.4	10	1.3	9.8	9.8	2
16	9.8	4.5	0	1.2	3.6	0	3.4	4	4	0	4.6	3.6	0	4	3	10	1.5	0	3.2
17	1	3	4	5	0	5	4	3.4	1.2	4	1.3	0	2	3.7	4.3	2.3	10	3.8	10
18	0.6	0.5	3.4	1.2	3	2	9.8	2	2.3	3	5.9	2.3	0	1.0	2.3	0.5	9.8	10	2.3
19	1	3	4.5	2.3	4.6	9.8	7.5	6.8	0	6	3.3	3	2	3.3	4.4	2.6	0.3	2.5	10

TABLE 6: Set for reorientations.

Part	Set for reorientation
1	{4, 9} {9, 11} {4, 6, 11} {11, 7} {7, 9} {6, 5} {13, 12, 18}
3	{5, 2} {1, 4, 5} {16, 18} {2, 1, 6} {2, 1, 4, 5} {13, 9, 4} {9, 6, 2} {6, 2, 10} {18, 17}
7	{5, 12, 6} {12, 6, 1} {9, 6, 16} {19, 2, 4, 9, 8} {12, 6} {7, 5} {9, 5} {10, 8, 6, 4}
8	{6, 7, 16} {9, 6, 5, 7} {12, 7} {15, 2} {16, 15, 5} {9, 5, 1}
10	{16, 8} {6, 7, 8} {15, 1} {7, 3} {9, 4} {19, 17, 3}
11	{4, 9, 16} {12, 13} {12, 5, 9} {13, 6, 16} {13, 12} {16, 18, 3}
16	{6, 7} {12, 6} {11, 9, 6} {5, 3, 18, 11, 13} {13, 12, 4, 19, 18, 6, 5, 3, 8}
17	{14, 15} {5, 2} {10, 14, 15} {13, 12, 9} {15, 13, 5} {19, 7}

TABLE 7: Comparison of optimisation techniques.

Opt. tech.	Assembly sequence	Assembly time	Reorientation
SA	2-1-4-9-3-12-13-16-5-15-18-6-11-7-8-10-14-17-19	528.6600	0
CS-GA*	2-1-4-9-3-12-13-16-5-15-18-11-6-7-8-10-14-17-19	528.4000	1 (6-7-8-10)
GA*	2-1-4-9-3-12-13-16-5-15-18-11-6-7-8-10-14-17-19	528.4000	1 (6-7-8-10)
DABC	2-1-4-9-3-12-13-16-5-15-18-11-6-7-8-14-10-17-19	528.3000	0

\*In these studies, sequencing starts with zero.

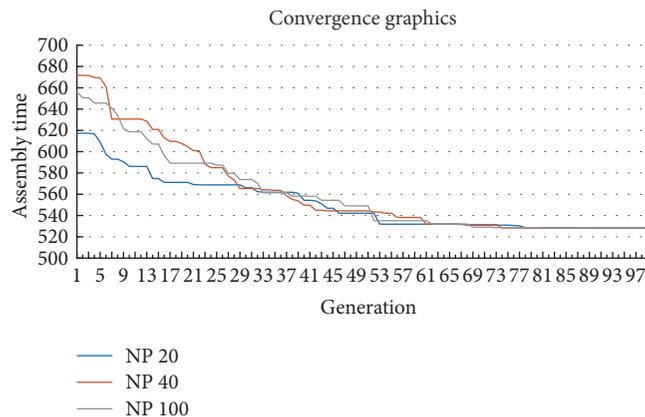


FIGURE 9: The result of DABC.

## References

- [1] C. Sinanoğlu and H. Rıza Börklü, "An assembly sequence-planning system for mechanical parts using neural network," *Assembly Automation*, vol. 25, no. 1, pp. 38–52, 2005.
- [2] H. Dilipak and A. Özdemir, "Generation of Assembly And Subassembly Sequences Of Industrial Products And The Optimization With Constraints," *Gazi University Journal of Science*, vol. 16, no. 4, pp. 797–810, 2003.
- [3] C. J. M. Heemskerk, L. N. Reijers, and H. J. J. Kals, "A Concept for Computer-Aided Process Planning of Flexible Assembly," *CIRP Annals - Manufacturing Technology*, vol. 39, no. 1, pp. 25–28, 1990.
- [4] T. H. Eng, Z. K. Ling, W. Olson, and C. McLean, "Feature-based assembly modeling and sequence generation," *Computers & Industrial Engineering*, vol. 36, no. 1, pp. 17–33, 1999.
- [5] X. F. Zha and H. Du, "A PDES/STEP-based model and system for concurrent integrated design and assembly planning," *Computer-Aided Design*, vol. 34, no. 14, pp. 1087–1110, 2002.
- [6] R. B. Gottipolu and K. Ghosh, "Representation and selection of assembly sequences in computer-aided assembly process planning," *International Journal of Production Research*, vol. 35, no. 12, pp. 3447–3466, 1997.
- [7] R. B. Gottipolu and K. Ghosh, "A simplified and efficient representation for evaluation and selection of assembly sequences," *Computers in Industry*, vol. 50, no. 3, pp. 251–264, 2003.
- [8] Y.-Q. Lee and S. R. Kumara, "A scheme for mechanical assembly design and assembly line layout conceptualization," *Computers & Industrial Engineering*, vol. 27, no. 1, pp. 261–264, 1994.
- [9] G. Dini and M. Santochi, "Automated sequencing and sub-assembly detection in assembly planning," *CIRP Annals - Manufacturing Technology*, vol. 41, no. 1, pp. 1–4, 1992.
- [10] Y. Z. Zhang, J. Ni, Z. Q. Lin, and X. M. Lai, "Automated sequencing and sub-assembly detection in automobile body assembly planning," *Journal of Materials Processing Technology*, vol. 129, no. 1-3, pp. 490–494, 2002.
- [11] O. Cizsak, "Computer aided determination of the assembly sequence of machine parts and sets," *Advances in Engineering Software*, vol. 48, no. 1, pp. 17–26, 2012.

- [12] M. Wu, Y. Zhao, and C. Wang, "Knowledge-based approach to assembly sequence planning for wind-driven generator," *Mathematical Problems in Engineering*, 2013.
- [13] J. Guo, H. Tang, Z. Sun et al., "An improved shuffled frog leaping algorithm for assembly sequence planning of remote handling maintenance in radioactive environment," *Science and Technology of Nuclear Installations*, vol. 2015, Article ID 516470, 16 pages, 2015.
- [14] S. Motavalli and A.-U. Islam, "Multi-criteria assembly sequencing," *Computers & Industrial Engineering*, vol. 32, no. 4, pp. 743–751, 1997.
- [15] Y. K. Choi, D. M. Lee, and Y. B. Cho, "An approach to multi-criteria assembly sequence planning using genetic algorithms," *The International Journal of Advanced Manufacturing Technology*, vol. 42, no. 1, pp. 180–188, 2009.
- [16] G. Karthik and S. Deb, "A Methodology for assembly sequence optimization by hybrid cuckoo-search genetic algorithm," *Journal of Advanced Manufacturing Systems*, vol. 17, no. 1, pp. 47–59, 2018.
- [17] J. A. A. Mukred, Z. Ibrahim, I. Ibrahim et al., "A binary particle swarm optimization approach to optimize assembly sequence planning," *Advanced Science Letters*, vol. 13, no. 1, pp. 732–738, 2012.
- [18] R. B. Gottipolu and K. Ghosh, "Integrated approach to the generation of assembly sequences," *International Journal of Computer Applications in Technology*, vol. 8, no. 3-4, pp. 125–138, 1995.
- [19] D. Karaboga, "An idea based on Honey Bee Swarm for Numerical Optimization," Technical Report-TR06 TR06, Erciyes University, Turkey, 2005.
- [20] D. Ye and Z. Chen, "A new approach to minimum attribute reduction based on discrete artificial bee colony," *Soft Computing*, vol. 19, no. 7, pp. 1893–1903, 2015.
- [21] I. Ribas, R. Companys, and X. Tort-Martorell, "An efficient Discrete Artificial Bee Colony algorithm for the blocking flow shop problem with total flowtime minimization," *Expert Systems with Applications*, vol. 42, no. 15-16, pp. 6155–6167, 2015.
- [22] C. Ozturk, E. Hancer, and D. Karaboga, "Dynamic clustering with improved binary artificial bee colony algorithm," *Applied Soft Computing*, vol. 28, pp. 69–80, 2015.
- [23] C. Ozturk, E. Hancer, and D. Karaboga, "A novel binary artificial bee colony algorithm based on genetic operators," *Information Sciences*, vol. 297, pp. 154–170, 2015.
- [24] Y.-Y. Han, D. Gong, and X. Sun, "A discrete artificial bee colony algorithm incorporating differential evolution for the flow-shop scheduling problem with blocking," *Engineering Optimization*, vol. 47, no. 7, pp. 927–946, 2015.
- [25] Z. Cui and X. Gu, "An improved discrete artificial bee colony algorithm to minimize the makespan on hybrid flow shop problems," *Neurocomputing*, vol. 148, pp. 248–259, 2015.
- [26] D. Karaboga and S. Aslan, "A discrete artificial bee colony algorithm for detecting transcription factor binding sites in DNA sequences," *Genetics and Molecular Research*, vol. 15, no. 2, Article ID 15028645, 2016.
- [27] Y. Huo, Y. Zhuang, J. Gu, S. Ni, and Y. Xue, "Discrete gbest-guided artificial bee colony algorithm for cloud service composition," *Applied Intelligence*, vol. 42, no. 4, pp. 661–678, 2015.
- [28] T. Batbat and C. Öztürk, "Protein structure prediction with discrete artificial bee colony algorithm," *International Journal of Informatics Technologies*, vol. 9, no. 3, pp. 263–274, 2016.
- [29] Ö. Özmen, T. Özen, and C. Sinanoğlu, "Hacim Kriterine Göre Matris Temelli Montaj Sırası Planlama Sistem Tasarımı," in *Proceedings of the The First International Symposium on Industrial Design Engineering (ISIDE14)*, pp. 278–284, Karabük, Turkey, May 2014.

## Review Article

# Operation Optimization of Natural Gas Transmission Pipelines Based on Stochastic Optimization Algorithms: A Review

Xia Wu <sup>1</sup>, Changjun Li <sup>1</sup>, Yufa He,<sup>2</sup> and Wenlong Jia <sup>1</sup>

<sup>1</sup>*School of Petroleum Engineering, Southwest Petroleum University, Chengdu 610500, China*

<sup>2</sup>*CNOOC Research Institute, Beijing 100027, China*

Correspondence should be addressed to Xia Wu; [xiawu-swpu@hotmail.com](mailto:xiawu-swpu@hotmail.com)

Received 7 October 2017; Revised 1 February 2018; Accepted 27 February 2018; Published 12 April 2018

Academic Editor: Guillermo Cabrera-Guerrero

Copyright © 2018 Xia Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Operation optimization of natural gas pipelines has received increasing attentions, due to such advantages as maximizing the operating economic benefit and the gas delivery amount. This paper provides a review on the most relevant research progress related to the steady-state operation optimization models of natural gas pipelines as well as corresponding solution methods based on stochastic optimization algorithms. The existing operation optimization model of the natural gas pipeline is a mixed-integer nonlinear programming (MINLP) model involving a nonconvex feasible region and mixing of continuous, discrete, and integer optimization variables, which represents an extremely difficult problem to be solved by use of optimization algorithms. A survey on the state of the art demonstrates that many stochastic algorithms show better performance of solving such optimization models due to their advantages of handling discrete variables and of high computation efficiency over classical deterministic optimization algorithms. The essential progress mainly with regard to the applications of the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Simulated Annealing (SA) algorithms, and their extensions is summarized. The performances of these algorithms are compared in terms of the quality of optimization results and the computation efficiency. Furthermore, the research challenges of improving the optimization model, enhancing the stochastic algorithms, developing an online optimization technology, researching the transient optimization, and studying operation optimization of the integrated energy network are discussed.

## 1. Introduction

The natural gas is a low-carbon, clean, and high-quality energy source. The BP Statistical Review reports that the global natural gas consumption amount has continuously increased by an average rate of 2.3% per year over past ten years, as shown in Figure 1. The total gas consumption reached  $3.54 \times 10^{12}$  m<sup>3</sup> in 2016 [1], which accounted for 25% of the primary energy production in the world. The pipeline is one of the most important ways that transport large amounts of natural gas from sources to end consumers due to its convenience, economy, and reliability. By the end of 2017, the total natural gas pipeline length worldwide was more than  $270 \times 10^4$  km [2]. In particular, the United States has more than 210 natural gas pipeline systems and  $198 \times 10^4$  km of natural gas pipelines, which represents the largest natural gas pipeline network in the world.

Normally, pipeline operators concern about three essential objectives, namely, the natural gas delivery amount, the economic benefit, and the line pack defined as the volume of natural gas stored in the pipeline at any moment, when operating a pipeline [3]. The gas delivery amount is usually constrained by production amounts of gas sources, consumption amounts required by consumers, the maximum allowable transmission amount of the pipeline, and gas stored in external facilities such as underground gas storage facilities. The economic benefit is determined by the gas purchasing expenditures, the gas sales income, and the pipeline operating cost. Lastly, the line pack relies on the pressures and temperatures along the pipeline. The aim of the pipeline operation optimization is to reach the maximum natural gas delivery amount, or to reach the maximum line pack, or to reach the maximum economic benefit, or to reach two or more objectives simultaneously. For example, the

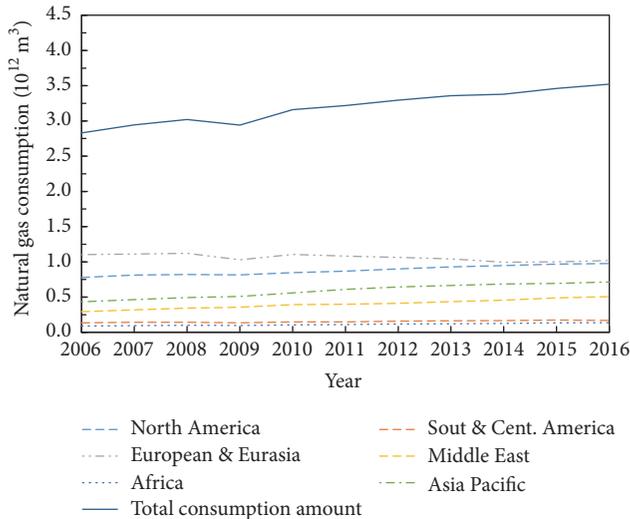


FIGURE 1: The natural gas consumption amount during the past ten years. The data were taken from BP Statistical Review of World Energy June 2017 [1].

optimization objective can be stated as maximizing the gas delivery amount while minimizing the total fuel consumption of the compressor stations. However, these two objectives are not easy to be reached simultaneously because a higher pressure is necessary for transporting larger gas delivery amount, whereas more fuel consumption amounts of compressors are required when acquire higher pressures. The higher gas delivery amounts do not necessarily mean high economic benefits. In other words, these two objectives conflict with each other. Ríos-Mercado and Borraz-Sánchez [4] did an excellent review on various optimization objectives.

The operation scheme of a natural gas pipeline usually consists of the pressures, temperatures, gas delivery amounts at all gas sources and gas terminals, the running statuses and powers of all compressors, and so on. The operation scheme is often developed based on the experience of the operator or by use of the pipeline simulation method [4]. There are two groups of methods broadly used to get the optimal operation schemes of natural gas pipelines: the operation scheme evaluating method and the mathematical optimization method. The previous method selects the best scheme from a set of existing and feasible schemes; thus the results are naturally limited by initial candidate schemes [3]. The latter method calculates the optimal scheme by building and solving a mathematical optimization model that typically consists of an objective function and many necessary constraints [5]. The objective function generally covers one or more optimization objectives as discussed earlier, and the constraints limit the optimization variables within specific physical bounds [6–8]. Obviously, the latter method is more likely to get a higher quality operation scheme in comparison with the previous one. Nevertheless, the operation optimization model has already been recognized as a nonconvex nonlinear problem (NLP) involving linear and nonlinear constraints and equality and inequality equations [9]. Furthermore, if the number

of running compressors along the pipeline is considered, the model will become a more complicated mixed-integer NLP (MINLP) model. These features make the solution of the model extremely difficult. Pfetsch et al. [10] made a detailed review on the NLP for the operation optimization problem of natural gas pipelines.

Over the past decades, a huge number of algorithms have been proposed to solve the optimization problems. The dynamic programming (DP), the generalized reduced gradient (GRG), and the linear programming (LP) methods are three typical deterministic methods that have been extensively involved in solving the operation optimization model of natural gas pipelines, especially to solve the minimization fuel cost problem (MFCCP) of the compressors [4]. The DP is the most successful method among these methods because it guarantees the global optimum and easily handles nonlinearity problem [5, 11]. However, the computation cost of the DP algorithm increases exponentially with increasing the number of the problem's dimension. Hence, the DP is difficult to be extended to large-scale and complicated pipeline networks involving hundreds of gas sources, consumers, and pipes. In comparison with DP, the GRG method handles the dimensionality issue relatively well and, thus, can be applied to large-scale and complicated pipeline networks [7]. However, the GRG method is easily entrapped into the local optimum due to the gradient search strategy. Many improved LP methods have a solid mathematical background [10] and the ability to find the global optimum, but they are not designed for solving the NLP problem. Moreover, none of the above three deterministic methods are designed for solving the resulting MINLP model related to some special optimization problems of natural gas pipelines [4].

Unlike previous deterministic methods, some newly emerging stochastic optimization algorithms have shown many advantages over classical deterministic methods in terms of dealing with the large-scale pipeline network and the MINLP problem. Recently, a number of successful industry projects associated with applications of stochastic algorithms have been reported, including applications of the Genetic Algorithm (GA) [12, 13], Ant Colony Optimization (ACO) [13], Simulated Annealing (SA) optimization [14], Particle Swarm Optimization (PSO) [15], and their extensions. These achievements show prospective ways of efficient solving the operation optimization model of natural gas pipelines.

The present review primary focuses on progress with regard to steady-state operation models of natural gas pipelines and related solution methods based on stochastic algorithms. In what follows, we briefly introduced the components consisting of gas pipelines and corresponding gas network topologies. Then, we summarized the most significant progress on operation optimization models, including the objective functions and constraints. After that, we reviewed the notable applications of GA, PSO, ACO, SA, and other algorithms on solving operation optimization problems. Also, the comparisons between stochastic and classical deterministic optimization algorithms are presented. Finally, the major challenges in this field are discussed in Section 4.

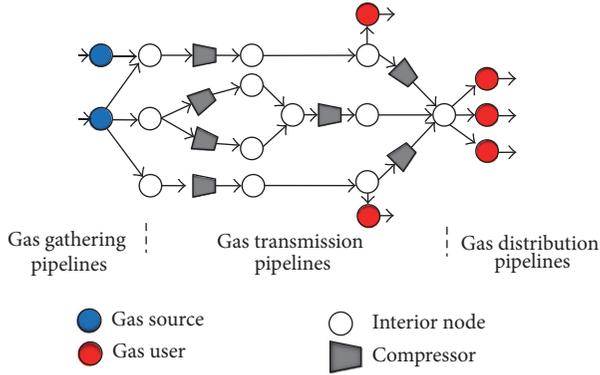


FIGURE 2: A schematic of a natural gas transmission system.

## 2. The Optimal Operation Model of Natural Gas Pipelines

**2.1. Natural Gas Pipelines.** The schematic of a typical natural gas transmission system is shown in Figure 2, which contains a set of gas gathering pipelines, transmission pipelines, distribution pipelines, compressor stations, and distribution stations [16]. The task of the gas gathering pipelines is gathering raw natural gas at production wells and transporting gas to processing plants. After removing impurities, transmission pipelines transport clean natural gas thousands of kilometers from processing plants to city gate stations. Finally, distribution pipelines distribute gas to end consumers.

The above three types of pipelines mainly vary with materials, diameters, operating pressure, and transmission distance. Among them, the transmission pipelines have the largest values regarding pressures, diameters, and lengths. Steel transmission pipelines are typically between 24 and 36 inches in diameter and operated at pressures ranging from 3 to 12 MPa. Due to the elevation change and friction loss along the pipeline, compressors installed in series or parallel are required in order to compensate for the lost pressure of the gas. The distance between two compressor stations typically varies from 90 to 180 km. It is estimated that 3–5% of the gas transported is consumed by the compressors, which takes 25–50% of the total operating budget of the pipeline [20, 21].

In reality, the natural gas transmission system is more complicated than that depicted in Figure 2. The trunk pipeline together with a lot of pipeline branches usually results in three kinds of pipeline network topologies: (a) linear or a gun barrel, (b) tree or branched, and (c) cyclic [4]. An optimization method, of course, should be applicable to pipeline networks with any kind of topologies.

A node-element matrix method is a practical way that depicts the pipeline network's topology. This method assumes that the pipeline system is composed a number of nodes and elements [22, 23]. The element contains all the pipes and devices, including compressors, regulators, valves, and other facilities, while gas sources, consumers, and connection points of elements are defined as the node. Using this method, the topology of a gas network involving  $N_n$  nodes and  $N_e$  elements can be described by a  $N_n \times N_e$  matrix.

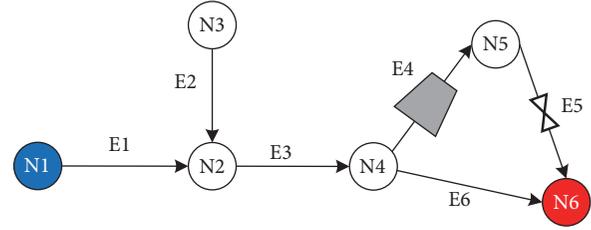


FIGURE 3: A gas pipeline network composed of a tree and a cyclic network.

Figure 3 shows a combined tree and cyclic gas pipeline network consisting of six nodes, four pipes, and two devices (one compressor and one valve). The corresponding node-element matrix is expressed by (1), where each row refers to an element and each column represents a node. The values 1 and  $-1$  in the matrix refer to the upstream and downstream nodes of the element, respectively. The value 0 indicates that there is no connection between the node and the element [15, 24]. Thus, all the inflow and outflow elements that connect with a specified node can be found by searching a fixed column of the resulting node-element matrix.

$$\alpha = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix}. \quad (1)$$

Normally, a pipeline has two operating statuses: steady state and transient state. The steady state indicates the status that all the operating parameters of the pipeline do not vary with time. On the contrary, the transient state considers the changes of operating parameters with time. Although the actual pipeline is in a rigorous transient state [25], using a series of steady state, namely, the quasisteady state method, is a practical way to approximate the real transient optimization problem [17], as shown in Figure 4. Therefore, in this paper, we focus on the steady-state operation optimization problem.

### 2.2. The Operation Optimization Model

**2.2.1. Objective Functions.** The objective function represents the target that should be achieved by use of the optimization method. For the optimal operation of natural gas pipelines, the objective functions generally fall into three aspects: (a) maximize the total throughput or maximize the gas delivery amount for a specific consumer, (b) maximize the line pack defined as the volume of natural gas stored in the pipeline at any moment, and (c) maximize the economic benefit [18].

(i) *Gas Delivery Amount Maximization.* The total gas delivery amount reflects whether the pipelines and related facilities have been sufficiently utilized or not [15, 18]. Constrained by gas supply plans, the gas delivery amount is usually fixed in

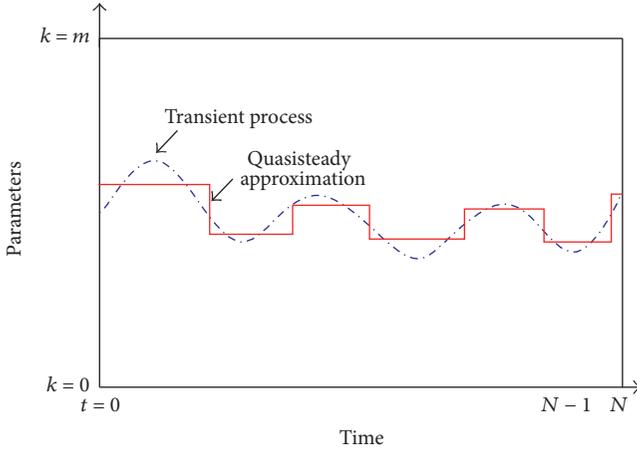


FIGURE 4: Approximation of the transient process by use of a quasisteady method [17].

a specific range. However, if a specified pipeline is a part of a pipeline network, maximizing the gas delivery amount will improve its contribution to the total transmission amount of the whole pipeline network. Thus, the gas delivery amount maximization is often considered as an objective function. Its general form is given by

$$\max f_D = \sum_{i=1}^{N_n} (\beta Q_{ni}), \quad (2)$$

where  $f_D$  is the gas delivery amount;  $N_n$  is the total number of nodes;  $Q_{ni}$  is the gas flow rate at the  $i$ th node;  $\beta$  is a coefficient; if the node is a gas source,  $\beta = 1$ ; otherwise,  $\beta = 0$ ; the subscript  $i$  refers to the node index.

(ii) *Line Pack Maximization.* The natural gas consumption amount varies with the time; thus the pipeline is also designed to be a short-term natural gas storage tool in order to meet the gas peak demands or to alleviate the fluctuation of gas supply. The line pack is defined as the volume of natural gas stored in the pipeline at any moment. Obviously, maximizing the line pack improves the pipeline peak ability [18, 28]. This objective function is formulated as follows:

$$\max f_{LP} = \sum_{j=1}^{N_p} (L_{Pj}), \quad (3)$$

where  $f_{LP}$  is the total line pack volume;  $N_p$  is the total number of pipes;  $L_{Pj}$  is the line pack volume of the  $j$ th pipe; the subscript  $j$  refers to the pipe index.

(iii) *Operation Economic Benefit Maximization.* The operation economic benefit is defined as the difference between the gas sales income and the costs associated with the gas purchasing, pipeline's operating, management, and compressors running costs [15, 29]. This objective function is given by

$$\max f_B = \sum_{i=1}^{N_n} (S_i Q_{ni}) - \sum_{j=1}^{N_p} (R_j f_m) - \sum_{l=1}^{N_c} (H_{cl} C_{cl} W_l), \quad (4)$$

where  $f_B$  is the economic benefit;  $S_i$  is the gas sale or purchase unity price at the  $i$ th node;  $Q_{ni}$  is the inflow/outflow rate at the  $i$ th node; for the inflow rate,  $Q_{ni}$  is set to a positive value, otherwise  $Q_{ni}$  is set to a negative value;  $R_j$  is the management and operation cost coefficient of the  $j$ th pipe;  $f_m$  is the pipeline management and operation unit cost;  $N_c$  is the total number of compressors;  $H_{cl}$  is the status of the  $l$ th compressor, which is a binary variable (1 or 0);  $C_{cl}$  is the  $l$ th compressor's cost coefficient;  $W_l$  is the  $l$ th compressor's power. The subscript  $l$  refers to the compressor index.

Equation (4) covers a series of operating costs; however, previous research shows that the fuel consumption of all compressors takes 25–50% of the total operating budget of a pipeline [21]. Thus, minimizing the fuel cost would greatly contribute to the economic benefit. The minimum fuel cost problem (MFCP) represents the most popular topic in the field of the optimal operation of natural gas pipelines since 1968 [5, 30]. The MFCP objective function can be extracted from (4) and expressed by the following [6, 12, 31]:

$$\min f_C = \sum_{l=1}^{N_c} (H_{cl} C_{cl} W_l). \quad (5)$$

More recently, another new objective function named the minimization of carbon dioxide emission has been proposed due to the increasing concerns on the environment and greenhouse control issues [32]. The  $\text{CO}_2$  emission amount is positively correlated with fuel consumption, so it can be simultaneously solved with the MFCP issue. For instance, Tabkhi [33] incorporated the fuel cost and  $\text{CO}_2$  emission cost into one minimum operating cost function.

(iv) *Multiobjective Functions.* As previously mentioned, some of the objectives are conflicting. To avoid the nonsolution problem, the traditional optimization models typically involve only one objective. More recently, development of multiobjective optimization algorithms offers a new way to simultaneously optimize two or more objectives by use of a unified optimization model.

Wu et al. [15] proposed a biobjective optimization model that considers the maximum gas delivery amount and the maximum operating benefit by use of a weighted hybrid objective function, which is essentially a single-objective optimization model. Unlike the weighted method, the Pareto optimality provides a method to determine the optimal solutions from a set of candidate multiobjective solutions [35]. According to the concept of the Pareto optimality, the multiobjective function can be formulated as follows [23]:

$$\min f(x) = \min (f_1(x), \dots, f_n(x)). \quad (6)$$

The concept of Pareto optimality yields a set of nondominated solutions, named the Pareto-optimal set or Pareto-optimal front, as shown in Figure 5. Sometimes, it is difficult to make a decision on which solution is the best one. Hence, an additional decision-making method is required to select the "best" one from a set of feasible solutions [36]. Related to this issue, Rodriguez et al. [23] proposed a biobjective model to simultaneously optimize the minimum fuel consumption

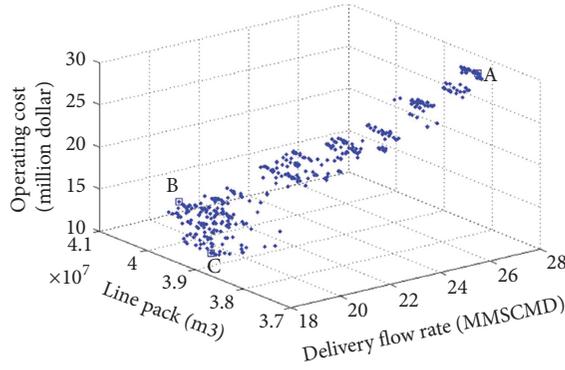


FIGURE 5: Pareto front for the multiobjective optimization of the line pack, the gas delivery amount, and the operating cost [18]. The points A, B, and C on the surface indicate the Pareto optimum points.

and the maximum gas delivery amount. Alinia Kashani and Molaei [18] established a unified model to optimize the maximum gas delivery amount, the maximum line pack, and the minimum fuel cost. In spite of that, published achievements regarding the multiobjective optimization are much fewer than the single-objective optimization problem.

### 2.2.2. Constraints

(i) *Inequality Constraints.* Inequality constraints are used to limit pipeline flow rates, pressures, and temperatures in specified ranges. The inequality constraints are often given at each node as follows:

$$\begin{aligned} Q_{nimin} &\leq Q_{ni} \leq Q_{nimax} & i = 1, 2, \dots, N_n \\ P_{nimin} &\leq P_{ni} \leq P_{nimax} & i = 1, 2, \dots, N_n \\ T_{nimin} &\leq T_{ni} \leq T_{nimax} & i = 1, 2, \dots, N_n \end{aligned} \quad (7)$$

where  $Q_{ni}$  is the volume inflow/outflow rate at the  $i$ th node;  $T_{ni}$  is the temperature at the  $i$ th node;  $P_{ni}$  is the pressure at the  $i$ th node; subscripts min and max refer to the minimum and maximum allowable values, respectively.  $T_{ni}$  can be calculated from the energy equation for the one-dimensional pipe flow [16].

(ii) *Equality Constraints.* Equality constraints mainly represent the governing equations of gas flowing in pipelines, which include the mass balance, pressure, and temperature equations. The equality constraints are given by following equations [15, 30]:

$$\sum_{k \in U_i} \alpha_{ik} M_{ik} = 0 \quad i = 1, 2, \dots, N_n; \quad k = 1, 2, \dots, N_e \quad (8)$$

$$P_{Qj}^2 - P_{Zj}^2 = f_p M_j^2 \quad j = 1, 2, \dots, N_p \quad (9)$$

$$T_{downi} = T_{0i} + (T_{upi} - T_{0i}) e^{-aL}, \quad (10)$$

where subscripts  $i$ ,  $j$ , and  $k$  refer to the index of the node, pipeline, and element, respectively;  $U_i$  is the set of elements connected with the  $i$ th node;  $M_{ik}$  is the absolute mass flow

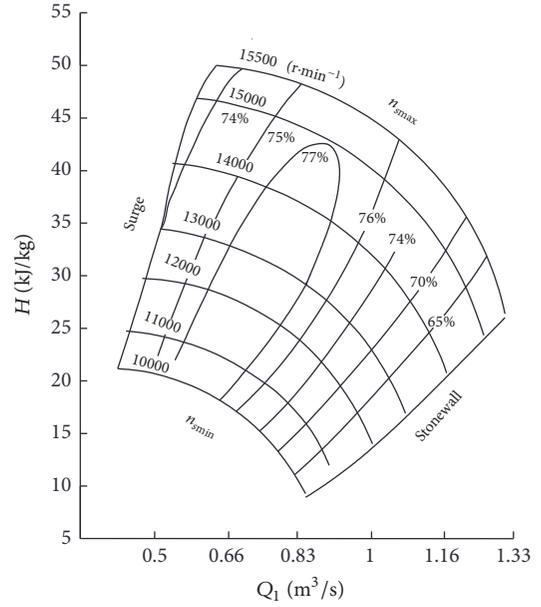


FIGURE 6: The operation envelop of  $Q$ ,  $H$ ,  $n_s$  for a single compressor unit [6].

rate of the  $k$ th element connected with the  $i$ th node;  $\alpha_{ik}$  is a constant,  $\alpha_{ik} = 1$  if the  $k$ th element comes out from the  $i$ th node;  $\alpha_{ik} = -1$  if  $k$ th element goes into the  $i$ th node;  $N_e = N_c + N_p$ .  $P_{Qj}$  is the  $j$ th pipe's inlet pressure;  $P_{Zj}$  is the  $j$ th pipe's outlet pressure;  $M_j$  is the absolute mass flow rate of the  $j$ th pipe;  $T_{upi}$  is the temperature at the pipe's inlet node;  $T_0$  is the environment temperature;  $T_{down}$  is the temperature at the pipe's outlet node;  $f_p$  is pressure function,  $f_p = 16\lambda ZRT/\pi^2 D^5$ ;  $\lambda$  is the friction factor;  $Z$  is the compressibility factor;  $R$  is gas constant;  $D$  is the internal diameter of the pipe;  $a$  is a coefficient defined as  $a = K\pi D/MC_p$ ;  $K$  is the overall heat transfer coefficient;  $C_p$  is the specific heat capacity of natural gas.

(iii) *Compressor Constraints.* There are two types of compressors installed along the gas transmission pipeline: centrifugal and reciprocating compressors. A reciprocating compressor is used to boost natural gas with high pressure and in small quantity, which uses the movement of the piston to create a vacuum inside the cylinder. The centrifugal compressor uses fans to create a vacuum and is used to boost natural gas with relatively lower pressure and in large quantity. Due to the centrifugal compressor can boost the significant higher amount of gas and requires less maintenance than the reciprocating compressor, almost all the trunk natural gas pipelines are equipped with the centrifugal compressor [37]. Hence, in this paper, we only review the significant progress regarding the constraints of centrifugal compressors.

The compressor constraints were originally developed based on an ideal compressor assumption, which does not consider the nonlinear relationships between the compressor's pressure head, power, efficiency, compressor ratio, and volume flow rate, as shown in Figure 6. However, these constraints also have been extensively used, primarily due

to their simplicity [12]. In an outstanding work done by Wu et al. [6], a set of polynomial correlations including the surge and stonewall curves were developed to describe the feasible domain of a centrifugal compressor, thus overcoming the defects of the previous idealized compressor model. These constraints given by (11) have already been widely adopted in gas pipeline optimization models [27, 30, 38] in the past fifteen years. In reality, these constraints are similar to the compressor models embedded in some well-known gas pipeline simulation tools, such as the Pipeline Studio and Synergi Pipeline Simulator [39]. Based on Wu's compressor's constraints, Sanaye and Mahmoudimehr [19] accounted for more correction parameters in compressor's constraints related to the ambient temperature, the part load operation, and the operation out-of-design rotational speed of compressors, yielding constraints that are more likely to represent the real operating behavior of compressors.

$$W = \frac{MH}{\eta}$$

$$\frac{H}{n_s^2} = A_H + B_H \left(\frac{Q}{n_s}\right) + C_H \left(\frac{Q}{n_s}\right)^2 + D_H \left(\frac{Q}{n_s}\right)^3,$$

$$\eta = A_E + B_E \left(\frac{Q}{n_s}\right) + C_E \left(\frac{Q}{n_s}\right)^2 + D_E \left(\frac{Q}{n_s}\right)^3 \quad (11)$$

$$n_{s\min} \leq n_s \leq n_{s\max}$$

$$\text{surge} \leq \frac{Q}{n_s} \leq \text{stonewall},$$

where subscripts  $H, s, E$  refer to the pressure head, rotation speed, and efficiency, respectively;  $A, B, C, D$  are coefficients;  $H$  is the pressure head;  $n$  is the rotation speed;  $\eta$  is the efficiency.

In addition to the compressor's operating parameters, the compressor's running status (ON or OFF) is an important parameter needed to be optimized, in particular for the systems with a series of compressors. Wu et al. [15] used the values of unity and zero to describe the compressor's status, which introduces new discrete variables into the model. Alternatively, the status can be determined according to the ratio of the discharge pressure to the suction pressure of the compressor. The condition that the compressor ratio is equal to or less than the unity indicates the OFF status; otherwise, the status is ON. Hence, the status can be treated as a continuous variable instead of a discrete variable.

**2.2.3. The Optimization Model.** In previous sections, we summarized various objective functions and constraints. For a specified optimization problem, the optimization model can always be formulated as a general form, given by (12), regardless of which objective function is selected and how many constraints are included [15, 18].

$$\begin{aligned} \min \quad & f_m(\mathbf{X}) \quad m = 1, 2, \dots, c \\ \text{Subject to:} \quad & h_i(\mathbf{X}) = 0 \quad i = 1, 2, \dots, \gamma \\ & g_j(\mathbf{X}) \geq 0 \quad j = 1, 2, \dots, \xi, \end{aligned} \quad (12)$$

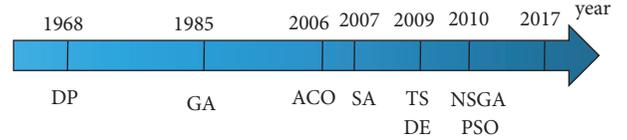


FIGURE 7: The timeline of applying stochastic algorithms to solve operation optimization problems of natural gas pipelines.

where  $f_m$  refers to the objective function;  $\mathbf{X}$  is the vector of optimization variables;  $h_i$  represents equality constraints;  $g$  represents inequality constraints;  $\gamma$  and  $\xi$  represent the total number of the equality constraints and inequality constraints, respectively.

In (12), the optimization variables are parameters in the model to be optimized and often include the pressure, temperature, inflow rate, and outflow rate at each node, the flow rate in each element (the pipe and compressor), and status and power of each compressor, as given by

$$\begin{aligned} \mathbf{X} = [ & P_{n1}, Q_{n1}, T_{n1}, P_{n2}, Q_{n2}, T_{n2}, \dots, P_{nN_n}, Q_{nN_n}, T_{nN_n}, M_1, \\ & M_2, \dots, M_{N_c}, W_1, W_2, \dots, W_{N_c}, H_{c1}, H_{c2}, \dots, H_{cN_c} ], \end{aligned} \quad (13)$$

where the pressures, flow rates, and compressors' powers are continuous parameters and the compressors' statuses are discrete parameters. Due to the presence of a nonlinear objective function, the nonconvex feasible region and mixing of continuous, discrete, and integer optimization variables, the optimization problem is attributed as a mixed-integer nonlinear programming (MINLP) problem that is difficult to be solved by optimization algorithms [40, 41].

### 3. Model Solution Based on Stochastic Optimization Algorithms

Unlike the conventional deterministic algorithms, the stochastic methods are inspired by either the social behavior of biological species or natural biological evolution, resulting in the evolutionary and heuristic algorithms. The biological species used in these stochastic algorithms originally are gathered by individual samples, so the solution strategy does not rely on the gradient information and is able to adapt to discrete variables. Also, stochastic algorithms are shown to be more efficient than some classical deterministic algorithms for solving the MINLP problems [12, 15, 42]. In this paper, we reviewed the essential progress on solving the operation optimization problems of natural gas pipelines by use of the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Simulated Annealing (SA), and other algorithms including the Differential Evolution (DE) and Tabu Search (TS). Figure 7 shows that using the stochastic algorithms to solve the gas pipeline operation optimization problems represents a growing concern in recent ten years.

**3.1. Genetic Algorithm (GA).** The Genetic Algorithm (GA) is a kind of metaheuristic search algorithm based on the Darwinian's evolutionary ideas of natural selection and

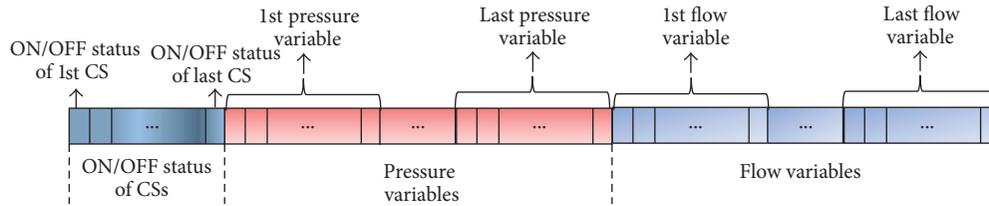


FIGURE 8: Schematic diagram of the chromosome of GA [19]. Each chromosome is composed of all variables in the optimization model.

genetics [36], which was first applied to solve the operation optimization problems of natural gas pipeline in 1987 by Goldberg [43]. GA starts from a population of individuals that represent feasible solutions of the optimization problem, where each individual is represented by a chromosome composed of all the optimization variables, as shown in Figure 8. The optimal solution is then obtained by implementing procedures of the selection, crossover, and mutation on the original individuals (chromosomes) through consecutive generations. In each generation, a fitness function related to the objective function and constraints is firstly used to evaluate the fitness value of each individual. The individuals having higher fitness values are more likely to be selected as parents to generate new individuals in the next generation, and some of the individuals having lower fitness values are dropped. Then, the crossover operator is applied to recombine different portions of selected individuals. As a result, some better candidate individuals might be created in the next generation. Finally, the mutation operator is randomly applied to some individuals, yielding the enhancement of the global search ability of the algorithm. These procedures are repeated until all individuals reach a satisfactory fitness value, or alternatively, say that until the algorithm converges [44].

Goldberg [43] firstly applied GA to solve the natural gas pipeline optimization problem. He built an operation optimization model for a natural gas pipeline system with forty compressors for the purpose of minimizing the total power of all compressors. He assumed that the compressors' efficiencies and head rises are constants; thus the variables to be optimized are the compressors' running statuses. After that, the initial population consisting of 100 individuals that represent all compressors' statuses was generated. The crossover probability and mutation probability were set to 0.7 and 0.01, respectively. Computations show that near-optimal results are obtained after 50 generations of evolution. In particular, even three different initial populations are applied, they finally yield similar optimal results, which shows a good convergence and reproducibility. Goldberg [45] later extended the GA to the transient optimization of a single line pipeline. He suggested that GAs are ready for application to other more difficult optimization cases.

Based on Goldberg's achievements, the most important progress focuses on using newly improved GAs to solve the optimization problems. Li et al. [24] used an adaptive genetic algorithm (AGA) proposed by Srinivas and Patnaik [46] to solve the pipeline operation optimization problem. The AGA method dynamically adjusts the crossover and mutation probabilities according to the average fitness of the

population, the maximum fitness of all individuals so far, and the fitness of a specified individual, which yields lower crossover and mutation probabilities when the specified individual is closer to the best individual so far. Therefore, the historical information belonging to "better individuals" is retained, and the convergence performance can be significantly enhanced in comparison to the original GA. Li et al. [24] reported a successful application of the AGA to a pipeline network consisting of 22 pipes and 21 nodes. However, there are no compressors in the example studied and, also, they did not compare the performance of the AGA with other existing GAs. Sanaye and Mahmoudimehr [19] applied the GA to solve the MFCP problem of three types of natural gas transmission network including the linear, branched, and cyclic pipeline structures. A detail compressor model associated with the pressure, efficiency, stonewall and surge curves, and related correction parameters is presented. Sanaye and Mahmoudimehr [19] presented that the GA has higher computation speed in comparison to the nonsequential dynamic programming (NDP) because the NDP's computing time exponentially depends on pressure and flow rate step sizes, as shown in Figure 9. MohamadiBaghmolaei et al. [47] used the GA to solve the MFCP of a real 56-inch pipeline involving four boosting units located in the south of Iran. They developed an Artificial Neural Network (ANN) model to describe the relationship between compressor's flow rate, head rise, and rotation speed instead of using the conventional compressor's equations as previously mentioned.

In addition to the single-objective optimization, significant achievements have been reached with regard to the multiobjective optimization by use of GAs. Botros et al. [48] reported several successful multiobjective optimization cases of applying GAs to large-scale pipeline networks operated by TransCanada Pipeline Ltd. They considered four different objectives: (a) minimizing fuel consumption, (b) minimizing fuel consumption while maximizing gas delivery amount, (c) minimizing line pack while maximizing gas delivery amount, and (d) maximizing line pack while maximizing gas delivery amount. In comparison to Goldberg's model [43], this model accounts for the constraints associated with the compressors' surge and stonewall curves related to the compressors' flow rates and rotation speeds. They built the fitness function by incorporating the objective function and constraints in the penalty function manner, which is shown to be sufficient enough for these optimization problems studied. This method was successfully applied to two pipeline networks. One of them has 10 compressor stations and 20 optimization variables, while another one has

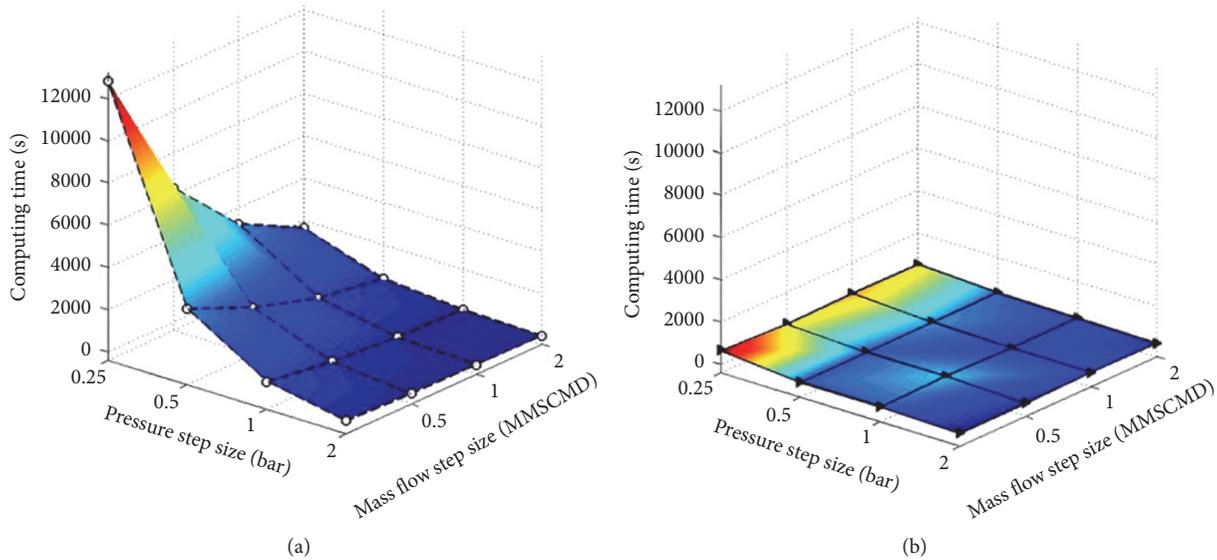


FIGURE 9: Comparison of required computing time for NDP and GA methods [19]. (a) Required computing time for the NDP method; (b) required computing time for the GA method.

25 compressor stations, resulting in 54 decision variables. The optimization results are given by Pareto fronts. However, when evaluating 50000 design cases, the total computing time is approximately one day, which indicates a relative expensive computation cost. Later, they researched more improved methods to enhance the computation efficiency, such as the hybrid GA and gradient-based algorithm and the parallel processing technical [49]. These improved methods were incorporated into an automated optimization system belonging to TransCanada Pipeline Ltd.

More recently, a novel Nondominated Sorting Genetic Algorithm II (NSGA-II) has attracted growing attentions for solving the multiobjective operation optimization problem [50]. NSGA-II uses a fast nondominated sorting approach, which reduces the amount of computation from  $O(mn^3)$  ( $m$  refers to the number of objectives and  $n$  refers to population size) in the NSGA to  $O(mn^2)$ . Also, an elitism strategy is adopted in NSGA-II, ensuring some good individuals not to be discarded in the evolutionary process. Finally, NSGA-II applies the fast crowded distance estimation procedure and simple crowded comparison operator, overcoming the problem of the NSGA that requires the specification of a sharing parameter. As a result, it is able to find a much better spread of solutions and better convergence near the true Pareto-optimal front in comparison to the Pareto-archived evolution strategy and strength-Pareto evolutionary algorithms [51].

Rodriguez et al. [23] used the NSGA-II to simultaneously optimize the maximum gas delivery amount, the minimum fuel consumption, and related CO<sub>2</sub> emissions amount of compressors. The Pareto front obtained from the multiobjective optimization algorithm can be served to determine the minimum and maximum line pack while considering the gas delivery and CO<sub>2</sub> emission amount. Also, the results offer a practical way to determine the minimal

CO<sub>2</sub> emission amount for a given gas delivery amount by operating compressor stations in appropriate manners. Alinia Kashani and Molaei [18] employed the NSGA-II algorithm to balance the maximum gas delivery amount, the maximum line pack, and the minimum operating cost for a pipeline consisting of two gas sources, ten pipe segments, and five compressors. In this model, the gas flow rates and pressures of gas sources, the statuses, and rotational speeds of compressors are taken as optimization variables. Hu et al. [31] used the NSGA-II algorithm to solve a multiobjective optimization model for planning the combined gas and electricity network. Both of the investment cost and the production cost of the combined system are minimized. In particular, a fuzzy decision approach is employed to select the final optimal solution from the resulting Pareto front.

In summary, GAs have been extensively involved in solving the optimal operation problems of natural gas pipelines over past 30 years. The feasibility and effectiveness of GAs also have been proven by a number of worldwide successful industry projects. More recently, the NSGA-II is shown to be a promising algorithm for solving multiobjective optimization models for natural gas pipelines [18, 31].

**3.2. Particle Swarm Optimization (PSO) Algorithm.** The Particle Swarm Optimization (PSO) algorithm is an evolutionary optimization algorithm inspired by the movement of organisms in a flock of birds or a school of fish [52]. Similar to GA, the candidate solutions in PSO are composed of a population of particles spread in the solution space. Each particle is featured by a position and a velocity, where the position represents a feasible solution for the optimization problem in the multidimensional solution space, and the velocity moves the particle itself from one position to another position over consecutive iterations. At each iteration step, every candidate particle is quantitatively evaluated by use of a fitness function.

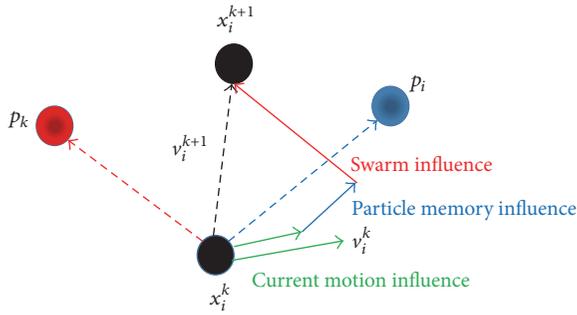


FIGURE 10: A schematic for the position updating of a particle [26].  $x_i^k$  represents the position of the  $i$ th particle at the  $k$ th step, namely, the old position;  $x_i^{k+1}$  is the new position;  $v_i^k$  is the current speed;  $p_i$ ,  $p_k$  refer to the personal and global best positions, respectively. The move speed  $v_i^{k+1}$  is calculated from the current speed and swarm influence.

During this process, every particle records its personal best position searched so far, and the particle having the best fitness value around the neighborhood is marked as the global/local best particle. The move speed of each particle is determined by the current motion, the personal best position, and the global best position, as shown in Figure 10. Thus, all the particles are prone to move to the region with the highest fitness value. By repeating the moving process, all the particles will eventually aggregate in the highest fitness value region, which indicates the convergence of the PSO algorithm. Since the PSO makes few assumptions regarding the problems to be optimized, it is able to search very large solution spaces.

The PSO algorithms cannot guarantee the global optimum solution because of the premature problem that indicates the particles converge to a local optimum region rather than a global one. Thus, some improvement strategies have already been applied to either the PSO algorithm itself or the optimization model in order to alleviate its premature problem. In fact, the prematurity is naturally a feature of many other algorithms, such as GA and ACO as will be discussed later.

Adjusting the inert weight represents a growing concern in reducing the optimization problem by use of the PSO algorithm. The inert weight represents the effect of the current motion of the particle on the movement of the next step. A higher inert weight indicates that the current motion has a greater effect on the movement of the particle. Li et al. [53] used the PSO algorithm to solve the MFCP of natural gas pipeline networks in which the relaxed gas flow equations are used as constraints. The relaxed flow equations accept approximate solutions of the flow equations as candidate particles, which enlarges the solution searching domain of this PSO algorithm in comparison with the method of using the rigorous flow equations. They presented that, under the same initial conditions and required accuracy, the proposed method and the simplex algorithm give the optimal gas consumption amounts of  $1.52 \text{ m}^3/\text{s}$  and  $1.5528 \text{ m}^3/\text{s}$ , respectively. The results demonstrate that the PSO algorithm is able to find higher quality solutions in comparison with the LP

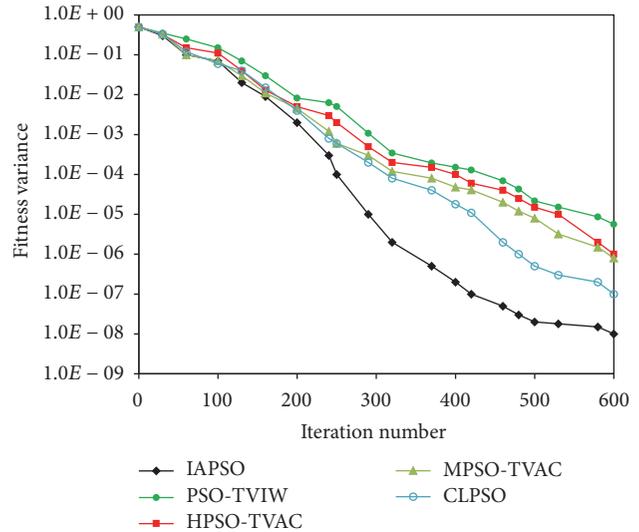


FIGURE 11: The converge curves of different PSO algorithms for an operation optimization problem of a natural gas pipeline [15].

method. Zheng and Wu [27] presented an improved PSO to solve the MFCP of gas pipelines. A method based on an exponential function is employed to change the inert weight of PSO with the evolution generation, which essentially enhances the global search ability and the convergence speed in comparison to the original PSO algorithm. That is, at the beginning, the inertia weight is a large value that enables the PSO algorithm to search large solution spaces, thus avoiding the premature convergence problem. After a certain number of evolution generations, all the particles tend to converge to the near-optimal domain; thus a small inertia weight is used to speed up the convergence. They tested this new method by applying it to a pipeline system with up to 17 compressor stations. The results show that the relative errors of the optimization results given by the improved PSO, ACO, and DP algorithms are less than 1%, whereas the computation cost of the PSO is 4 to 8 times less than that of the ACO and 170 times less than that of the DP. These comparisons show that the proposed PSO algorithm has a high computation efficiency while keeping the similar optimization results.

Wu et al. [15] used an inertia-adaptive PSO (IAPSO) to solve a biobjective optimization model that considers the maximum operating benefit and the maximum gas delivery amount. The IAPSO automatically increases the inertia weight of PSO when the particle is apart from the global best value and conversely and decreases the inertia weight when the particle is approaching the global best value. This adaptive adjustment method gives the IAPSO stronger global search ability when candidate particles are apart from the global best position so far. This method was applied to a pipeline with four compressor stations and compared to five other PSO algorithms including the PSO-TVIW, HPSO-TVAC, MPSO-TVAC, and CLPSO algorithms [54]. They presented that the IAPSO not only shows faster convergence speed but also gives better optimization results in terms of the biobjective functions, as shown in Figure 11. These results show that

the PSO algorithm should be well designed to get the better optimization solution. In other words, the final optimization results are dependent on how the original PSO is improved.

In addition to apply the PSO to solve the operation optimization problems of natural gas pipelines, there are also a number of successful applications in the fields of the operation optimization of oil pipeline [55], the layout optimization of the oil and gas pipelines [56], and the optimal control of compressors [57]. Although these optimization problems are different from the operation optimization of gas pipelines, they also mentioned that some specific techniques should be adopted to overcome the premature shortcoming when using the original PSO.

In summary, the PSO is featured by its high computation speed. Nevertheless, how to alleviate the premature defect of the original PSO algorithm is quite challenging since the main idea in PSO resembles the overall idea in gradient search methods, but adapted to a population. Actually, the parameters setting (e.g., the number of particles and the maximum number of function evaluations) affects the final performance of the PSO algorithm when solving the MINLP problem [15].

**3.3. Ant Colony Optimization (ACO) Algorithm.** The Ant Colony Optimization (ACO) algorithm is a nature-inspired optimization method proposed by Dorigo in 1992 [58]. This method mimics the process of ants searching for the shortest path between the nest and the food source. At the beginning, a school of ants starts from their nest to randomly search for food in the region around the nest. If one ant succeeds in finding food, it goes back to the nest and leaves a chemical pheromone trail along the searching path. The other ants will follow the path identified by the pheromone trail, so all the ants will be able to find the same food source in the shortest way [59]. The two critical procedures of the ACO algorithm are the edge selection and pheromone updating. Each edge represents a feasible solution consisting of the optimization variables. The initial edges are developed by the random search of ants. All edges construct the multidimensional solution space for the optimization problem. The ants move along the edges and left pheromone in edges during the iteration process. The edge with a higher level of pheromone is more likely to be selected by ants, and, in turn, more pheromone is left if more ants pass through one edge [60].

Chebouba et al. [13, 60] are the first people who solve operation optimization problems of natural gas pipelines by use of the ACO algorithm. They used a two-stage technique (local updating and global updating) to update the pheromone trail, yielding an improved two-stage ACO algorithm. This method was applied to solve the MFCP of the Hassi R'mel-Arzew gas pipeline involving one gas source, one gas user, six pipes, and five compressor stations. Computations show that the relative errors between ACO results and DP results are typically less than 1%, but the computation speed of ACO is more than 14 times faster than that of the DP. These achievements prove the feasibility of applying the ACO to solve the gas pipeline optimization problem. In 2012, Chebouba and Megloulouli [42] used a combined ANN and ACO algorithm to optimize the number

of operating compressor and related discharge pressures of a gas pipeline system under the steady-state assumption. In comparison to the previous work, they used the ANN method to describe the compressor's performance rather than the nonlinear equations mentioned in Section 2.2. MohamadiBaghmolaei et al. [47] also employed the ANN to calculate the compressor's performance, but they used the GA as the optimization algorithm instead of the ACO.

Arya and Honwad [20] applied the original ACO to solve the MFCP of a pipeline network involving 18 nodes, 2 compressor stations, and 3 identical compressors in each station. Under the fixed gas flow rate, the fuel costs calculated by the ACO and GRG methods were compared. They presented that the ACO is able to lead a further fuel savings of 0.015 kg/s and a total economic saving of nearly US \$ 350,000 per year in comparison to the results calculated by the GRG method. These results demonstrate that ACO yields higher quality solutions in comparison with the GRG method. However, to the best of our knowledge, ACO has not been implemented in systems with more complicated pipelines so far.

**3.4. Simulated Annealing (SA) Algorithm.** Annealing in metallurgy technical is a process involving heating and controlled cooling of a material in order to change the properties of crystals and to reduce defects of a material [61]. According to the thermodynamic theory, the heating process increases the thermodynamic free energy of the metal material, and the cooling process decreases the thermodynamic free energy. Eventually, the thermodynamic free energy will drop to a global minimum value when the material reaches the equilibrium state. Inspired by such processes, Khachaturyan et al. [62] proposed the Simulated Annealing (SA) algorithm in 1979 that approximates the global optimum of a specified function in a probabilistic searching manner. That is, an energy function is analogous to the thermodynamic free energy of the system and acts as an objective function of the optimization problem. During the cooling process, a current solution can be replaced with a "neighbor" solution with a certain probability even if the current solution is better than the "neighbor" one. This feature enhances the stochastic search and global optimization abilities of the SA. Although Kirkpatrick et al. [63] provided a useful framework for the combinational optimization of large and complex systems based on the SA in 1983, SA has only been applied to operation optimization of natural gas pipelines since 2007.

Zhang et al. [64] employed a hybrid GA and SA (GASA) algorithm to calculate the maximum operating benefit a large-scale gas pipeline network. The investigated pipeline network contains 98 nodes but no compressors. They found that the hybrid GASA has better performance than the improved complex algorithm. Rodríguez et al. [14] presented a research of using a metaheuristic algorithm called Simulated Annealing with GAMS (SAG) to optimize the layout of long-distance hydrocarbon pipelines. They show that SAG is robust because a high percentage of the near-optimal solutions can be found. For more topics with regard to the application of SA to other pipeline optimization problems, the readers are referred to Chen et al. [65], Samora et al. [66], and Zhou et al. [55].

TABLE 1: Comparison of optimization algorithms for solving the operation optimization problems of natural gas pipelines.

Objective function	Optimization algorithm	Reference
MFCP	GA, SIMPLEX	[49]
MFCP	GA, ES, MILP	[72, 73]
MFCP of integrated energy system	NDPTS, DP, GRG	[69]
MFCP	ACO, DP	[13]
MFCP and the maximum gas delivery	GA, $\epsilon$ -constraint	[23]
MFCP	GA, NDP	[19]
MFCP	PSO, ACO, DP	[27]
The maximum operating benefit and the maximum gas delivery	IAPSO, PSO-TVIW, HPSO-TVAC, MPSO-TVAC, CLPSO	[15]
MFCP	ACO, GRG	[20]
MFCP	PSO, GA	[71]

*3.5. Other Algorithms and Hybrid Ones.* Some other algorithms also have been applied to solve the operation optimization problem of natural gas pipelines, including the Differential Evolution (DE) algorithm [67], Artificial Neural Networks (ANN) [47], Support Vector Machine (SVM) [68], and the Tabu Search (TS) algorithm [69]. Also, a lot of efforts have been made to alleviate inherent shortcomings of original stochastic algorithms by use of hybrid strategies which combine advantages of two or more stochastic algorithms yielding new hybrid optimization algorithms.

DE, similar to GA, is an evolutionary algorithm composed of three operations, the mutation, crossover, and selection. However, DE is featured by its mutation process that creates a new solution by adding the weighted difference between two solutions to a third solution. Based on the literature survey, DE is often applied in some hybrid algorithms. Malamura and Murata [67] proposed a hybrid method by combing the Differential Evolution (DE), Genetic Algorithm (GA), and Hybrid Petri net (HPN) methods in order to solve the MFCP of gas networks. This hybrid algorithm is implemented based on the framework of DE, where firstly the HPN gets results in the reduction of complexity in the optimization problem formulation, then DE is used to optimize the continuous variables (flow rates for compressors in each station), and finally GA is applied to optimize the discrete variables (status of compressors) when DE reaches the optimal solution in each iteration. The proposed method reduces the complexity of the optimization problem. Also, it provides an effective way to deal with the optimization model containing discrete and continuous variables. Wu et al. [70] replaced the reproduction operator of the original GA with that of the Differential Evolution (DE) algorithm. The resulting hybrid DE-GA algorithm was applied to get the minimal annual operating cost of compressor stations of a pipeline network system associated with 11 pipes and 2 compressor stations. They demonstrated that the hybrid DE-GA method improves the global optimization ability and effectively overcomes the premature convergence of the original GA.

SVM and ANN are mainly applied to fit or to forecast the pipeline's operation parameters, such as the prediction of

the gas load and the compressor's performance [47, 68]. So, SVM and ANN are usually combined with other optimization algorithms when applied to solve the operation optimization problem of gas pipelines.

Borraz-Sánchez and Ríos-Mercado [69] proposed a hybrid nonsequential DP (NDP) and Tabu Search method, namely, the NDPTS method to get the optimal flow rates and pressures in a natural gas pipeline. They used TS to find the best non-tabu list from an initial feasible set of flows, and then the NDP method determines the optimal pressures. The results demonstrated that the performance of the NDPTS has been improved by 1%–17% in terms of the solution speed and optimized values when compared with GRG and NDP methods.

Besides, variations of many stochastic optimization algorithms have been developed in recent years and applied to other industry fields [71], but applications of them in the operation optimization of natural gas pipelines are rarely reported. On the other hand, hybrid algorithms take advantage of different stochastic algorithms; thus they have better theoretical foundations than some randomly improved algorithms [67, 70]. Hence, applying hybrid stochastic algorithms probably represents a development tendency of solving the operation optimization problems of natural gas pipelines.

*3.6. Comparison of Stochastic Algorithms.* Comparing different optimization algorithms provides insightful information to understand the advantages and disadvantages of each algorithm. Table 1 gives the published literature related to the comparison of optimization algorithms used to solve operation optimization models of natural gas pipelines. Since the global optimization ability and the computation speed represent two main concerns of an optimization algorithm, almost all comparison focus on these two points.

Table 1 shows that comparisons between stochastic algorithms and gradient-based/dynamical programming methods are common. Comparisons between stochastic algorithms and gradient-based (e.g., GRG) methods will demonstrate the global optimization ability of stochastic algorithms due to the fact that gradient-based methods are prone to be entrapped into the local optimum. Arya and Honwad [20]

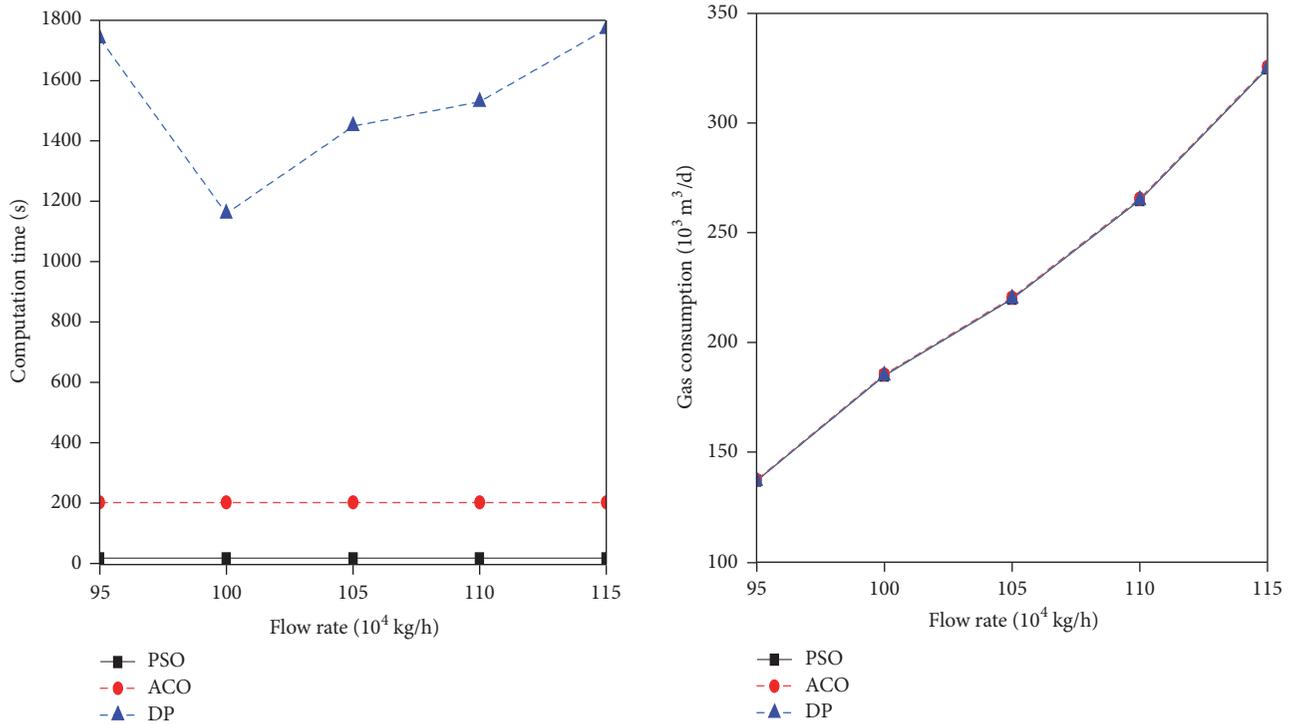


FIGURE 12: Comparison of computing time and gas consumption for a gas pipeline with five compressor stations optimized from PSO, ACO, and DP [27].

used ACO and GRG to solve the MFCP of a gas pipeline with six compressors. The length of the pipeline is 300 km, the diameter is from 0.787 m to 0.889 m, and the mass flow rate is 150 kg/s. The results demonstrated that the ACO yields lower suction pressures and higher efficiencies of compressors in comparison with the results of GRG, resulting in the gas fuel saving of 0.015 kg/s and of approximately US\$350,000 per year. Borraz-Sánchez and Ríos-Mercado [69] solved the MFCP of seven gas pipelines by use of NDPTS and GRG. It is observed that NDPTS yields significant better solutions than the GRG. The relative improvement of NDPTS over GRG is from 0.004% to 112.37% for seven cases.

On the other hand, comparisons between stochastic algorithms and the dynamical programming methods (e.g., DP and NDP) are used to validate the final optimization results and to check the computation efficiency because the DP has the global optimization ability but extremely expensive computation cost. Almost all related comparisons show that stochastic optimization algorithms not only are able to find similar solutions in comparison to the gradient-based methods but also have higher computation efficiency than traditional deterministic algorithms [20, 23, 27, 49]. For example, Nguyen et al. [72, 73] simultaneously optimized the fuel cost, maintenance cost, startup cost, shut cost, and oversupply cost of all compressors by use of the mixed-integer linear programming (MILP), the expert system (ES), and the original GA provided in the MATLAB toolbox. Based on their research findings, MILP and GA give exactly the same results for the costs except for the startup cost, for which GA gives a slightly higher result. On the other

hand, the ES generates worse results with higher costs for all these objectives. Sanaye and Mahmoudimehr [19] applied GA and NDP to solve the MFCP of one branched pipeline and two cyclic pipelines. They presented that the difference between objective values obtained from GA and NDP is in the range of 0–0.55%, which shows a very similar optimization result. However, the computing time of NDP exponentially increases with increasing of the pressure and flow rate step sizes, while the computing time of GA did not show such a dependency on these parameters. Under some specific conditions, the computing time of NDP is 20 times longer than that of GA. Chebouba et al. [13] found that, for the MFCP of gas pipelines, the difference between objective values obtained by the ACO and DP is less than 0.22% while the ACO is 14–27 times faster than the DP.

Besides, comparisons of stochastic algorithms also have been reported in many published papers, which would provide guidance to select a proper optimization method. Published literature reveals that GA gives the similar optimization results as the ACO. Khademi and Khosravi [71] compared the performance of GA and PSO when applying to optimize a weighted summation of the total number of running compressor stations and their total fuel consumption. They showed that GA leads to more accurate results to match the real operating conditions of the pipeline. Zheng and Wu [27] presented the quantitative comparisons of ACO, PSO, and DP in terms of computing time and gas consumption amount when these methods are applied to solve the MFCP of a gas pipeline with five compressor stations, as shown in Figure 12. They found that the differences between the optimization

results obtained by these three methods are in the range of 0–0.37%. However, the computing speed of the PSO is 8.52 times faster than that of the ACO. Moreover, the average computing time of DP method is about 8 times longer than that of the ACO.

Besides different kinds of stochastic optimization algorithms, variations of the same kind of stochastic algorithms show different optimization results as well as the convergence speed. Wu et al. [15] found that when IAPSO, PSO-TVIW, HPSO-TVAC, MPSO-TVAC, and CLPSO are applied to solve the maximum flow rate and maximum operation benefit of a gas pipeline, the difference between the optimal maximum flowrates goes from 2.94% to 6.98%, and the difference between the optimal operation benefit goes from 4.52% to 12.76%. Also, IAPSO presents the fastest convergence speed whereas the PSO-TVIW shows the slowest convergence speed, as depicted in Figure 11.

In summary, the operation optimization model of natural gas pipelines is recognized as a MINLP model which is not easy to be solved even by previous mentioned stochastic algorithms. However, among GA, PSO, ACO, and SA algorithms, GA represents the most successful one over past thirty years in terms of the quality of the optimization results, the computation speed, and the ability of handling the discrete variables, whereas other three algorithms have been used only in recent ten years and their applications are relatively few. The GA and its extensions have been applied to a huge number of pipeline networks that are featured by different scales and topology structures including the gun barrel, the tree, and the cyclic networks. In particular, the NSGA-II has been taken as a standard algorithm for solving multiobjective optimization models in recent years [31, 50]. Nevertheless, it should be noted that stochastic algorithms cannot guarantee the global optimum and the premature convergence problem is naturally a feature of these algorithms.

#### 4. Research Challenges

Application of stochastic algorithms contributes to the technical breakthrough of solving the resulting MINLP problem. However, there are still quite a few areas that pose a wide range of challenges.

The first research challenge focuses on the improvement of the optimization model. The objective functions in existing models mainly cover the economic benefit and environment aspects, such as the gas delivery amount, the line package, the fuel consumption, the CO<sub>2</sub> emission amount, and the pipeline operating and management costs. Besides these objectives, recently, the pipeline safety has attracted increasing concerns because the pipeline failure would cause severe disasters and economic loss [74]. Adding the pipeline reliability objective (e.g., the gas hydrate formation risk and the pipe rupture risk) to existing models is a promising way to simultaneously optimize the pipeline operating economic, environmental, and safety aspects [3], as shown in Figure 13. This work will require further research on the quantitative risk assessment and related failure probability predictions and failure consequence computations [75]. Since the safety objective has a different dimension with existing objective

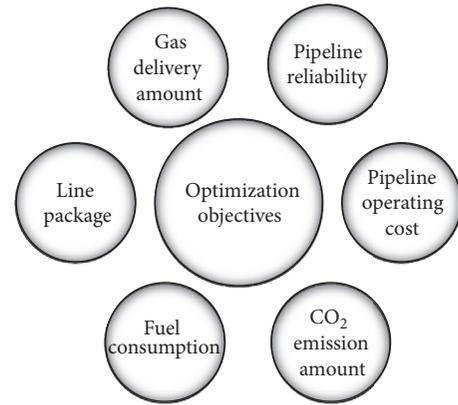


FIGURE 13: The development of optimization objectives.

functions, a proper decision method should be developed to select the best solution from the Pareto front if a multiobjective objective function is used.

In addition to the objective functions, constraints still need to be improved to match the pipeline's real operating cases. Since some constraints use the simplified forms of the gas flow governing equations, they cannot precisely describe the pipe flow. As a result, the optimization results cannot completely satisfy the real pipeline's hydraulic and thermal constraints. Although using the original governing equations in the optimization model is more likely to get optimization results that satisfy the pipeline's real operating conditions, it makes the model itself extremely complicated. A practical method to solve this problem is combining the pipeline simulation model and the optimization model. That is, once the optimization results are obtained from the optimization model, the simulation model is then applied to adjust the optimized parameters that deviate from the simulation models. Considering that the simulation is time-consuming, the simulation method is recommended to be performed every certain number of optimization iterations rather than applying it to each candidate solution at each optimization iteration. The flow chart of combining the pipeline simulation and optimization models is depicted in Figure 14. Except for validating the optimization results, the simulation methods also can offer the initial candidate solutions for stochastic algorithms, as presented by Wu et al. [15].

The second challenge is developing robust and efficient optimization algorithms. The strong global optimization ability and high computation efficiency are two most important concerns of stochastic optimization algorithms. Among GA, PSO, ACO, and SA algorithms, only the GA has been extensively applied in industries. The other three algorithms have not been thoroughly studied. Since many parameters in these algorithms influence the global search ability and the convergence speed, the sensitivity analysis of related parameters should be firstly researched subject to different cases. The expected achievements will yield general rules to set optimal parameters for these algorithms. Then, the performances of these algorithms need to be investigated in order to figure out the applicable ranges of

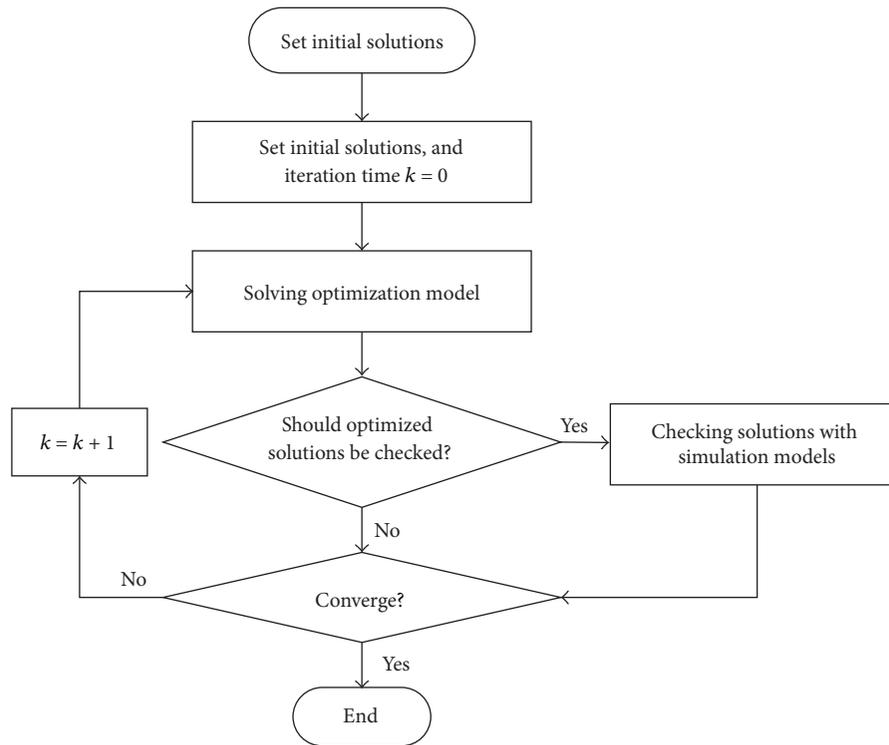


FIGURE 14: The flow chart of combining simulation and optimization models.

each algorithm, which will eventually provide a guidance to select a proper optimization algorithm for a specified case. Based on the sensitivity analyses and comparison of different algorithms, the improvement methods should be researched to overcome some shortcomings of existing optimization algorithms. Hybrid stochastic algorithms could contribute to the significant improvements of the original methods because they combine the advantages of different methods, such as the hybrid SA-GA [70] and DE-GA [64] algorithms. However, published literature regarding this topic is limited. Many hybrid strategies have not yet been applied to the pipeline optimization, such as the hybrid GA-PSO [76] and GA-ACO [77] algorithms. Moreover, most of the published literature focuses on solving the optimization model for a gun barrel or a branched pipeline system. The optimization method for the more complicated cyclic pipeline systems also should be researched.

The third challenge is researching the online optimization technology. The supervisory control and data acquisition (SCADA) system has already been widely installed on pipelines. Combining the online simulation technology and the pipeline operation optimization technology yields an online optimization technology, which is a promising way to make full use of optimization results [78]. The framework of this technology is shown in Figure 15. With the online optimization technology, the actual pipeline operating data collected from the SCADA system can be set as the initial candidate solutions of stochastic algorithms, and, in turn, the optimization results give useful information to optimally adjust the pipeline. Moreover, a decision-making system

makes a decision on whether the suggested optimal operating scheme is adopted or not. Obviously, how to connect the simulation model, optimization model, and the SCADA system is a core problem that needs to be solved when developing an online optimization technology.

The fourth challenge is the research of the transient optimization for natural gas pipelines. Although the previously introduced quasisteady optimization methods can obtain the approximate results of transient optimization, it cannot precisely depict the real transient process. Published literature introduced that the biggest difference between the steady-state and transient optimization problems is the increasing of the constraint equations and decision variables. A transient optimization requires the use of partial differential equations involving the continuity, energy, and momentum equations to describe the change of related decision variables, such as gas flow, velocity, density, pressure, and temperature, with time. As a result, the inherent complexity of the optimization problem has been increased. However, stochastic optimization algorithms have already shown potential advantages over classical optimization algorithms when solving the transient problems. Chebouba [38] used the NSGA-II solver embedded in the modeFRONTIER optimization software to firstly maximize the line pack and secondly to minimize the total power of a pipeline in a period of 24 hours. They presented that this method is able to minimize the CO<sub>2</sub> emission since the CO<sub>2</sub> emission amount has a positive relationship with the compressor's power. Mahlke et al. [79] used the SA to solve the MFCP of a transient natural gas pipeline. They built the energy function by incorporating the

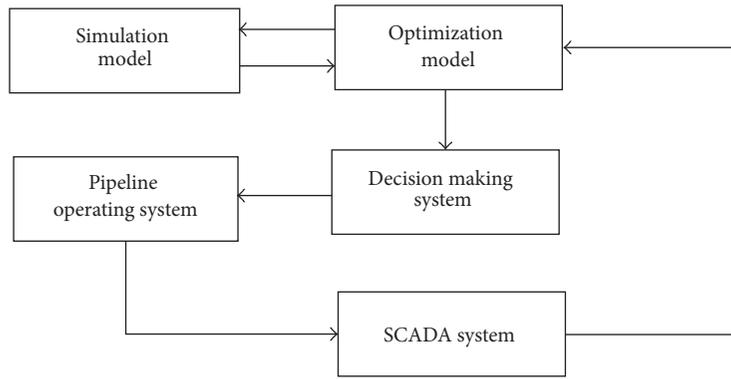


FIGURE 15: The framework of the online optimization technology.

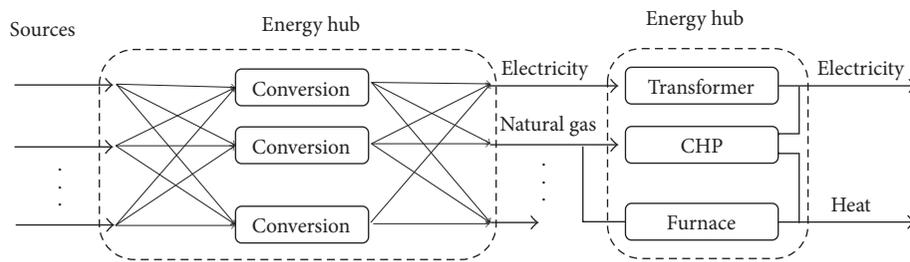


FIGURE 16: The schematic of solving the operation optimization problem of the energy network [34].

objective function and all constraints by use of a penalty function. Many necessary details with regard to performing the SA procedures were presented, including the design of the neighborhood structure, step size selection, generation of the initial feasible solution, and cooling scheme. Besides, in order to enlarge the solution space to be searched, gas flow governing equations are relaxed by adding appropriate penalty factors. In particular, they used piecewise linear functions to approximate the nonlinear equations in the optimization model. Based on these special strategies, they proposed a combinational SA and a branch-and-cut algorithm which is tested by three real-world cases involving 11 to 31 pipes and 3 to 15 compressor stations provided by the German gas company E.ON Ruhrgas AG. It can be concluded that applying the stochastic optimization algorithms will contribute to the development of transient optimization of natural gas pipelines. In summary, works on this area are still in a developing phase [4].

Nowadays, energy network, which is a new concept and includes several infrastructures such as electrical and gas networks, has been proposed [80]. The electrical generation and gas production are named as sources, and all the consumers take energy from the energy hubs. The introduction of the conversion between different energies (electrical and natural gas) makes the energy network highly flexibility to fully utilize all the energies. With this regard, the term “energy flow” is adopted to refer to terms “power flow” and “gas flow.” The aim of the operation focuses on how to simultaneously optimize the different types of energy networks such as

the electrical network and the gas network, as depicted in Figure 16.

In general, the optimization model of the energy network is a nonconvex, nonsmooth, nondifferential, high-dimensional, and highly nonlinear optimal energy flow model. It is extremely difficult to solve. Most recently, Derafshi Beigvand et al. [34] presented that the Gravitational Search Algorithm, PSO, GA, and DE are able to solve the operation optimization model of the energy network involving the electrical and gas networks. Since many hybrid optimization algorithms and variations of the basic PSO, GA, and DE algorithms have already shown advantages over the basic optimization algorithms when solving the operation optimization problem of natural gas pipelines, these hybrid and extended stochastic optimization algorithms are expected to solve the operation optimization of energy networks in the future.

## 5. Conclusions

We reviewed important progress on steady-state operation optimization models of natural gas pipelines and solutions methods based on stochastic algorithms including the GA, PSO, ACO, SA, and other algorithms. The operation optimization model is confirmed to be a mixed-integer nonlinear programming (MINLP) model consisting of a nonconvex feasible region and mixing of continuous, discrete, and integer optimization variables. Current optimization models cover objectives involving the gas delivery amount, the line pack,

the compressor's fuel cost, the management cost, and the CO<sub>2</sub> emission amount. However, the pipeline safety is receiving less attention. Adding the pipeline safety objective function is a promising way to simultaneously optimize the pipeline operating economic, environmental, and safety concerns.

Since 1985, stochastic optimization algorithms have contributed to the technical breakthrough of solving the MINLP problem due to their advantages of handling discrete variable and of higher computation speed over classical optimization algorithms. In particular, GA and its extensions have been successfully applied to a large number of industry cases that are featured by different scales and topology structures including the gun barrel, the tree, and the cyclic networks, and the new NSGA-II algorithm is promising to solve the multiobjective operation optimization problems of natural gas pipelines. Although applications of PSO, ACO, and SA algorithms are relatively few, PSO has already shown much higher computation speed than the GA and ACO. However, the premature problem of all these stochastic optimization algorithms remains unsolved. Moreover, the hybrid algorithms would provide useful methods to alleviate the shortcomings of existing algorithms because the hybrid strategies combine advantages of two or more stochastic algorithms, but related applications with regard to the operation optimization problems of the gas pipeline are rarely reported.

The online optimization technology that integrates the optimization model, the simulation model, the decision-making system, and the SCADA system is a promising way of making use of the optimization results. The pipeline operation data provides initial candidate solutions for the optimization model, and, in turn, the optimization results provide useful information for the pipeline control. Moreover, research challenges involving the transient optimization and operation optimization of the integrated energy network are discussed.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This paper is funded by the National Natural Science Foundation of China (no. 51604233, no. 51504206, no. 51474184, and no. 51674213) and a subproject of the National Science and Technology Major Project of China (no. 2016ZX05028-001-006).

## References

- [1] B. Company, "BP Statistical Review of World Energy 2017" <http://www.bp.com/en/global/corporate/energy-economics/statistical-review-of-world-energy.html>.
- [2] CIA, *The World Factbook*, Central Intelligence Agency, Washington, DC, USA, 2017.
- [3] C. J. Li, W. L. Jia, E. B. Liu, and X. Wu, "A multi-hierarchy grey relational analysis model for natural gas pipeline operation schemes comprehensive evaluation," *International Journal of Industrial Engineering : Theory, Applications and Practice*, vol. 19, no. 6, pp. 241–251, 2012.
- [4] R. Z. Ríos-Mercado and C. Borraz-Sánchez, "Optimization problems in natural gas transportation systems: A state-of-the-art review," *Applied Energy*, vol. 147, pp. 536–555, 2015.
- [5] P. J. Wong and R. E. Larson, "Optimization of Natural-Gas Pipeline Systems via Dynamic Programming," *IEEE Transactions on Automatic Control*, vol. 13, no. 5, pp. 475–481, 1968.
- [6] S. Wu, R. Z. Ríos-Mercado, E. A. Boyd, and L. R. Scott, "Model relaxations for the fuel cost minimization of steady-state gas pipeline networks," *Mathematical and Computer Modelling*, vol. 31, no. 2-3, pp. 197–220, 2000.
- [7] P. B. Percell and M. J. Ryan, "Steady state optimization of gas pipeline network operation," in *PSIG Annual Meeting*, PSIG, Tulsa, USA, 1987.
- [8] M. B. Feldman, "Optimization of Gas Transmission Systems Using Linear Programming," in *PSIG Annual Meeting*, PSIG, Toronto, Canada, 1988.
- [9] A. Martin, M. Möller, and S. Moritz, "Mixed integer models for the stationary case of gas network optimization," *Mathematical Programming*, vol. 105, no. 2-3, pp. 563–582, 2006.
- [10] M. E. Pfetsch, A. Fügenschuh, B. Geißler et al., "Validation of nominations in gas network optimization: Models, methods, and solutions," *Optimization Methods and Software*, vol. 30, no. 1, pp. 15–53, 2015.
- [11] R. G. Carter, "Pipeline optimization: dynamic programming after 30 years," in *PSIG Annual Meeting*, PSIG, Denver, USA, 1998.
- [12] D. E. Goldberg and C. H. Kuo, "Genetic algorithms in pipeline optimization," *Journal of Computing in Civil Engineering*, vol. 1, no. 2, pp. 128–141, 1987.
- [13] A. Chebouba, F. Yalaoui, A. Smati, L. Amodeo, K. Younsi, and A. Tairi, "Optimization of natural gas pipeline transportation using ant colony optimization," *Computers & Operations Research*, vol. 36, no. 6, pp. 1916–1923, 2009.
- [14] D. A. Rodríguez, P. P. Oteiza, and N. B. Brignole, "Simulated annealing optimization for hydrocarbon pipeline networks," *Industrial & Engineering Chemistry Research*, vol. 52, no. 25, pp. 8579–8588, 2013.
- [15] X. Wu, C. Li, W. Jia, and Y. He, "Optimal operation of trunk natural gas pipelines via an inertia-adaptive particle swarm optimization algorithm," *Journal of Natural Gas Science and Engineering*, vol. 21, pp. 10–18, 2014.
- [16] S. Mokhatab and W. A. Poe, *Handbook of natural gas transmission and processing*, Elsevier, 2012, <http://www.sciencedirect.com/science/book/9780123869142>.
- [17] A. Gopalakrishnan and L. T. Biegler, "Economic Nonlinear Model Predictive Control for periodic optimal operation of gas pipeline networks," *Computers & Chemical Engineering*, vol. 52, pp. 90–99, 2013.
- [18] A. H. Alinia Kashani and R. Molaei, "Techno-economical and environmental optimization of natural gas network operation," *Chemical Engineering Research and Design*, vol. 92, no. 11, pp. 2106–2122, 2014.
- [19] S. Sanaye and J. Mahmoudimehr, "Minimization of fuel consumption in cyclic and non-cyclic natural gas transmission networks: Assessment of genetic algorithm optimization method as an alternative to non-sequential dynamic programming," *Journal of the Taiwan Institute of Chemical Engineers*, vol. 43, no. 6, pp. 904–917, 2012.
- [20] A. K. Arya and S. Honwad, "Modeling, simulation, and optimization of a high-pressure cross-country natural gas pipeline: Application of an ant colony optimization technique," *Journal*

- of *Pipeline Systems Engineering and Practice*, vol. 7, no. 1, Article ID 04015008, 2016.
- [21] C. Luongo, B. Gilmour, and D. Schroeder, "Optimization in natural gas transmission networks: A tool to improve operational efficiency," in *Proceedings of the Third SIAM conference on optimization*, Boston, USA, 1989.
  - [22] A. Osiadacz, *Simulation and Analysis of Gas Networks*, Gulf Publisher, 1987.
  - [23] G. H. Rodriguez, L. G. Pibouleau, C. A. Pantel, and S. Domenech, "Optimization of gas transmission networks under energetic and environmental considerations," *International Journal of Chemical Reactor Engineering*, vol. 8, article no. A136, 2010.
  - [24] C. Li, W. Jia, Y. Yang, and X. Wu, "Adaptive genetic algorithm for steady-state operation optimization in natural gas networks," *Journal of Software*, vol. 6, no. 3, pp. 452–459, 2011.
  - [25] R. Alamian, M. Behbahani-Nejad, and A. Ghanbarzadeh, "A state space model for transient flow simulation in natural gas pipelines," *Journal of Natural Gas Science and Engineering*, vol. 9, pp. 51–59, 2012.
  - [26] C. A. Souza Lima Jr., C. M. F. Lapa, C. M. D. N. A. Pereira, J. J. Da Cunha, and A. C. M. Alvim, "Comparison of computational performance of GA and PSO optimization techniques when designing similar systems - Typical PWR core case," *Annals of Nuclear Energy*, vol. 38, no. 6, pp. 1339–1346, 2011.
  - [27] Z. Zheng and C. Wu, "Power optimization of gas pipelines via an improved particle swarm optimization algorithm," *Petroleum Science*, vol. 9, no. 1, pp. 89–92, 2012.
  - [28] C. K. Sun, V. Uraikul, C. W. Chan, and P. Tontiwachwuthikul, "Integrated expert system/operations research approach for the optimization of natural gas pipeline operations," *Engineering Applications of Artificial Intelligence*, vol. 13, no. 4, pp. 465–475, 2000.
  - [29] H. Üster and Ş. Dilaveroğlu, "Optimization for design and operation of natural gas transmission networks," *Applied Energy*, vol. 133, pp. 56–69, 2014.
  - [30] F. Tabkhi, L. Pibouleau, G. Hernandez-Rodriguez, C. Azzaro-Pantel, and S. Domenech, "Improving the performance of natural gas pipeline networks fuel consumption minimization problems," *AIChE Journal*, vol. 56, no. 4, pp. 946–964, 2010.
  - [31] Y. Hu, Z. Bie, T. Ding, and Y. Lin, "An NSGA-II based multi-objective optimization for combined gas and electricity network expansion planning," *Applied Energy*, vol. 167, pp. 280–293, 2016.
  - [32] J.-P. Nicot and I. J. Duncan, "Common attributes of hydraulically fractured oil and gas production and CO<sub>2</sub> geological sequestration," *Greenhouse Gases: Science and Technology*, vol. 2, no. 5, pp. 352–368, 2012.
  - [33] F. Tabkhi, *Optimization of gas transport networks*, INPT, 2007.
  - [34] S. Derafshi Beigvand, H. Abdi, and M. La Scala, "Optimal operation of multicarrier energy systems using Time Varying Acceleration Coefficient Gravitational Search Algorithm," *Energy*, vol. 114, pp. 253–265, 2016.
  - [35] Y. Collette and P. Siarry, *Optimisation multiobjectif: Algorithmes*, Editions Eyrolles, 2011.
  - [36] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: a tutorial," *Reliability Engineering & System Safety*, vol. 91, no. 9, pp. 992–1007, 2006.
  - [37] M. Schmidt, M. C. Steinbach, and B. M. Willert, "High detail stationary optimization models for gas networks," *Optimization and Engineering*, vol. 16, no. 1, pp. 131–164, 2015.
  - [38] A. Chebouba, "Multi objective optimization of line pack management of gas pipeline system," *Journal of Physics: Conference Series*, vol. 574, no. 1, Article ID 012114, 2014.
  - [39] T. Mora and M. Uliuru, "Minimization of energy use in pipeline operations - An application to natural gas transmission systems," in *Proceedings of the IECON 2005: 31st Annual Conference of IEEE Industrial Electronics Society*, pp. 2190–2197, USA, November 2005.
  - [40] X. Li, E. Armagan, A. Tomasgard, and P. I. Barton, "Stochastic pooling problem for natural gas production network design and operation under uncertainty," *AIChE Journal*, vol. 57, no. 8, pp. 2120–2135, 2011.
  - [41] M. Tawarmalani and N. V. Sahinidis, "Global optimization of mixed-integer nonlinear programs: A theoretical and computational study," *Mathematical Programming*, vol. 99, no. 3, pp. 563–591, 2004.
  - [42] A. Chebouba and H. Megloui, "A new combined artificial neural networks and ant colony optimization algorithm for energy minimization in pipeline operations," in *Proceedings of the 3rd International Conference on Industrial Engineering and Operations Management*, IEOM Forum, Istanbul, Turkey, 2012.
  - [43] D. E. Goldberg, "Computer-aided gas pipeline operation using genetic algorithms and rule learning part II: Rule learning control of a pipeline under normal and abnormal conditions," in *PSIG Annual Meeting*, PSIG, Albuquerque, USA, 1985.
  - [44] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–72, 1992.
  - [45] D. E. Goldberg, "Computer-aided pipeline operation using genetic algorithms and rule learning. PART I: Genetic algorithms in pipeline optimization," *Engineering with Computers*, vol. 3, no. 1, pp. 35–45, 1987.
  - [46] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 4, pp. 656–667, 1994.
  - [47] M. MohamadiBaghmolaei, M. Mahmoudy, D. Jafari, R. MohamadiBaghmolaei, and F. Tabkhi, "Assessing and optimization of pipeline system performance using intelligent systems," *Journal of Natural Gas Science and Engineering*, vol. 18, pp. 64–76, 2014.
  - [48] K. K. Botros, D. Sennhauser, K. J. Jungowski, G. Poissant, H. Golshan, and J. Stoffregen, "Multi-objective optimization of large pipeline networks using genetic algorithm," in *Proceedings of the 2004 International Pipeline Conference*, pp. 2005–2015, ASME, October 2004.
  - [49] K. K. Botros, D. Sennhauser, J. Stoffregen, K. J. Jungowski, and H. Golshan, "Large pipeline network optimization: Summary and conclusions of transcanada research effort," in *Proceedings of the 2006 6th International Pipeline Conference, IPC 2006*, pp. 657–670, ASME, 2006.
  - [50] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
  - [51] J. Knowles and D. Corne, "The pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation," in *Proceedings of the Congress on Evolutionary Computation (CEC '99)*, vol. 1, pp. 98–105, IEEE, July 1999.
  - [52] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*, pp. 760–766, Springer, 2010.
  - [53] L.-G. Li, Z.-H. Zhang, Y.-S. Dai, W.-F. Sun, and C.-Q. Dong, "Operation optimization based on improved pattern search

- algorithm in gas transmission networks,” *Journal of China University of Petroleum (Edition of Natural Science)*, vol. 36, no. 4, pp. 139–143, 2012.
- [54] K. Suresh, S. Ghosh, D. Kundu, A. Sen, S. Das, and A. Abraham, “Inertia-adaptive particle swarm optimizer for improved global search,” in *Proceedings of the 8th International Conference on Intelligent Systems Design and Applications, ISDA 2008*, pp. 253–258, IEEE, November 2008.
- [55] M. Zhou, Y. Zhang, and S. Jin, “Dynamic optimization of heated oil pipeline operation using PSO-DE algorithm,” *Measurement*, vol. 59, pp. 344–351, 2015.
- [56] M. R. Javadi, K. Mazlumi, and A. Jalilvand, “Application of GA, PSO and ABC in optimal design of a stand-alone hybrid system for north-west of Iran,” in *Proceedings of the 7th International Conference on Electrical and Electronics Engineering, ELECO 2011*, pp. 203–210, tur, December 2011.
- [57] A. Marzoughi, H. Selamat, and F. Marzoughi, “Application of particle swarm optimization approach to improve PID performance for a gas turbine temperature control system,” in *Proceedings of the 2010 8th IEEE Student Conference on Research and Development - Engineering: Innovation and Beyond, SCORED 2010*, pp. 224–229, IEEE, Malaysia, December 2010.
- [58] M. Dorigo, *Optimization, Learning And Natural Algorithms*, Politecnico di Milano, Italy, 1992.
- [59] M. Dorigo, M. Birattari, C. Blum, M. Clerc, T. Stützle, and A. Winfield, “Ant Colony Optimization and Swarm Intelligence,” in *Proceedings of the 6th International Conference, ANTS*, Springer, Brussels, Belgium, 2008.
- [60] A. Chebouba, F. Yalaoui, L. Amodeo, A. Smati, and A. Tairi, “New method to minimize fuel consumption of gas pipeline using ant colony optimization algorithms,” in *Proceedings of the 2006 International Conference on Service Systems and Service Management*, pp. 947–952, IEEE, France, October 2006.
- [61] P. J. M. Van Laarhoven and E. H. L. Aarts, “Simulated annealing,” in *Simulated Annealing: Theory and Applications*, pp. 7–15, Springer, Dordrecht, Netherlands, 1987.
- [62] A. Khachatryan, S. Semenovskaya, and B. Vainstein, “A statistical-thermodynamic approach to determination of structure amplitude phases,” *Sov. Phys. Crystallography*, vol. 24, no. 5, pp. 519–524, 1979.
- [63] S. Kirkpatrick, J. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *American Association for the Advancement of Science: Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [64] H. Zhang, Y. Yang, and M. Luo, “Optimization of large-scale natural gas pipeline network base on the combined GASA algorithm,” in *Proceedings of the 2010 International Conference on Computational and Information Sciences, ICCIS2010*, pp. 254–257, IEEE, Chengdu, China, December 2010.
- [65] H. Chen, C. Wu, L. Zuo, W. Li, and Y. Yu, “Applying the Simulated Annealing Algorithm to Optimize the Scheduling of Products Pipelines,” in *PSIG Annual Meeting*, PSIG, Vancouver, Canada, 2016.
- [66] I. Samora, M. J. Franca, A. J. Schleiss, and H. M. Ramos, “Simulated Annealing in Optimization of Energy Production in a Water Supply Network,” *Water Resources Management*, vol. 30, no. 4, pp. 1533–1547, 2016.
- [67] E. Malamura and T. Murata, “Hybrid system modeling and operation schedule optimization for gas transportation network based on combined method of DE, GA and hybrid petri net,” in *Proceedings of the 5th IIAI International Congress on Advanced Applied Informatics, IIAI-AAI 2016*, pp. 1032–1035, IEEE, Japan, July 2016.
- [68] W. Qiao, B. Chen, S. Wu, C. Li, J. Mao, and J. Ma, “A forecasting model of natural gas daily load based on wavelet transform and LSSVM - DE,” *Natural Gas Industry*, vol. 34, no. 9, pp. 118–124, 2014.
- [69] C. Borraz-Sánchez and R. Z. Ríos-Mercado, “Improving the operation of pipeline systems on cyclic structures by tabu search,” *Computers & Chemical Engineering*, vol. 33, no. 1, pp. 58–64, 2009.
- [70] L. Wu, L. Min, Y. Liu, Y. Xu, and Y. Xinglan, “Decision of optimal scheduling scheme for gas field pipeline network based on hybrid genetic algorithm,” in *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, pp. 369–374, ACM, China, June 2009.
- [71] M. Khademi and A. Khosravi, “Exergoeconomic analysis and optimisation of a gas-turbine power plant using PSO, GA and fuzzy logic system,” *International Journal of Exergy*, vol. 19, no. 2, pp. 259–275, 2016.
- [72] H. H. Nguyen and C. W. Chan, “Optimal scheduling of gas pipeline operation using genetic algorithms,” in *Proceedings of the Canadian Conference on Electrical and Computer Engineering 2005*, pp. 2195–2198, IEEE, Canada, May 2005.
- [73] H. H. Nguyen, V. Uraikul, C. W. Chan, and P. Tontiwachwuthikul, “A comparison of automation techniques for optimization of compressor scheduling,” *Advances in Engineering Software*, vol. 39, no. 3, pp. 178–188, 2008.
- [74] R. Kanés, A. Basha, L. N. Véchot, and M. Castier, “Simulation of venting and leaks from pressure vessels,” *Journal of Loss Prevention in the Process Industries*, vol. 40, pp. 563–577, 2016.
- [75] Y.-D. Jo and J. A. Bum, “A method of quantitative risk assessment for transmission pipeline carrying natural gas,” *Journal of Hazardous Materials*, vol. 123, no. 1-3, pp. 1–12, 2005.
- [76] Y.-T. Kao and E. Zahara, “A hybrid genetic algorithm and particle swarm optimization for multimodal functions,” *Applied Soft Computing*, vol. 8, no. 2, pp. 849–857, 2008.
- [77] T. A. El-Mihoub, A. A. Hopgood, L. Nolle, and A. Battersby, “Hybrid genetic algorithms: a review,” *Engineering Letters*, vol. 13, no. 2, pp. 124–137, 2006.
- [78] D. Marqués and M. Morari, “On-line optimization of gas pipeline networks,” *Automatica*, vol. 24, no. 4, pp. 455–469, 1988.
- [79] D. Mahlke, A. Martin, and S. Moritz, “A simulated annealing algorithm for transient optimization in gas networks,” *Mathematical Methods of Operations Research*, vol. 66, no. 1, pp. 99–115, 2007.
- [80] M. Geidl and G. Andersson, “Optimal power flow of multiple energy carriers,” *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 145–155, 2007.

## Research Article

# Grey Relational Bidirectional Projection Method for Multicriteria Decision Making with Hesitant Intuitionistic Fuzzy Linguistic Information

Yuqi Zang , Xiaodong Zhao, Shiyong Li , and Adnan Nazir

*School of Economics and Management, Yanshan University, Qinhuangdao 066004, China*

Correspondence should be addressed to Shiyong Li; shiyongli@ysu.edu.cn

Received 6 December 2017; Accepted 28 February 2018; Published 4 April 2018

Academic Editor: Guillermo Cabrera-Guerrero

Copyright © 2018 Yuqi Zang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We investigate a novel approach for multicriteria decision making (MCDM) with hesitant intuitionistic fuzzy linguistic information. To compare the hesitant intuitionistic fuzzy linguistic term sets (HIFLTSs), we propose a comparison method of HIFLTSs. A family of distance measures of HIFLTSs is developed. After that, we propose the grey relational bidirectional projection method based on the proposed comparison method and distance measures of IVHFLNs for dealing with MCDM problems. Furthermore, we establish a nonlinear optimization model to obtain the weight vector of criteria. Finally, an illustrative example is given to demonstrate the effectiveness and flexibility of the proposed approach.

## 1. Introduction

Since 1975, Zadeh [1] first proposed the fuzzy linguistic term set which can depict the qualitative fuzzy information. It has become the beginning of qualitative decision making research. Qualitative decision making has been widely concerned by scholars and successfully applied to many areas [2–10]. However, in many practical qualitative decision making cases, it is often difficult for decision makers to express preferences by using a single linguistic term. The hesitant fuzzy linguistic term set (HFLTS) was proposed by Rodriguez et al. [11], which permits decision makers to use several possible linguistic terms to express preferences. Because the hesitant fuzzy linguistic term set can adequately express vague and imprecise information, more close to the decision maker's qualitative thinking cognition, which is regarded as a powerful tool for dealing with qualitative decision making problems, scholars had an in-depth study of hesitant fuzzy linguistic term sets and made some improvements.

In the aspect of the comparison method, Rodriguez et al. [11] first proposed the possibility degree for comparing HFLTSs; the definition of the possibility degree is based on the interval value which is constructed by the HFLTSs' envelopes. Wei et al. [12] pointed out that Rodriguez et

al.'s method may not accord with common sense, and they constructed the new suitable possibility degree formula of HFLTSs by using the possibility degree theory. Lee and Chen [13] noted that the Rodriguez et al.'s and Wei et al.'s methods were defined by using the maximum and minimum operators, which can not rank the preference orders in some cases. Thus, they proposed a novel comparison method of HFLTSs based on the likelihood. Tian et al. [14] also from the perspective of the likelihood proposed a qualitative flexible multiple criteria method, which can measure the consistency and inconsistency of the preference order under the hesitant fuzzy linguistic environment.

As to the distance measure, Liao et al. [15] defined a family of basic distance measures of HFLTs, then they proposed the distance measures between two collections of HFLTs in continuous and discrete cases and applied the measure method to the evaluation of the quality of movies. Liao and Xu [16] proposed a series of cosine distance measures of HFLTs from the geometric point of view, then the proposed measures were applied to the selection of ERP systems. Wang et al. [17] presented the hesitant fuzzy linguistic Hausdorff distance measure, which is not necessary to add and rearrange any linguistic terms in HFLTs and applied the distance measure to

the TOPSIS and TODIM methods. Wang et al. [18] proposed the Euclidean distance measure based on the function of the position index for HFLTSS. Meng and Chen [19] considered the ordered positions and the internal interactions, developed the generalized hesitant fuzzy linguistic 2-additive Choquet weighted distance measure and the generalized hesitant fuzzy linguistic 2-additive Shapley Choquet weighted distance measure, and then applied the proposed measure to evaluate investment.

There are also some extensions of classical MCDM methods. Liao and Xu [16] explored the TOPSIS and VIKOR methods based on the cosine distance measure for solving the hesitant fuzzy linguistic multicriteria decision making. Liao et al. [20] used the VIKOR method for solving the hesitant fuzzy linguistic multicriteria decision making in which the criteria conflict with each other. Wei et al. [21] considered the decision maker's psychological behavior in the MCDM and proposed a hesitant fuzzy linguistic TODIM method. They applied the proposed method to evaluate the telecommunications service providers. Wang et al. [18] proposed the ELECTRE I method to deal with the MCDM problems based on the distance measure for HFLTSS. Liao et al. [22] defined the correlation coefficient of HFLTSS, then the traditional Chinese medical diagnosis was given to illustrate the applicability of the proposed method. Lin et al. [23] derived a family of hesitant fuzzy linguistic aggregation operators and applied the proposed operators to multiattribute decision making. Farhadinia [24] solved the hesitant fuzzy linguistic multiple criteria decision making problems with completely unknown weights by entropy measure; Gou et al. [25] proposed a hesitant fuzzy linguistic alternative queuing method based on the entropy and cross-entropy measures for the hesitant fuzzy linguistic term sets and applied the proposed measure to the tertiary hospital management.

However, while using the hesitant fuzzy linguistic term sets, only the linguistic terms in the membership function are used to express the degree of certainty of the property, and the importance of the uncertainty is ignored. In this case, Beg and Rashid [26] proposed the hesitant intuitionistic fuzzy linguistic term sets (HIFLTSS) which take into account both membership and nonmembership. The hesitant intuitionistic fuzzy linguistic term sets can depict the hesitation more comprehensively when faced with qualitative decision making problems. They used the TOPSIS method to deal with the hesitant intuitionistic fuzzy linguistic multicriteria decision making problem. Liu et al. [27] extended the WA and OWA operators to the hesitant intuitionistic fuzzy linguistic environment and developed some hesitant intuitionistic fuzzy linguistic aggregation operators and applied the HIFLWA operator to the MCDM. Rashid et al. [28] constructed an ELECTRE-based outranking method to deal with MCDM. Faizi et al. [29] proposed an outranking method for hesitant intuitionistic fuzzy linguistic group decision making based on the support function, the risk function, and the credibility function.

As mentioned previously, since HIFLTS was proposed in 2014, the study of MCDM methods based on hesitant intuitionistic fuzzy linguistic information is still at the initial stage, only a few studies are involved, and there is still much

work we need to do for improving the research. First of all, the possibility degree for comparing HIFLTSS has not been studied. With respect to the shortcomings of the existing possibility degree of HFLTSS [11–13], we propose an improved probability degree of HFLTSS, then on the basis of the above, we define the possibility degree of HIFLTSS. This is the first motivation of our work. Then, on the study of distance measure, Beg and Rashid [26] defined the distance measure of HIFLTSS; however, the proposed distance measure was based on the envelope of HIFLTSS. If two HIFLTSS have the same envelope, they will obtain the same distance, which is unreasonable. Rashid et al. [28] only considered the case that the number of elements in the membership degree and nonmembership degree is equal. The distance measure should be improved to overcome the drawback of the existing distance measures. This is the second motivation of our work. In addition, there are many famous methods for solving the MCDM problems with hesitant fuzzy linguistic information or hesitant intuitionistic fuzzy linguistic information, but none of studies have used grey relational projection method to handle the MCDM problems with hesitant fuzzy linguistic information or hesitant intuitionistic fuzzy linguistic information. Grey relational projection technology is one of the effective methods to deal with MCDM problems [30–32]. We improved the traditional grey relational projection method and proposed the grey relational bidirectional projection method to deal with the MCDM problems under the hesitant intuitionistic fuzzy linguistic environment. This is the third motivation of our study.

The rest of this paper is organized as follows: In Section 2, we present some definitions of the LTSs, HFLTSS, and HIFLTSS. In Section 3, we propose the comparison method and the distance measures of HFLTSS. Section 4 denotes studying the hesitant intuitionistic fuzzy linguistic MCDM based on the grey relational bidirectional projection. In Section 5, we present a numerical example to demonstrate the effectiveness and practicality of the proposed method and also discuss the advantages of the proposed method. In Section 6, we briefly conclude the paper.

## 2. Preliminaries

**2.1. Linguistic Term Sets.** Let  $S = \{s_i \mid i = 0, 1, \dots, g\}$  be a finite linguistic term set with odd cardinality, where  $s_i$  represents a possible value for a linguistic variable, and the following characteristics should be satisfied [33]:

- (1) The set is ordered:  $s_\alpha \geq s_\beta \Leftrightarrow \alpha \geq \beta$ .
- (2) There is a negation operator:  $\text{neg}(s_\alpha) = s_{g-\alpha}$ .
- (3) If  $s_\alpha > s_\beta$ , then  $\max\{s_\alpha, s_\beta\} = s_\alpha$ ,  $\min\{s_\alpha, s_\beta\} = s_\beta$ .

### 2.2. Hesitant Fuzzy Linguistic Term Sets

**Definition 1** (see [11]). Let  $S = \{s_0, s_1, \dots, s_g\}$  be a linguistic term set; let  $H_S$  be an HFLTSS which is an ordered finite subset of the consecutive linguistic terms of  $S$ .

*Definition 2* (see [11]). Let  $S = \{s_0, s_1, \dots, s_g\}$  be a linguistic term set, let  $H_S$  be an HFLTS, and the operational laws are defined as

- (1) the upper bound:  $H_{S^+} = \max(s_i) = s_j, s_i \in H_S$  and  $s_i \leq s_j \forall i$ ;
- (2) the lower bound:  $H_{S^-} = \min(s_i) = s_j, s_i \in H_S$  and  $s_i \geq s_j \forall i$ ;
- (3) the envelope:  $\text{env}(H_S) = [H_{S^-}, H_{S^+}]$ ;
- (4) the complement:  $H_S^c = S - H_S = \{s_i, s_i \in S \text{ and } s_i \notin H_S\}$ .

### 2.3. Hesitant Intuitionistic Fuzzy Linguistic Term Sets

*Definition 3* (see [26]). Let  $X$  be fixed; HIFLTSs on  $X$  are functions  $H(x)$  and  $G(x)$  that when applied to  $X$  return ordered finite subsets of the consecutive linguistic term set,  $S = \{s_0, s_1, \dots, s_g\}$ ; the mathematical symbol is defined as

$$E_S = \{(x, H_S(x), G_S(x)) \mid x \in X\}, \quad (1)$$

where  $H(x)$  and  $G(x)$  are the subsets of the consecutive linguistic terms of  $S$ , denoting the possible membership degrees and nonmembership degrees of the element  $x \in X$  to the set  $E_S$ , respectively, with the conditions:  $\max(H_S(x)) + \min(G_S(x)) \leq s_g$ ;  $\min(H_S(x)) + \max(G_S(x)) \leq s_0$ .

For convenience, the pair  $(H_S(x), G_S(x))$  is called the hesitant intuitionistic fuzzy linguistic term element (HIFLTE), denoted as  $(H_S, G_S)$ .

*Definition 4* (see [26]). Let  $E_S = (H_S, G_S)$  be an HIFLTS; the upper bound and lower bound are defined as

$$\begin{aligned} H_{S^+} &= \max(s_i) = s_j, \quad s_i \in H_S, \quad s_i \leq s_j \quad \forall i; \\ H_{S^-} &= \min(s_i) = s_j, \quad s_i \in H_S, \quad s_i \geq s_j \quad \forall i; \\ G_{S^+} &= \max(s_i) = s_j, \quad s_i \in G_S, \quad s_i \leq s_j \quad \forall i; \\ G_{S^-} &= \min(s_i) = s_j, \quad s_i \in G_S, \quad s_i \geq s_j \quad \forall i. \end{aligned} \quad (2)$$

*Definition 5* (see [26]). The envelope of the HIFLTS,  $\text{env}(E_S)$ , is defined as

$$\text{env}(E_S) = \{[H_{S^-}, H_{S^+}], [G_{S^-}, G_{S^+}]\}. \quad (3)$$

*Definition 6*. The complement of HIFLTS,  $E_S$ , is defined as

$$\begin{aligned} E_S^c &= \{(S - H_S), (S - G_S)\} \\ &= \{(s_i, s_i \in S, s_i \notin H_S), (s_i, s_i \in S, s_i \notin G_S)\}. \end{aligned} \quad (4)$$

## 3. A Comparison Method and Distance Measures of HIFLTSs

*3.1. A Comparison Method of HIFLTSs.* Possibility degree is one of the most appropriate comparison methods to rank the preference order of different arguments. We have already pointed out in Introduction that some scholars have

studied the possibility formula of HFLTSs; however, they have the drawback that they cannot compare HFLTSs in some cases. For instance, when the hesitant fuzzy linguistic term sets reduce to only one element, the possibility formula denominator is zero by using Lee and Chen's method [13]; obviously, that is unreasonable. Wei et al.'s method [12] may be a complex formula, but they gave us a good inspiration to develop the possibility degree of HIFLTSs.

In this paper, we combine the ideas of methods in [12, 13] and propose the improved possibility formula of HIFLTSs, which can rank the preference order more efficiently. The possibility degree formula of HIFLTSs is given as

$$\begin{aligned} P(H_S^1 \geq H_S^2) &= \max \left\{ 1 \right. \\ &\quad \left. - \max \left( \frac{\text{Ind}(H_S^{2+}) - \text{Ind}(H_S^{1-}) + 1}{\#H_S^1 + \#H_S^2}, 0 \right), 0 \right\}, \end{aligned} \quad (5)$$

where  $\#H_S^1, \#H_S^2$  are the number of elements in  $H_S^1, H_S^2$ .  $\text{Ind}(s_i) = i$  (it provides the index associated with the label).

As with the possibility degree axiom of the interval number, the possibility degree of  $p(H_S^1 \geq H_S^2)$  satisfies the following properties:

- (1)  $0 \leq p(H_S^1 \geq H_S^2) \leq 1$ ;
- (2)  $p(H_S^1 \geq H_S^1) = 0.5$ ;
- (3) if  $H_S^1 \geq H_S^2$ , then  $p(H_S^1 \geq H_S^2) = 1$ ;  
if  $H_S^1 \leq H_S^2$ , then  $p(H_S^1 \geq H_S^2) = 0$ ;
- (4)  $p(H_S^1 \geq H_S^2) + p(H_S^2 \geq H_S^1) = 1$ ;
- (5) if  $H_S^1 = H_S^2$ , then  $p(H_S^1 \geq H_S^2) = p(H_S^2 \geq H_S^1) = 0.5$ .

Similar to the definition of the possibility degree for HFLTS, we give the definition for HIFLTSs.

*Definition 7.* Let  $S = \{s_0, s_1, \dots, s_g\}$  be a linguistic term set, let  $E_S^1 = (H_S^1, G_S^1)$  and  $E_S^2 = (H_S^2, G_S^2)$  be two HIFLTSs on  $S$ , and the possibility degree of  $p(E_S^1 \geq E_S^2)$  is denoted as

$$\begin{aligned} p(E_S^1 \geq E_S^2) &= \frac{1}{2} \left( p(H_S^1 \geq H_S^2) + p(G_S^2 \geq G_S^1) \right) \\ &= \frac{1}{2} \left( \max \left\{ 1 \right. \right. \\ &\quad \left. \left. - \max \left( \frac{\text{Ind}(H_S^{2+}) - \text{Ind}(H_S^{1-}) + 1}{\#H_S^1 + \#H_S^2}, 0 \right), 0 \right\} \right. \\ &\quad \left. + \max \left\{ 1 \right. \right. \\ &\quad \left. \left. - \max \left( \frac{\text{Ind}(G_S^{1+}) - \text{Ind}(G_S^{2-}) + 1}{\#G_S^1 + \#G_S^2}, 0 \right), 0 \right\} \right), \end{aligned} \quad (6)$$

where  $\#H_S^1, \#H_S^2$  are the number of elements in  $H_S^1, H_S^2$  and  $\#G_S^1, \#G_S^2$  are the number of elements in  $G_S^1, G_S^2$ , respectively.  $\text{Ind}(s_i) = i$  (it provides the index associated with the label).

The possibility degree of  $p(E_S^1 \geq E_S^2)$  satisfies the following properties:

- (1)  $0 \leq p(E_S^1 \geq E_S^2) \leq 1$ ;
- (2)  $p(E_S^1 \geq E_S^1) = 0.5$ ;
- (3) if  $E_S^1 \geq E_S^2$ , then  $p(E_S^1 \geq E_S^2) = 1$ ;  
if  $E_S^1 \leq E_S^2$ , then  $p(E_S^1 \geq E_S^2) = 0$ ;
- (4)  $p(E_S^1 \geq E_S^2) + p(E_S^2 \geq E_S^1) = 1$ ;
- (5) if  $E_S^1 = E_S^2$ , then  $p(E_S^1 \geq E_S^2) = p(E_S^2 \geq E_S^1) = 0.5$ .

**3.2. Distance Measure of HIFLTSs.** Xu [34] first proposed the distance measure of the linguistic term sets as follows.

*Definition 8* (see [34]). Let  $S = \{s_0, s_1, \dots, s_g\}$  be a linguistic term set, let  $s_\alpha, s_\beta$  be two linguistic terms, and then the deviation degree between  $s_\alpha$  and  $s_\beta$  is

$$d(s_\alpha, s_\beta) = \frac{\alpha - \beta}{g + 1}, \quad (7)$$

where  $g + 1$  is the number of linguistic terms in the set  $S$ .

Motivated by the definition of the distance measure for LTSs, we define the distance measure of HIFLTSs  $E_S^1 = (H_S^1, G_S^1)$  and  $E_S^2 = (H_S^2, G_S^2)$ .

$$\begin{aligned} E_S^1 &= (H_S^1, G_S^1) = \left( \bigcup_{\sigma_l^1 \in H_S^1} \{s_{\sigma_l^1} \mid l = 1, \dots, \#H_S^1\}, \right. \\ &\quad \left. \bigcup_{\delta_k^1 \in G_S^1} \{s_{\delta_k^1} \mid k = 1, \dots, \#G_S^1\} \right), \\ E_S^2 &= (H_S^2, G_S^2) = \left( \bigcup_{\sigma_l^2 \in H_S^2} \{s_{\sigma_l^2} \mid l = 1, \dots, \#H_S^2\}, \right. \\ &\quad \left. \bigcup_{\delta_k^2 \in G_S^2} \{s_{\delta_k^2} \mid k = 1, \dots, \#G_S^2\} \right), \end{aligned} \quad (8)$$

where  $\#H_S^1, \#H_S^2$  are the number of linguistic terms in  $H_S^1, H_S^2$  and  $\#G_S^1, \#G_S^2$  are the number of linguistic terms in  $G_S^1, G_S^2$ , respectively. Where  $\#H_S^1 = \#H_S^2 = L$  and  $\#G_S^1 = \#G_S^2 = K$ , we need to make them equivalently by adding some linguistic terms to the shorter HIFLTS, according to the following principles: pessimistic principle, the lower bound will be added; optimistic principle, the upper bound will be added.

The distance measures for HIFLTSs  $E_S^1 = (H_S^1, G_S^1)$ ,  $E_S^2 = (H_S^2, G_S^2)$  are defined as follows.

The hesitant intuitionistic fuzzy linguistic Hamming distance is as follows:

$$\begin{aligned} d_{\text{hifhd}}(E_S^1, E_S^2) &= \frac{1}{2} \left( \frac{1}{L} \sum_{l=1}^L \frac{|\sigma_l^1 - \sigma_l^2|}{g+1} + \frac{1}{K} \sum_{k=1}^K \frac{|\delta_k^1 - \delta_k^2|}{g+1} \right). \end{aligned} \quad (9)$$

The hesitant intuitionistic fuzzy linguistic Euclidean distance is as follows:

$$\begin{aligned} d_{\text{hifed}}(E_S^1, E_S^2) &= \left( \frac{1}{2} \left( \frac{1}{L} \sum_{l=1}^L \left( \frac{|\sigma_l^1 - \sigma_l^2|}{g+1} \right)^2 \right. \right. \\ &\quad \left. \left. + \frac{1}{K} \sum_{k=1}^K \left( \frac{|\delta_k^1 - \delta_k^2|}{g+1} \right)^2 \right) \right)^{1/2}. \end{aligned} \quad (10)$$

With the generalization of (9) and (10), the generalized hesitant intuitionistic fuzzy linguistic distance can be obtained:

$$\begin{aligned} d_{\text{ghifld}}(E_S^1, E_S^2) &= \left( \frac{1}{2} \left( \frac{1}{L} \sum_{l=1}^L \left( \frac{|\sigma_l^1 - \sigma_l^2|}{g+1} \right)^\lambda \right. \right. \\ &\quad \left. \left. + \frac{1}{K} \sum_{k=1}^K \left( \frac{|\delta_k^1 - \delta_k^2|}{g+1} \right)^\lambda \right) \right)^{1/\lambda}. \end{aligned} \quad (11)$$

Based on the Hausdorff distance measure, the generalized hesitant intuitionistic fuzzy linguistic Hausdorff distance is expressed as

$$\begin{aligned} d_{\text{ghighld}}(E_S^1, E_S^2) &= \left( \max \left\{ \max_l \left( \frac{|\sigma_l^1 - \sigma_l^2|}{g+1} \right)^\lambda, \right. \right. \\ &\quad \left. \left. \max_k \left( \frac{|\delta_k^1 - \delta_k^2|}{g+1} \right)^\lambda \right\} \right)^{1/\lambda}. \end{aligned} \quad (12)$$

With the combination of (11) and (12), we get the generalized hybrid hesitant intuitionistic fuzzy linguistic distance:

$$\begin{aligned} d_{\text{ghighld}}(E_S^1, E_S^2) &= \left( \frac{1}{2} \left( \frac{1}{L} \sum_{l=1}^L \left( \frac{|\sigma_l^1 - \sigma_l^2|}{g+1} \right)^\lambda \right. \right. \\ &\quad \left. \left. + \frac{1}{K} \sum_{k=1}^K \left( \frac{|\delta_k^1 - \delta_k^2|}{g+1} \right)^\lambda \right) \right. \\ &\quad \left. + \max \left\{ \max_l \left( \frac{|\sigma_l^1 - \sigma_l^2|}{g+1} \right)^\lambda, \max_k \left( \frac{|\delta_k^1 - \delta_k^2|}{g+1} \right)^\lambda \right\} \right)^{1/\lambda}, \end{aligned} \quad (13)$$

where  $\sigma_l^1$  and  $\sigma_l^2$  are the  $l$ th largest linguistic term in  $H_S^1$  and  $H_S^2$  and  $\delta_k^1$  and  $\delta_k^2$  are the  $k$ th largest linguistic term in  $G_S^1$  and  $G_S^2$ .

**Definition 9.** The distance measure  $d(E_S^1, E_S^2)$  between  $E_S^1$  and  $E_S^2$  satisfies the following properties:

- (1)  $0 \leq d(E_S^1, E_S^2) \leq 1$ ;
- (2)  $d(E_S^1, E_S^2) = 0$  if and only if  $E_S^1 = E_S^2$ ;
- (3)  $d(E_S^1, E_S^2) = d(E_S^2, E_S^1)$ .

#### 4. Proposed Method for MCDM

The MCDM problem with hesitant intuitionistic fuzzy linguistic information is shown as follows. Suppose that there are  $m$  alternatives  $A = \{A_1, A_2, \dots, A_m\}$  and  $n$  criteria  $C = \{C_1, C_2, \dots, C_n\}$ ; the weight vector of the criteria is  $W = (w_1, w_2, \dots, w_n)$ , where  $w_j \geq 0$ ,  $\sum_{j=1}^n w_j = 1$ ; assume that the evaluation values are taken in the form of HIFLTSSs, where  $E_{ij}$  is the evaluation value of alternative  $A_i$  respect to criteria  $C_j$ . Hence, we construct the hesitant intuitionistic fuzzy linguistic decision matrix  $E = [E_{ij}]_{m \times n}$ .

*Note.* This paper omits the process of transforming the linguistic information into HIFLTSSs. The detailed process was shown in [11].

**4.1. Normalized Decision Matrix.** We should translate the decision matrix  $E = [E_{ij}]_{m \times n}$  into the normalized decision matrix  $\tilde{E} = [\tilde{E}_{ij}]_{m \times n}$  before calculating the grey relational bidirectional projection. In the MCDM system, there are usually two types of criteria, the benefit criteria and the cost criteria; the normalized  $\tilde{E}_{ij}$  is shown as

$$\tilde{E}_{ij} = \begin{cases} E_{ij}, & C_j \in \text{benefit criteria} \\ E_{ij}^c, & C_j \in \text{cost criteria.} \end{cases} \quad (14)$$

**4.2. Grey Relational Bidirectional Projection.** Grey relational projection method combines the advantages of the grey relational analysis [35–40] and the projection method [41–44], which is an effective method to deal with MCDM problems. However, the traditional projection method sometimes has some shortcomings, for example, during the projection of two vectors  $a$  and  $c$  onto the ideal solution  $b$  at the same vertical point, as shown in Figure 1. Their projection value is equal; that is, the alternatives can not be compared.

Ye [45] proposed the bidirectional projection method, which can overcome the shortcomings of the traditional projection method; that is,

$$\begin{aligned} \text{BP}(a, b) &= \frac{1}{1 + |(a \cdot b) / |a| - (a \cdot b) / |b|} \\ &= \frac{|a| \cdot |b|}{|a| \cdot |b| + ||a| - |b|| \cdot a \cdot b}. \end{aligned} \quad (15)$$

Motivated by Ye's method, we combine the grey relational analysis and the bidirectional projection method and propose the grey relational bidirectional projection method, which can rank the preference order of alternatives efficiently. Then, we present the grey relational bidirectional projection

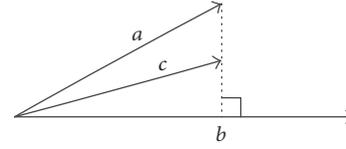


FIGURE 1: The projection vector.

method and apply it to the hesitant intuitionistic fuzzy linguistic MCDM.

Firstly, determine the hesitant intuitionistic fuzzy linguistic ideal solution. For a normalized hesitant intuitionistic fuzzy linguistic decision making matrix, the hesitant intuitionistic fuzzy linguistic positive ideal solution (HIFLPIS) and the hesitant intuitionistic fuzzy linguistic negative ideal solution (HIFLNIS) are expressed as

$$\begin{aligned} \tilde{E}^+ &= \{\tilde{E}_1^+, \tilde{E}_2^+, \dots, \tilde{E}_n^+\} \\ \tilde{E}^- &= \{\tilde{E}_1^-, \tilde{E}_2^-, \dots, \tilde{E}_n^-\}. \end{aligned} \quad (16)$$

Then calculate the generalized hybrid hesitant intuitionistic fuzzy linguistic distance between  $\tilde{E}_{ij}$  and  $\tilde{E}^+$ , and that between  $\tilde{E}_{ij}$  and  $\tilde{E}^-$  is shown as

$$\begin{aligned} d_{ij}^+ &= \left( \frac{1}{2} \left( \frac{1}{L} \sum_{l=1}^L \left( \frac{|\sigma_l^{ij} - \sigma_l^+|}{g+1} \right)^\lambda + \frac{1}{K} \sum_{k=1}^K \left( \frac{|\delta_k^{ij} - \delta_k^+|}{g+1} \right)^\lambda \right) \right. \\ &\quad \left. + \max \left\{ \max_l \left( \frac{|\sigma_l^{ij} - \sigma_l^+|}{g+1} \right)^\lambda, \max_k \left( \frac{|\delta_k^{ij} - \delta_k^+|}{g+1} \right)^\lambda \right\} \right)^{1/\lambda}, \\ d_{ij}^- &= \left( \frac{1}{2} \left( \frac{1}{L} \sum_{l=1}^L \left( \frac{|\sigma_l^{ij} - \sigma_l^-|}{g+1} \right)^\lambda + \frac{1}{K} \sum_{k=1}^K \left( \frac{|\delta_k^{ij} - \delta_k^-|}{g+1} \right)^\lambda \right) \right. \\ &\quad \left. + \max \left\{ \max_l \left( \frac{|\sigma_l^{ij} - \sigma_l^-|}{g+1} \right)^\lambda, \max_k \left( \frac{|\delta_k^{ij} - \delta_k^-|}{g+1} \right)^\lambda \right\} \right)^{1/\lambda}. \end{aligned} \quad (17)$$

Thus, the grey relational coefficient of each alternative from the HIFLPIS and HIFLNIS can be formulated as

$$\begin{aligned} r_{ij}^+ &= \frac{\min_i \min_j d_{ij}^+ + \theta \max_i \max_j d_{ij}^+}{d_{ij}^+ + \theta \max_i \max_j d_{ij}^+}, \\ r_{ij}^- &= \frac{\min_i \min_j d_{ij}^- + \theta \max_i \max_j d_{ij}^-}{d_{ij}^- + \theta \max_i \max_j d_{ij}^-}, \end{aligned} \quad (18)$$

where  $\theta \in [0, 1]$  is the resolution coefficient which is decided by the decision maker; thus we can construct the grey relational coefficient matrices  $r^+ = [r_{ij}^+]_{m \times n}$  and  $r^- = [r_{ij}^-]_{m \times n}$ .

The grey relational coefficient between HIFLPIS and HIFLPIS and that between HIFLNIS and HIFLNIS are, respectively, shown as

$$\begin{aligned} r_0^+ &= (r_{01}^+, r_{02}^+, \dots, r_{0n}^+) = \frac{1, 1, \dots, 1}{n}, \\ r_0^- &= (r_{01}^-, r_{02}^-, \dots, r_{0n}^-) = \frac{1, 1, \dots, 1}{n}. \end{aligned} \quad (19)$$

We can get the weighted grey relational coefficient matrices  $R^+ = [R_{ij}^+]_{m \times n} = [w_j r_{ij}^+]_{m \times n}$  and  $R^- = [R_{ij}^-]_{m \times n} = [w_j r_{ij}^-]_{m \times n}$ .

The weighted grey relational coefficient between HIFLPIS and HIFLPIS and that between HIFLNIS and HIFLNIS are, respectively, shown as

$$\begin{aligned} R_0^+ &= (R_{01}^+, R_{02}^+, \dots, R_{0n}^+) = (w_1, w_2, \dots, w_n), \\ R_0^- &= (R_{01}^-, R_{02}^-, \dots, R_{0n}^-) = (w_1, w_2, \dots, w_n). \end{aligned} \quad (20)$$

Combining the weighted grey relational coefficient and the bidirectional projection method, we get the weighted grey relational bidirectional projection between  $R_i^+$  and  $R_0^+$ , and that between  $R_i^-$  and  $R_0^-$  is, respectively, shown as

$$\begin{aligned} BP_i^+ &= \frac{1}{1 + |(R_i^+ \cdot R_0^+) / |R_i^+| - (R_i^+ \cdot R_0^+) / |R_0^+||} = \frac{|R_i^+| \cdot |R_0^+|}{|R_i^+| \cdot |R_0^+| + ||R_i^+| - |R_0^+|| \cdot R_i^+ \cdot R_0^+} \\ &= \frac{\sqrt{\sum_{j=1}^n (R_{ij}^+)^2} \cdot \sqrt{\sum_{j=1}^n w_j^2}}{\sqrt{\sum_{j=1}^n (R_{ij}^+)^2} \cdot \sqrt{\sum_{j=1}^n w_j^2} + \left| \sqrt{\sum_{j=1}^n (R_{ij}^+)^2} - \sqrt{\sum_{j=1}^n w_j^2} \right| \cdot \sum_{j=1}^n R_{ij}^+ w_j}, \\ BP_i^- &= \frac{1}{1 + |(R_i^- \cdot R_0^-) / |R_i^-| - (R_i^- \cdot R_0^-) / |R_0^-||} = \frac{|R_i^-| \cdot |R_0^-|}{|R_i^-| \cdot |R_0^-| + ||R_i^-| - |R_0^-|| \cdot R_i^- \cdot R_0^-} \\ &= \frac{\sqrt{\sum_{j=1}^n (R_{ij}^-)^2} \cdot \sqrt{\sum_{j=1}^n w_j^2}}{\sqrt{\sum_{j=1}^n (R_{ij}^-)^2} \cdot \sqrt{\sum_{j=1}^n w_j^2} + \left| \sqrt{\sum_{j=1}^n (R_{ij}^-)^2} - \sqrt{\sum_{j=1}^n w_j^2} \right| \cdot \sum_{j=1}^n R_{ij}^- w_j}. \end{aligned} \quad (21)$$

For the alternative  $A_i$ , the closer the value of  $BP_i^+$  is to 1, the closer it is to HIFLPIS, the closer the value of  $BP_i^-$  is to 1, the closer it is to HIFLNIS, obviously, the larger the value of  $BP_i^+$  is, the better  $A_i$  is, and the smaller the value of  $BP_i^-$  is, the better  $A_i$  is and vice versa. Thus we construct the relative closeness formula.

$$\tilde{C}_i = \frac{BP_i^+}{BP_i^+ + BP_i^-}. \quad (22)$$

In general, the larger  $\tilde{C}_i$  is, the better  $A_i$  is and vice versa.

**4.3. Determine the Attribute Weight.** In this paper we consider the case where the criteria weight information is partly known. As we know, there are many methods for deriving criteria weights, such as the deviation-based method [46, 47] and the entropy-based method [48, 49]. We combine the advantages of the deviation-based method and the entropy-based method and construct the combined optimization model.

We first construct the optimization model  $M1$  according to the minimum deviation method. The grey relational coefficient deviation between the alternative  $A_i$  and HIFLPIS is  $(1 - r_{ij}^+)$ , to eliminate the effects of symbols, we take the form of the squared, and totally, we have

$$\begin{aligned} M1: \min \quad C_1(w) &= \sum_{i=1}^m \sum_{j=1}^n [(1 - r_{ij}^+) w_j]^2 \\ \text{s.t.} \quad 0 &\leq w_j^L \leq w_j \leq w_j^U \leq 1 \\ &\sum_{j=1}^n w_j = 1. \end{aligned} \quad (23)$$

The principle of information entropy method is that, in all feasible solutions or possible solutions, the maximum entropy is chosen. The maximum entropy means that the amount of information obtained is the smallest. In the process of solving, the amount of information added is the least, so the entropy-based method is reasonable when the criteria weight information is partly known. We construct the optimization model  $M2$  as follows:

$$\begin{aligned} M2: \max \quad C_2(w) &= -\sum_{j=1}^n w_j \ln w_j \\ \text{s.t.} \quad 0 &\leq w_j^L \leq w_j \leq w_j^U \leq 1 \\ &\sum_{j=1}^n w_j = 1. \end{aligned} \quad (24)$$

Combine  $M1$  and  $M2$ , then we have optimization model  $M3$ :

$$\begin{aligned}
 M3: \min \quad & C(w) \\
 & = \alpha \sum_{i=1}^m \sum_{j=1}^n [(1 - r_{ij}^+) w_j]^2 \\
 & \quad - (1 - \alpha) \sum_{j=1}^n w_j \ln w_j \quad (25) \\
 \text{s.t.} \quad & 0 \leq w_j^L \leq w_j \leq w_j^U \leq 1 \\
 & \sum_{j=1}^n w_j = 1,
 \end{aligned}$$

where  $\alpha$  represents the equilibrium coefficient; without losing generality, we let  $\alpha = 0.5$ . The criteria weights can be obtained by Matlab software.

**4.4. The Procedure of the MCDM Method.** The algorithm for the proposed method is shown as follows.

*Step 1.* Construct the decision matrix  $E = [E_{ij}]_{m \times n}$ , then translate the decision matrix into the normalized decision matrix  $\tilde{E} = [\tilde{E}_{ij}]_{m \times n}$ .

*Step 2.* Determine the hesitant intuitionistic fuzzy linguistic positive ideal solution (HIFLPIS)  $\tilde{E}^+$  and the hesitant intuitionistic fuzzy linguistic negative ideal solution (HIFLNIS)  $\tilde{E}^-$ , according to Definition 7.

*Step 3.* Calculate the generalized hybrid hesitant intuitionistic fuzzy linguistic distance between  $\tilde{E}_{ij}$  and  $\tilde{E}^+$  and that between  $\tilde{E}_{ij}$  and  $\tilde{E}^-$ , shown in (17), based on the distance measure. We get the grey relational coefficient of each alternative from the HIFLPIS and HIFLNIS, according to (18), then the grey relational coefficient matrices  $r^+ = [r_{ij}^+]_{m \times n}$  and  $r^- = [r_{ij}^-]_{m \times n}$  are constructed.

*Step 4.* Obtain the criteria weight by solving the optimization model  $M3$ .

*Step 5.* Calculate the grey relational bidirectional projection between each alternative and the HIFLPIS and that between each alternative and the HIFLNIS, according to (21); thus, the relative closeness can be constructed in (22).

*Step 6.* Rank all alternatives according to the relative closeness.

### 5. Illustrative Example

**5.1. Example.** In this section, an illustrative example about courses evaluation for a MCDM problem adopted from [28] is given to show the method proposed in this paper. There are three courses we need to evaluate,  $A_1, A_2$ , and  $A_3$ , and four criteria,  $C_1, C_2, C_3$ , and  $C_4$ . The criteria weights information is partly known, assuming  $0.38 \leq w_1 \leq 0.42, 0.30 \leq w_2 \leq 0.35, 0.18 \leq w_3 \leq 0.21$ , and  $0.07 \leq w_4 \leq 0.09$ . The linguistic term set  $S = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6\} = \{s_0 = \text{very poor}, s_1 = \text{poor}, s_2 = \text{medium}, s_3 = \text{fair}, s_4 = \text{medium good}, s_5 = \text{good}, s_6 = \text{very good}\}$ , the linguistic information. The evaluation information takes the form of HIFLTSs, where  $E_{ij}$  is the evaluation value of the alternative  $A_i$  on the criteria  $C_j$ , and then the decision matrix is as follows:

$$E = \begin{bmatrix} \{(s_2, s_3), (s_0)\}, & \{(s_4, s_5, s_6), (s_0)\}, & \{(s_0, s_1, s_2), (s_4)\}, & \{(s_4, s_5), (s_0, s_1)\} \\ \{(s_3), (s_2, s_3)\}, & \{(s_2, s_3, s_4), (s_0, s_1)\}, & \{(s_2, s_3), (s_3, s_4)\}, & \{(s_6), (s_0)\} \\ \{(s_3, s_4), (s_0, s_1)\}, & \{(s_3, s_4), (s_0, s_2)\}, & \{(s_4), (s_0, s_2)\}, & \{(s_0, s_1, s_2, s_3), (s_3)\} \end{bmatrix}. \quad (26)$$

The procedure to obtain the most desirable alternative is as follows.

*Step 1.* Transform the decision matrix  $E = [E_{ij}]_{3 \times 4}$  into the normalized decision matrix  $\tilde{E} = [\tilde{E}_{ij}]_{3 \times 4}$ . We get

$$\tilde{E} = \begin{bmatrix} \{(s_2, s_3), (s_0, s_0)\}, & \{(s_4, s_5, s_6), (s_0, s_0)\}, & \{(s_0, s_1, s_2), (s_4, s_4)\}, & \{(s_4, s_4, s_4, s_5), (s_0, s_1)\} \\ \{(s_3, s_3), (s_2, s_3)\}, & \{(s_2, s_3, s_4), (s_0, s_1)\}, & \{(s_2, s_2, s_3), (s_3, s_4)\}, & \{(s_6, s_6, s_6, s_6), (s_0, s_0)\} \\ \{(s_3, s_4), (s_0, s_1)\}, & \{(s_3, s_3, s_4), (s_0, s_2)\}, & \{(s_4, s_4, s_4), (s_0, s_2)\}, & \{(s_0, s_1, s_2, s_3), (s_3, s_3)\} \end{bmatrix}. \quad (27)$$

*Step 2.* According to Definition 7, we determine HIFLPIS and HIFLNIS and have

$$\begin{aligned}
 \tilde{E}^+ & = \{ \{(s_3, s_4), (s_0, s_1)\}, \{(s_4, s_5, s_6), (s_0, s_0)\}, \{(s_4, s_4, s_4), (s_0, s_2)\}, \{(s_6, s_6, s_6, s_6), (s_0, s_0)\} \}, \\
 \tilde{E}^- & = \{ \{(s_3, s_3), (s_2, s_3)\}, \{(s_2, s_3, s_4), (s_0, s_1)\}, \{(s_0, s_1, s_2), (s_4, s_4)\}, \{(s_0, s_1, s_2, s_3), (s_3, s_3)\} \}.
 \end{aligned} \quad (28)$$

*Step 3.* Calculate the generalized hybrid hesitant intuitionistic fuzzy linguistic distances  $d_{ij}^+$  and  $d_{ij}^-$ , based on (17), let  $\lambda = 2$ , and then we construct the distance matrix as follows:

$$D^+ = \begin{bmatrix} 0.1336 & 0.0000 & 0.5273 & 0.2720 \\ 0.2525 & 0.2673 & 0.3712 & 0.0000 \\ 0.0000 & 0.2525 & 0.0000 & 0.7660 \end{bmatrix}, \quad (29)$$

$$D^- = \begin{bmatrix} 0.3571 & 0.2673 & 0.0000 & 0.4974 \\ 0.0000 & 0.0000 & 0.2422 & 0.7660 \\ 0.2525 & 0.1129 & 0.5273 & 0.0000 \end{bmatrix}.$$

According to (18), we calculate the grey relational coefficient of each alternative from HIFLPIS and HIFLNIS, without losing generality; let  $\theta = 0.5$ . We have

$$r^+ = \begin{bmatrix} 0.7413 & 1.0000 & 0.4207 & 0.5847 \\ 0.6026 & 0.5890 & 0.5079 & 1.0000 \\ 1.0000 & 0.6026 & 1.0000 & 0.3333 \end{bmatrix}, \quad (30)$$

$$r^- = \begin{bmatrix} 0.5175 & 0.5890 & 1.0000 & 0.4350 \\ 1.0000 & 1.0000 & 0.6126 & 0.3333 \\ 0.6026 & 0.7723 & 0.4207 & 1.0000 \end{bmatrix}.$$

*Step 4.* Solve the optimization model  $M3$  by using the `fmincon` function in Matlab software. We get

$$W = (0.3837, 0.3163, 0.2100, 0.0900)^T. \quad (31)$$

*Step 5.* Calculate the grey relational bidirectional projection between each alternative and the HIFLPIS and the grey relational bidirectional projection between each alternative and the HIFLNIS, according to (21). We have

$$\begin{aligned} BP_1^+ &= 0.9040, \\ BP_2^+ &= 0.8216, \\ BP_3^+ &= 0.9374, \\ BP_1^- &= 0.8376, \\ BP_2^- &= 0.9691, \\ BP_3^- &= 0.8445. \end{aligned} \quad (32)$$

Calculate the relative closeness, shown in (22). We obtain

$$\begin{aligned} \bar{C}_1 &= 0.5191, \\ \bar{C}_2 &= 0.4588, \\ \bar{C}_3 &= 0.5261. \end{aligned} \quad (33)$$

TABLE 1: The relative closeness with respect to  $\lambda = 2$ .

$\theta$	$A_1$	$A_2$	$A_3$	Rankings
0.2	0.4922	0.4506	0.5550	$A_3 > A_1 > A_2$
0.4	0.4980	0.4611	0.5445	$A_3 > A_1 > A_2$
0.6	0.4999	0.4684	0.5370	$A_3 > A_1 > A_2$
0.8	0.5005	0.4734	0.5317	$A_3 > A_1 > A_2$
1.0	0.5007	0.4771	0.5277	$A_3 > A_1 > A_2$

TABLE 2: The relative closeness with respect to  $\lambda = 5$ .

$\theta$	$A_1$	$A_2$	$A_3$	Rankings
0.2	0.4923	0.4504	0.5558	$A_3 > A_1 > A_2$
0.4	0.4984	0.4608	0.5452	$A_3 > A_1 > A_2$
0.6	0.5005	0.4681	0.5376	$A_3 > A_1 > A_2$
0.8	0.5012	0.4731	0.5322	$A_3 > A_1 > A_2$
1.0	0.5014	0.4768	0.5282	$A_3 > A_1 > A_2$

*Step 6.* Rank all the alternatives

$$A_3 > A_1 > A_2. \quad (34)$$

Furthermore, for comparative analysis, we apply the weight vector of the criteria in [26], then the result is as follows.

*Step 5\*.* Calculate the grey relational bidirectional projection between each alternative and the HIFLPIS and the grey relational bidirectional projection between each alternative and the HIFLNIS, according to (21). We have

$$\begin{aligned} BP_1^+ &= 0.8671, \\ BP_2^+ &= 0.8270, \\ BP_3^+ &= 0.9702, \\ BP_1^- &= 0.8690, \\ BP_2^- &= 0.9433, \\ BP_3^- &= 0.8319. \end{aligned} \quad (35)$$

Calculate the relative closeness, shown in (22). We obtain

$$\begin{aligned} \bar{C}_1 &= 0.4995, \\ \bar{C}_2 &= 0.4671, \\ \bar{C}_3 &= 0.5384. \end{aligned} \quad (36)$$

*Step 6\*.* Rank all the alternatives

$$A_3 > A_1 > A_2. \quad (37)$$

The results are the same as those in [26], so the proposed method in this paper is effective.

From Tables 1–4, we note that when the parameters  $\theta$ ,  $\lambda$  change, the relative closeness changes trends: when  $\lambda$  is fixed,

TABLE 3: The relative closeness with respect to  $\theta = 0.2$ .

$\lambda$	$A_1$	$A_2$	$A_3$	Rankings
2	0.4922	0.4506	0.5550	$A_3 > A_1 > A_2$
4	0.4922	0.4504	0.5557	$A_3 > A_1 > A_2$
6	0.4923	0.4504	0.5559	$A_3 > A_1 > A_2$
8	0.4924	0.4504	0.5559	$A_3 > A_1 > A_2$
10	0.4924	0.4504	0.5559	$A_3 > A_1 > A_2$

TABLE 4: The relative closeness with respect to  $\theta = 0.5$ .

$\lambda$	$A_1$	$A_2$	$A_3$	Rankings
2	0.4992	0.4651	0.5404	$A_3 > A_1 > A_2$
4	0.4996	0.4648	0.5410	$A_3 > A_1 > A_2$
6	0.4998	0.4648	0.5411	$A_3 > A_1 > A_2$
8	0.5000	0.4648	0.5411	$A_3 > A_1 > A_2$
10	0.5000	0.4649	0.5411	$A_3 > A_1 > A_2$

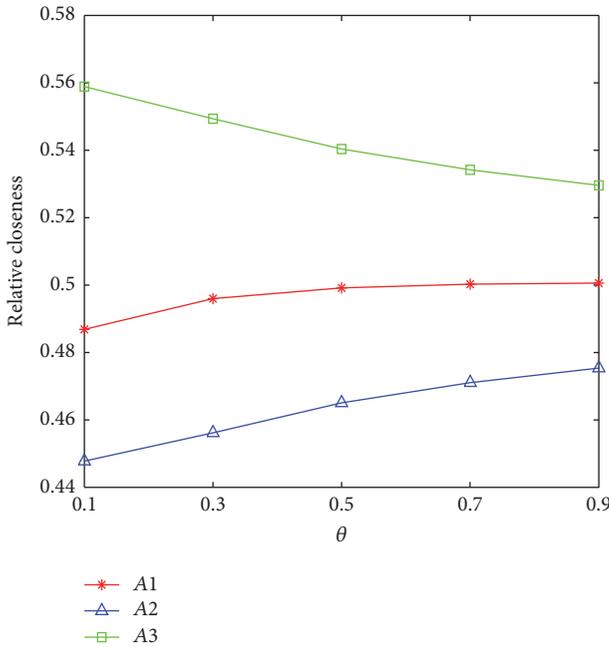


FIGURE 2: Relative closeness with respect to  $\theta$  when  $\lambda = 2$ .

the change  $\theta$  has obvious effect on the relative closeness, but when  $\theta$  is fixed, the change  $\lambda$  has no obvious effect on the relative closeness. All ranking results are  $A_3 > A_1 > A_2$ , which means that  $A_3$  is the best alternative. Moreover, the trends of the relative closeness of three alternatives can be shown directly in Figures 2–5, and we also present the relative closeness for the three alternatives when  $\lambda$  and  $\theta$  change simultaneously, which is shown in Figures 6–8; the results show that  $A_3$  is always the best alternative. Thus, the choice of the parameter  $\theta$  can reflect the decision maker’s risk attitude, if the decision maker is risk averse, let  $\theta$  be a larger value, and vice versa.

5.2. Advantages of the Proposed Method

- (1) Compared with the possibility degree of HFLTSS studied in [11–13], the possibility degree formula proposed in this paper can overcome the drawback of the

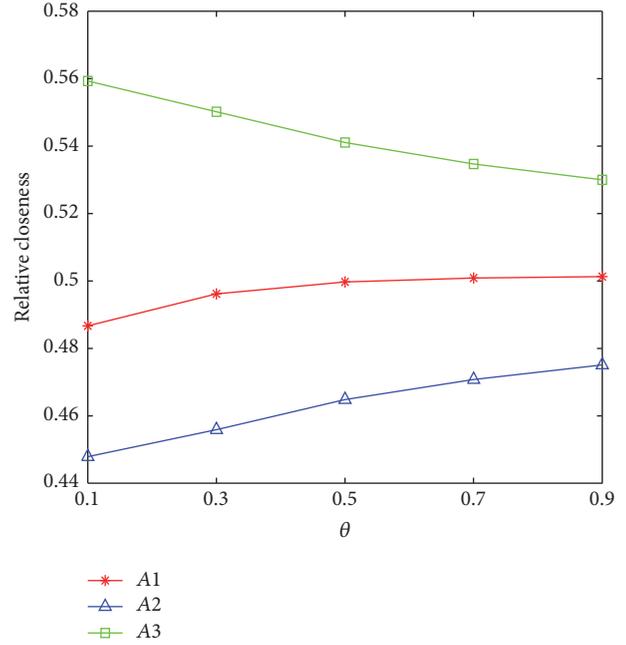


FIGURE 3: Relative closeness with respect to  $\theta$  when  $\lambda = 5$ .

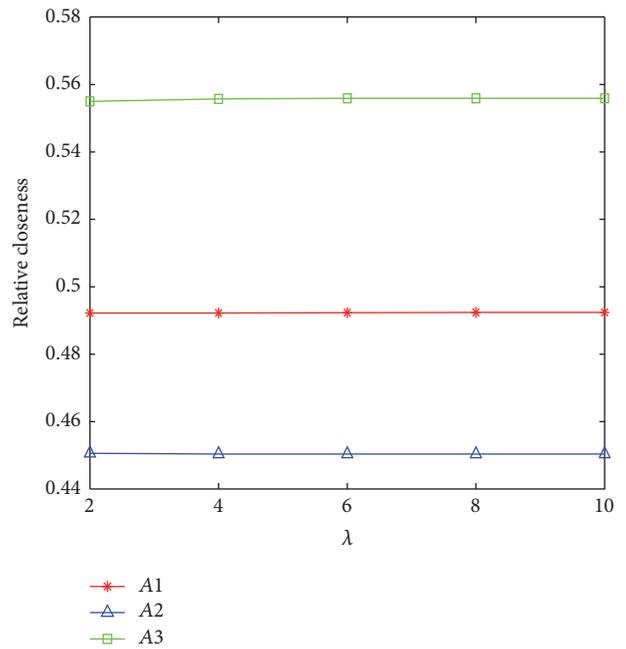


FIGURE 4: Relative closeness with respect to  $\lambda$  when  $\theta = 0.2$ .

previous work. We derived the possibility degree of HIFLTSS based on the possibility degree of HFLTSS, which can compare the HIFLTSS more effectively. Moreover, the possibility degree and the distance measure which are proposed in this paper are not only the foundation of the proposed grey relational bidirectional projection method but also foundation of many classical decision making methods including TOPSIS, PROMETHEE, and VIKOR.

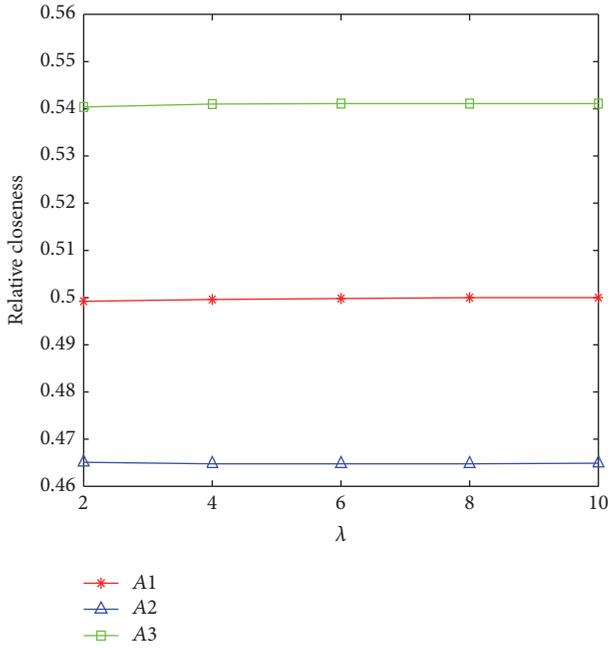


FIGURE 5: Relative closeness with respect to  $\lambda$  when  $\theta = 0.5$ .

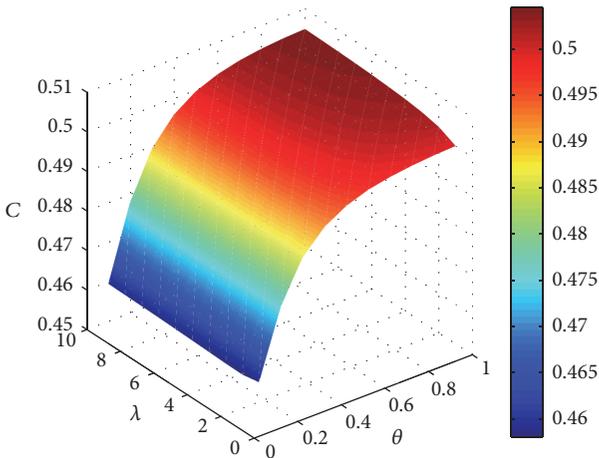


FIGURE 6: Relative closeness for  $A_1(\lambda \in (0, 10], \theta \in (0, 1])$ .

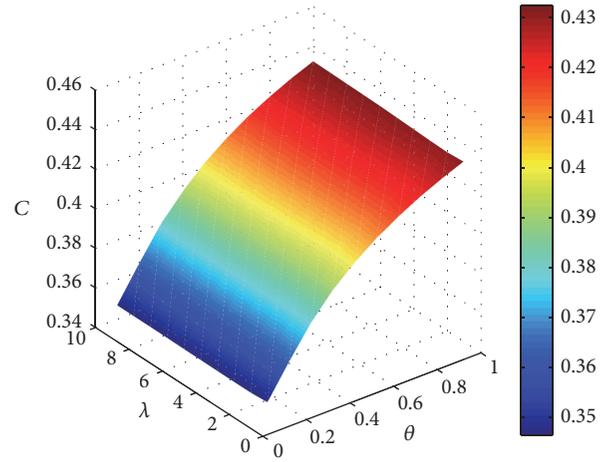


FIGURE 7: Relative closeness for  $A_2(\lambda \in (0, 10], \theta \in (0, 1])$ .

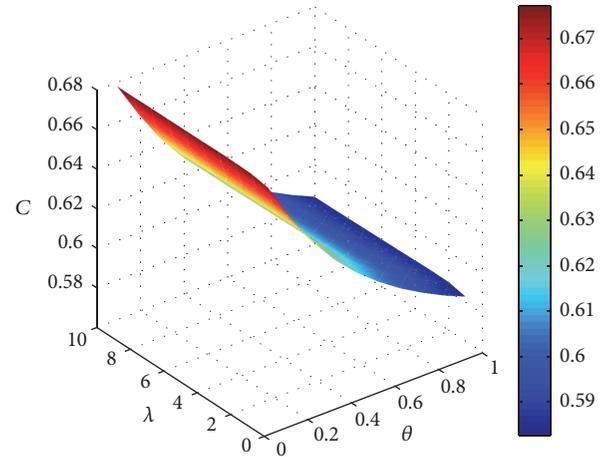


FIGURE 8: Relative closeness for  $A_3(\lambda \in (0, 10], \theta \in (0, 1])$ .

- (2) In contrast to the studies in [15, 22], the grey relational bidirectional projection method combines the distance measure and the correlation coefficient, which can integrate the influence of the whole criteria space and avoid the unidirectional deviation.
- (3) The criteria weights are determined by the nonlinear optimization model, which consists of the minimum deviation model [46, 47] and the information entropy model [48, 49]. The optimization model combines the advantages of both methods, which is based on the evaluation and the weights information. Therefore, we can reduce the impact of subjective human factors and obtain the reasonable criteria weights.
- (4) The proposed grey relational bidirectional projection method integrates the grey relational analysis into the

bidirectional projection. The method can overcome the drawbacks of the traditional grey relational projection [30–32], which cannot rank the preference order of the alternative in some situations.

## 6. Conclusion

The hesitant intuitionistic fuzzy linguistic term sets have a good advantage in the expression of hesitation information, which is especially important in the background of qualitative decision making. In this paper, we have investigated a novel qualitative decision making method based on the grey relational bidirectional projection with the hesitant intuitionistic fuzzy linguistic information. We have derived the possibility degree for comparing the HIFLTs. We have proposed a family of distance measure for the HIFLTs. Based on the possibility degree and the distance measure, we have proposed the grey relational bidirectional projection method, which combines the grey relational analysis and the vector projection theory. The combined nonlinear optimization model has been used to determine the criteria weight,

where the information on criteria weights is partly known. The numerical example shows that the proposed method is suitable for dealing with the MCDM. Moreover, the model has good practicability and can be further applied in wider fields.

In future research, we will apply our possibility degree and distance measures to order decision making methods, such as ELECTRE and PROMETHEE. Furthermore, we will also focus on the extended hesitant intuitionistic fuzzy linguistic term sets theory, including the interval-valued hesitant intuitionistic fuzzy linguistic term sets and the hesitant intuitionistic fuzzy uncertain linguistic term sets.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

The authors would like to acknowledge the support from the National Natural Science Foundation of China (nos. 71671159 and 71301139), the Natural Science Foundation of Hebei Province (nos. G2018203302 and G2016203236), the Project Funded by Hebei Education Department (nos. BJ2017029 and BJ2016063), and Hebei Talents Program (no. A2017002108).

## References

- [1] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning-I," *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975.
- [2] J. Ye, "Multiple attribute group decision making based on interval neutrosophic uncertain linguistic variables," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 3, pp. 1–12, 2015.
- [3] R. Verma, "Multiple attribute group decision making based on generalized trapezoid fuzzy linguistic prioritized weighted average operator," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 6, pp. 1993–2007, 2017.
- [4] H. Zhang, "Uncertain linguistic power geometric operators and their use in multiattribute group decision making," *Mathematical Problems in Engineering*, vol. 2015, Article ID 948380, 2015.
- [5] K.-Y. Song, G. T. G. Seniuk, J. A. Kozinski, W.-J. Zhang, and M. M. Gupta, "An innovative fuzzy-neural decision analyzer for qualitative group decision making," *International Journal of Information Technology & Decision Making*, vol. 14, no. 3, pp. 659–696, 2015.
- [6] W. Yang, Y. Pang, J. Shi, and C. Wang, "Linguistic hesitant intuitionistic fuzzy decision-making method based on VIKOR," *Neural Computing and Applications*, pp. 1–14, 2016.
- [7] Y. Ju and S. Yang, "A new method for multiple attribute group decision-making with intuitionistic trapezoid fuzzy linguistic information," *Soft Computing*, vol. 19, no. 8, pp. 2211–2224, 2015.
- [8] C.-S. Lin, C.-T. Chen, F.-S. Chen, and W.-Z. Hung, "A novel multiperson game approach for linguistic multicriteria decision making problems," *Mathematical Problems in Engineering*, vol. 2014, Article ID 592326, 2014.
- [9] D.-C. Luo, M.-H. Yang, and L.-F. Gao, "Research on risk assessment of financial market's venture capital project with hesitant fuzzy uncertain linguistic information," *Journal of Computational and Theoretical Nanoscience*, vol. 13, no. 10, pp. 7115–7119, 2016.
- [10] S. Yang, Z. Sun, Y. Ju, and C. Qiao, "A Novel multiple attribute satisfaction evaluation approach with hesitant intuitionistic linguistic fuzzy information," *Mathematical Problems in Engineering*, vol. 2014, Article ID 692782, pp. 1–15, 2014.
- [11] R. M. Rodriguez, L. Martinez, and F. Herrera, "Hesitant fuzzy linguistic term sets for decision making," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 1, pp. 109–119, 2012.
- [12] C. Wei, N. Zhao, and X. Tang, "Operators and comparisons of hesitant fuzzy linguistic term sets," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 3, pp. 575–585, 2014.
- [13] L.-W. Lee and S.-M. Chen, "Fuzzy decision making based on likelihood-based comparison relations of hesitant fuzzy linguistic term sets and hesitant fuzzy linguistic operators," *Information Sciences*, vol. 294, pp. 513–529, 2015.
- [14] Z.-P. Tian, J. Wang, J.-Q. Wang, and H.-Y. Zhang, "A likelihood-based qualitative flexible approach with hesitant fuzzy linguistic information," *Cognitive Computation*, vol. 8, no. 4, pp. 670–683, 2016.
- [15] H. Liao, Z. Xu, and X.-J. Zeng, "Distance and similarity measures for hesitant fuzzy linguistic term sets and their application in multi-criteria decision making," *Information Sciences*, vol. 271, pp. 125–142, 2014.
- [16] H. Liao and Z. Xu, "Approaches to manage hesitant fuzzy linguistic information based on the cosine distance and similarity measures for HFLTSs and their application in qualitative decision making," *Expert Systems with Applications*, vol. 42, no. 12, pp. 5328–5336, 2015.
- [17] J.-Q. Wang, J.-T. Wu, J. Wang, H.-Y. Zhang, and X.-H. Chen, "Multi-criteria decision-making methods based on the Hausdorff distance of hesitant fuzzy linguistic numbers," *Soft Computing*, vol. 20, no. 4, pp. 1621–1633, 2016.
- [18] J.-Q. Wang, J. Wang, Q.-H. Chen, H.-Y. Zhang, and X.-H. Chen, "An outranking approach for multi-criteria decision-making with hesitant fuzzy linguistic term sets," *Information Sciences*, vol. 280, pp. 338–351, 2014.
- [19] F. Meng and X. Chen, "A hesitant fuzzy linguistic multi-granularity decision making model based on distance measures," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 28, no. 4, pp. 1519–1531, 2015.
- [20] H. Liao, Z. Xu, and X.-J. Zeng, "Hesitant Fuzzy Linguistic VIKOR Method and Its Application in Qualitative Multiple Criteria Decision Making," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 5, pp. 1343–1355, 2015.
- [21] C. Wei, Z. Ren, and R. M. Rodríguez, "A hesitant fuzzy linguistic TODIM method based on a score function," *International Journal of Computational Intelligence Systems*, vol. 8, no. 4, pp. 701–712, 2015.
- [22] H. Liao, Z. Xu, X.-J. Zeng, and J. M. Merigó, "Qualitative decision making with correlation coefficients of hesitant fuzzy linguistic term sets," *Knowledge-Based Systems*, vol. 76, pp. 127–138, 2015.
- [23] R. Lin, X. Zhao, H. Wang, and G. Wei, "Hesitant fuzzy linguistic aggregation operators and their application to multiple attribute decision making," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 27, no. 1, pp. 49–63, 2014.
- [24] B. Farhadinia, "Multiple criteria decision-making methods with completely unknown weights in hesitant fuzzy linguistic term setting," *Knowledge-Based Systems*, vol. 93, pp. 135–144, 2016.

- [25] X. Gou, Z. Xu, and H. Liao, "Hesitant fuzzy linguistic entropy and cross-entropy measures and alternative queuing method for multiple criteria decision making," *Information Sciences*, vol. 388-389, pp. 225–246, 2017.
- [26] I. Beg and T. Rashid, "TOPSIS for hesitant fuzzy linguistic term sets," *International Journal of Intelligent Systems*, vol. 28, no. 12, pp. 1162–1171, 2013.
- [27] X. Y. Liu, Y. B. Ju, and S. H. Yang, "Hesitant intuitionistic fuzzy linguistic aggregation operators and their applications to multiple attribute decision making," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, 2014.
- [28] T. Rashid, S. Faizi, Z. Xu, and S. Zafar, "ELECTRE-based outranking method for multi-criteria decision making using hesitant intuitionistic fuzzy linguistic term sets," *International Journal of Fuzzy Systems*, 2017.
- [29] S. Faizi, T. Rashid, and S. Zafar, "An outranking method for multi-criteria group decision making using hesitant intuitionistic fuzzy linguistic term sets," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 32, no. 3, pp. 2153–2164, 2017.
- [30] G. Zheng, Y. Jing, H. Huang, and Y. Gao, "Application of improved grey relational projection method to evaluate sustainable building envelope performance," *Applied Energy*, vol. 87, no. 2, pp. 710–720, 2010.
- [31] X. Zhang, F. Jin, and P. Liu, "A grey relational projection method for multi-attribute decision making based on intuitionistic trapezoidal fuzzy number," *Applied Mathematical Modelling*, vol. 37, no. 5, pp. 3467–3477, 2013.
- [32] H.-C. Liu, J.-X. You, X.-J. Fan, and Q.-L. Lin, "Failure mode and effects analysis using D numbers and grey relational projection method," *Expert Systems with Applications*, vol. 41, no. 10, pp. 4670–4679, 2014.
- [33] M. Delgado, J. L. Verdegay, and M. A. Vila, "Linguistic decision-making models," *International Journal of Intelligent Systems*, vol. 7, no. 5, pp. 479–492, 1992.
- [34] Z. S. Xu, "Deviation measures of linguistic preference relations in group decision making," *Omega*, vol. 33, no. 3, pp. 249–254, 2005.
- [35] Y. Kuo, T. Yang, and G.-W. Huang, "The use of grey relational analysis in solving multiple attribute decision-making problems," *Computers and Industrial Engineering*, vol. 55, no. 1, pp. 80–93, 2008.
- [36] Y.-C. Hu, "A novel fuzzy classifier with Choquet integral-based grey relational analysis for pattern classification problems," *Soft Computing*, vol. 12, no. 6, pp. 523–533, 2008.
- [37] G. Wei, "Gray relational analysis method for intuitionistic fuzzy multiple attribute decision making," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11671–11677, 2011.
- [38] S. Pramanik and D. Mukhopadhyaya, "Grey relational analysis based intuitionistic fuzzy multi-criteria group decision-making approach for teacher selection in higher education," *International Journal of Computer Application*, vol. 34, pp. 21–29, 2013.
- [39] H. X. Tang, "A novel fuzzy soft set approach in decision making based on grey relational analysis and Dempster-Shafer theory of evidence," *Applied Soft Computing*, vol. 31, pp. 317–325, 2015.
- [40] N. Xie, Y. Han, and Z. Li, "A novel approach to fuzzy soft sets in decision making based on grey relational analysis and MYCIN certainty factor," *International Journal of Computational Intelligence Systems*, vol. 8, no. 5, pp. 959–976, 2015.
- [41] Z. Xu and Q. Da, "Projection method for uncertain multi-attribute decision making with preference information on alternatives," *International Journal of Information Technology and Decision Making*, vol. 3, no. 3, pp. 429–434, 2004.
- [42] S. Zeng, T. Balezentis, J. Chen, and G. Luo, "A projection method for multiple attribute group decision making with intuitionistic fuzzy information," *Informatika*, vol. 24, no. 3, pp. 485–503, 2013.
- [43] Z. Yue, "Approach to group decision making based on determining the weights of experts by using projection method," *Applied Mathematical Modelling*, vol. 36, no. 7, pp. 2900–2910, 2012.
- [44] Y. Ju and A. Wang, "Projection method for multiple criteria group decision making with incomplete weight information in linguistic setting," *Applied Mathematical Modelling*, vol. 37, no. 20-21, pp. 9031–9040, 2013.
- [45] J. Ye, "Bidirectional projection method for multiple attribute group decision making with neutrosophic numbers," *Neural Computing and Applications*, vol. 28, no. 5, pp. 1–9, 2015.
- [46] G. Wei and X. Zhao, "Minimum deviation models for multiple attribute decision making in intuitionistic fuzzy setting," *International Journal of Computational Intelligence Systems*, vol. 4, no. 2, pp. 174–183, 2011.
- [47] R. Sahin and P. Liu, "Maximizing deviation method for neutrosophic multiple attribute decision making with incomplete weight information," *Neural Computing Applications*, vol. 7, pp. 1–13, 2010.
- [48] D.-L. Mon, C.-H. Cheng, and J.-C. Lin, "Evaluating weapon system using fuzzy analytic hierarchy process based on entropy weight," *Fuzzy Sets and Systems*, vol. 62, no. 2, pp. 127–134, 1994.
- [49] J. Ye, "Multicriteria fuzzy decision-making method using entropy weights-based correlation coefficients of interval-valued intuitionistic fuzzy sets," *Applied Mathematical Modelling*, vol. 34, no. 12, pp. 3864–3870, 2010.

## Research Article

# An Improved Shuffled Frog Leaping Algorithm and Its Application in Dynamic Emergency Vehicle Dispatching

Xiaohong Duan , Tianyong Niu, and Qi Huang

*School of Economics and Management, North China University of Technology, Beijing 100144, China*

Correspondence should be addressed to Xiaohong Duan; 12116350@bjtu.edu.cn

Received 22 November 2017; Revised 20 January 2018; Accepted 28 January 2018; Published 28 March 2018

Academic Editor: Guillermo Cabrera-Guerrero

Copyright © 2018 Xiaohong Duan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The traditional method for solving the dynamic emergency vehicle dispatching problem can only get a local optimal strategy in each horizon. In order to obtain the dispatching strategy that can better respond to changes in road conditions during the whole dispatching process, the real-time and time-dependent link travel speeds are fused, and a time-dependent polygonal-shaped link travel speed function is set up to simulate the predictable changes in road conditions. Response times, accident severity, and accident time windows are taken as key factors to build an emergency vehicle dispatching model integrating dynamic emergency vehicle routing and selection. For the unpredictable changes in road conditions caused by accidents, the dispatching strategy is adjusted based on the real-time link travel speed. In order to solve the dynamic emergency vehicle dispatching model, an improved shuffled frog leaping algorithm (ISFLA) is proposed. The global search of the improved algorithm uses the probability model of estimation of distribution algorithm to avoid the partial optimal solution. Based on the Beijing expressway network, the efficacy of the model and the improved algorithm were tested from three aspects. The results have shown the following: (1) Compared with SFLA, the optimization performance of ISFLA is getting better and better with the increase of the number of decision variables. When the possible emergency vehicle selection strategies are  $8^{15}$ , the objective function value of optimal selection strategies obtained by the base algorithm is 210.10% larger than that of ISFLA. (2) The prediction error of the travel speed affects the accuracy of the initial emergency vehicle dispatching. The prediction error of  $\pm 10$  can basically meet the requirements of the initial dispatching. (3) The adjustment of emergency vehicle dispatching strategy can successfully bypassed road sections affected by accidents and shorten the response time.

## 1. Introduction

Urban Expressway can ease the traffic pressure on large cities and plays an important role in the urban traffic system. However, with the increasing amount of traffic, accidents often happen on expressways and cause great damage to people's life and property. Rapid emergency rescue can effectively reduce accident loss, and emergency vehicle dispatching is the key to emergency rescue.

Research of emergency vehicle dispatching started with assumptions of static travel time or distance, and the original dispatching problem mainly included two basic issues. In the problem with one accident, we only need to choose the nearest emergency vehicle to rescue the accident. The core is the shortest paths of emergency vehicles in the road network

[1]. In the problem with multiple accidents, choosing the nearest emergency vehicle need not be an optimal decision [2]. We also need to select suitable emergency vehicles for different accidents to minimize their response times. It is a combinatorial optimization problem. Therefore, models for solving combinatorial optimization, such as Hungarian method, direct cost model, and opportunity cost model, were used to solve the emergency vehicle dispatching [3]. Aiming at the random resource requirements of potential incidents, Ozbay et al. used probabilistic constraint to improve the opportunity cost model [4]. Emergency vehicle dispatching with multiple accidents is a complex problem relating to various factors. In addition to response time, factors such as fairness, cost, and loss were considered to set up multi-objective dispatching models [5, 6]. In order to solve these

NP-hard problems, ant colony algorithm [7], particle swarm optimization [8], genetic algorithm [9], and other intelligent optimization algorithms had been widely applied.

Emergency vehicle dispatching problem considering the dynamic change of link weight (travel time, distance) started in the late 1990s. Taking the minimum response time as objective, Zografos et al. [10] integrated routing and dispatching module to set up an emergency response decision support system. Haghani et al. [11] built a simulation model of emergency vehicle dispatching. The model can assist decision makers to select suitable emergency vehicles and guide them to avoid congested areas. Dan et al. [12] divided dynamic dispatching problem into a series of static problems. Dispatching strategy was updated based on the time axis. A multiobjective model was established to solve these static problems. Yang et al. [13] set up an online dispatching and routing model for emergency vehicles. One day was divided into a number of intervals, and dispatching strategy was updated according to link travel time in each interval. Fu et al. [14] calculated the earliest response time using iteration method. A dynamic emergency resource dispatching system was designed. This research is essentially a static method for solving the dynamic problem. Dispatching strategies are continuously adjusted based on real-time traffic data at each decision-making instant. They suppose traffic data remain unchanged in the decision-making horizon. Emergency vehicles may be stopped up on the way to accidents because of changes in road conditions.

If link weight is thought to be a time-dependence function, this network is a kind of time-dependence network. Research on the time-dependence network still focuses on universal shortest path problem. On the assumption of discrete link weight based on travel time, Cooke and Halsey [15] proposed a model of the time-dependence network. Iteration method was used to solve the shortest path problem. Kaufman and Smith [16] proved that the time-dependence shortest path problem can be solved by polynomials only when the network satisfies the first-in-first-out (FIFO) property. Most of the research on the shortest path of time-dependence traffic network is based on FIFO property [17]. However, road network with discrete link travel time proposed by Cooke is not FIFO [16]. In order to solve this problem, Duan et al. [18] proposed a universal shuffled frog leaping algorithm for solving the shortest path of the non-FIFO and FIFO network. Ichoua et al. [19] replaced link travel time with link travel speed to build time-dependent function. Travel time calculated by travel speed function changes continuously and satisfies the FIFO property. Since link travel speed cannot be known ahead of the decision-making instant, Ichoua et al. divided a day into three horizons to distinguish between congestion and free flow. During each time horizon, link travel speed remains unchanged. If this travel speed function is used to solve emergency vehicle dispatching problem, the whole dispatching process may be at a certain time horizon with constant link travel speed. This travel speed function cannot satisfy the requirement of emergency vehicle dispatching. On the assumption that link travel speed decreases continuously with the entry instant, Yuan and Wang [20] proposed an emergency vehicle routing

model taking the shortest travel time as objective. Zhou et al. [21] built a multiobjective optimization model to solve the multiperiod dynamic emergency resource scheduling problems. In order to solve the model, a multiobjective evolutionary algorithm was proposed. Zhou and Liu [22] designed a multiagent genetic algorithm to solve the multiperiod emergency resource scheduling problem considering the uncertainty of traffic. The experimental results show that the multiagent genetic algorithm precedes genetic algorithm for the problem.

According to the review of the literature, link travel speed function can reflect the dynamic changes of road conditions. However, it is difficult to model link travel speed function and solve the dynamic dispatching problem, and the dynamic problem is usually divided into a series of static problems. It only can get a local optimal strategy in each horizon. In order to get the overall optimal strategy, real-time data (reflecting the real-time road condition at decision-making instant) fuses with prediction data (reflecting the change of road condition in the whole dispatching process) to establish the link travel speed function. Meanwhile, considering the unpredictable changes in road conditions during the dispatching process, dispatching strategies are adjusted according to real-time travel speed. Multiple-incident and multiple-response (MIMR) emergency vehicle dispatching discussed in this paper is NP-hard problem with large scale variables. The metaheuristic algorithm has advantages in solving the NP-hard problem. Shuffled frog leaping algorithm (SFLA) is a relatively new heuristic algorithm. It was first proposed and applied in water distribution network designed by Eusuff et al. [23, 24]. This algorithm combines the advantages of particle swarm optimization (PSO) and shuffled complex evolution (SCE) algorithm, and it has been proved that the algorithm has good performance in convergence speed and solution precision [25, 26]. It was used to solve many real-world problems such as job shop scheduling and cloud computing resource allocation [27–29].

According to the above, in this paper, the polygonal time-dependent function based on real-time and prediction link travel speed is built to simulate real road conditions in expressway network. Integrating routing and selection of emergency vehicles, a dynamic dispatching model is built. The model takes time-dependent travel speed, response time, time window, and accident severity as key factors to get the optimal strategy. And the dispatching strategy is adjusted when the new accidents happen. An improved shuffled frog leaping algorithm (IFSFLA) is put forward to solve the dynamic dispatching model. The algorithm uses the probabilistic model of the distribution estimation algorithm to generate new frog population. It can avoid a local optimum of shuffled frog leaping algorithm.

## 2. Problem Statement

Based on graph theory, the expressway network is abstracted as a time-dependent directed network model  $(N, E, T(t) \times Q)$  as shown in Figure 1.  $N = \{n_1, n_2, \dots, n_M\}$  is the node set. It consists of hubs and interchanges on the expressway.  $M$  is the total number of nodes. The link arc between adjacent nodes  $n_i$

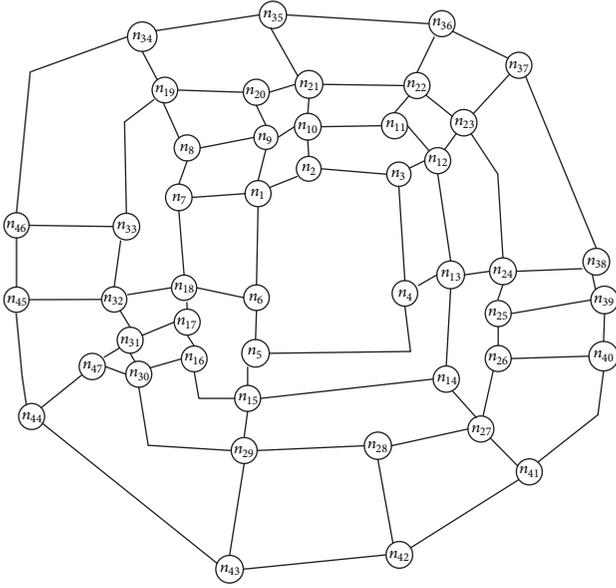


FIGURE 1: Expressway network model.

and  $n_j$  is road section  $(n_i, n_j) \in E$ . The expressway network is a directed network, so  $(n_i, n_j) \neq (n_j, n_i)$ . Emergency vehicles must run along the direction of the road section, and they

can change the direction at the nodes.  $Q$  is the interested time horizon.  $T_{ij}(t) \in T(t)$  is the link travel time function defined in time horizon  $Q$ . It represents a time for the emergency vehicle, leaving at an instant  $t$ , traveling from node  $n_i$  to  $n_j$ .  $\forall t \in Q$ ,  $t + T_{ij}(t)$  is always defined. For  $t \notin Q$ ,  $T_{ij}(t)$  is defined as infinity.

(1) *Link Travel Time Function.* Taking  $\kappa$  as the minimum time interval,  $Q$  is divided into discrete time intervals, that are  $Q = \{[t_0, t_1], [t_1, t_2], \dots, [t_\phi, t_{\phi+1}], \dots, [t_{\Phi-1}, t_\Phi]\}$ ,  $t_\phi = t_0 + \phi \cdot \kappa$ ,  $\phi = 0, 1, \dots, \Phi - 1$ .  $t_0$  is the initial dispatching decision-making instant.  $t_\Phi = t_0 + \Phi \cdot \kappa$  is the last instant, and make sure it is large enough for the emergency vehicle to arrive at the accident during the time period  $[t_0, t_\Phi]$ .  $v_{ij}(t)$  is the link travel speed function. It represents the average speed of the road section  $(n_i, n_j) \in E$  at  $t \in [t_0, t_\Phi]$ . Based on the link travel speed function of Ichoua et al. [19], it is assumed that travel speed in each time interval changes in the form of the polygonal line, and the polygonal-shaped travel speed function is shown in Figure 2. At the decision-making instant  $t_0$ , the real-time link travel speed  $v_{ij}^1(t_0)$  is known. However, the real-time link travel speed  $v_{ij}^1(t)$ ,  $t \in \{t_1, t_2, \dots, t_\phi, \dots, t_\Phi\}$ ,  $t_\phi = t_0 + \phi \cdot \kappa$ ,  $\phi = 1, 2, \dots, \Phi$ , cannot be obtained. Therefore, they are approximated by the prediction travel speeds  $v_{ij}^2(\phi)_{t_0}$ .

The polygonal-shaped travel speed function shown in Figure 2 can be expressed as

$$v_{ij}(t)_{t_0} = \begin{cases} v_{ij}^2(1)_{t_0} - v_{ij}^1(t_0) & \cdot t + \frac{t_1 \cdot v_{ij}^1(t_0) - t_0 \cdot v_{ij}^2(1)_{t_0}}{\kappa} & t_0 \leq t < t_1 \\ v_{ij}^2(\phi+1)_{t_0} - v_{ij}^2(\phi)_{t_0} & \cdot t + \frac{t_{\phi+1} \cdot v_{ij}^2(\phi)_{t_0} - t_\phi \cdot v_{ij}^2(\phi+1)_{t_0}}{\kappa} & t_\phi \leq t < t_{\phi+1}, \phi = 1, \dots, \Phi - 1. \end{cases} \quad (1)$$

$v_{ij}(t)_{t_0}$  is continuous on  $[t_0, t_\Phi]$ , and it must be integrable on the interval  $[t_0, t_\Phi]$ . If the emergency vehicle enters the road section  $(n_i, n_j)$  at the instant  $y$  ( $y \geq t_0$ ), its travel distance is a function of travel time  $x$  ( $y \leq x \leq t_\Phi$ ).

$$\int_y^x v_{ij}(t)_{t_0} dt = \int_{t_0}^x v_{ij}(t)_{t_0} dt - \int_{t_0}^y v_{ij}(t)_{t_0} dt \quad (2)$$

$$= \eta(x) - \vartheta(y),$$

in which,

$$\eta(x) = \int_{t_0}^x v_{ij}(t)_{t_0} dt = \int v_{ij}(x)_{t_0} dx + C_1, \quad (3)$$

$$\vartheta(y) = \int_{t_0}^y v_{ij}(t)_{t_0} dt = \int v_{ij}(y)_{t_0} dy + C_2.$$

Let formula (2) equal to the length  $Ls_{ij}$  of the road section  $(n_i, n_j)$ ; the time when the emergency vehicle leaves the road section is

$$x = g(Ls_{ij} + \vartheta(y)), \quad (4)$$

in which,  $g$  is the inverse function of  $\eta$ .

When the emergency vehicle enters the road section at instant  $t$ , the travel time function of the road section is

$$T_{ij}(t) = g(Ls_{ij} + \vartheta(t)) - t. \quad (5)$$

(2) *Dynamic Emergency Vehicle Dispatching Process.*  $U(t)$  accidents are waiting for rescue at the instant  $t$ , and they compose set  $AC(t)$ .  $Ac_u(t) \in AC(t)$  is the  $u$ th accident,  $u = 1, 2, \dots, U(t)$ . The road section where the accident  $Ac_u(t)$  occurred is expressed as  $(n_0^u, n_1^u) \in E$ , and its location node is  $Nc_u(t)$ , its rescue time window is  $[0, T_{\max}^u(t)]$ , and the required number of emergency vehicles is  $Na_u(t) > 0$ .  $As_u(t)$  represents the severity of the accident  $Ac_u(t)$ .  $EV(t)$  is the emergency vehicle set, and  $Ev_l(t) \in EV(t)$ ,  $l = 1, 2, \dots, L(t)$ , stands for the  $l$ th emergency vehicle.  $L(t)$  is the total number of emergency vehicles. The road section where the emergency vehicle  $Ev_l(t)$  is located is expressed as  $(n_0^l, n_1^l) \in E$ , and its location node is  $Nv_l(t)$ . When the emergency vehicle  $Ev_l(t) \in EV(t)$  starts traveling at an instant  $t$ , the shortest time path to the accident  $Ac_u(t) \in AC(t)$  is  $P_{lu}(t)$ , and the shortest travel time is  $T_{lu}(t)$ .

The basic purpose of emergency vehicle dispatching is to shorten the response time of accidents. The response time

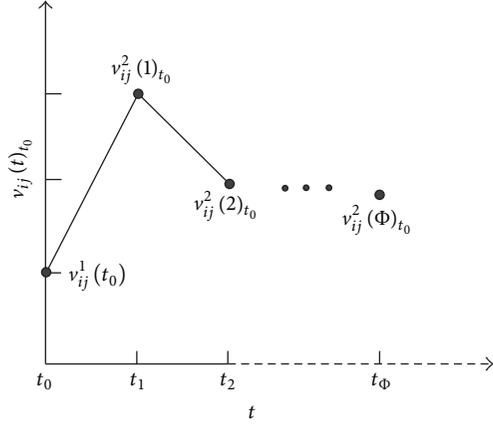


FIGURE 2: Polygonal-shaped travel speed function.

mainly depends on the travel time of the emergency vehicles. In the complex expressway network, emergency vehicles can reach accidents through multiple paths. Therefore, the shortest time path problem should be solved and the shortest travel time  $T_{lu}(t)$  from each emergency vehicle  $Ev_l(t) \in EV(t)$  to each accident  $Ac_u(t) \in AC(t)$  should be obtained firstly. Then, taking  $T_{lu}(t)$  as the key factor, the suitable emergency vehicles are selected to rescue the accidents, that is, to solve the problem of emergency vehicle selection.

According to the emergency vehicle selection strategy, emergency vehicles head for accidents along the shortest paths. In this process, if there are no new accidents in the road network, travel speed function of each road section does not change obviously, but once an accident happened,

travel speeds of road sections in the accident area will be greatly affected. In order to avoid the rescue delay caused by sudden changes in road conditions, it is necessary to update link travel speed function according to the real-time speed, and the dispatching decision should be adjusted. We only need to update the travel time functions of road sections where the new accidents are located and their upstream road sections. If the rescue paths of the accident go through the new accidents, the rescue strategy for the affected accident needs to be recalculated, and the rescue strategies for the new accidents need to be calculated too.

The dynamic dispatching process consists of two stages: (1) At the decision instant  $t = t_{\varphi}$ , taking the polygonal-shaped travel speed function as the weight of the road section, the travel path from  $Ev_l(t_{\varphi})$  to  $Ac_u(t_{\varphi})$  is planned and the shortest travel time  $T_{lu}(t_{\varphi})$  is obtained. (2) Taking the shortest travel time  $T_{lu}(t_{\varphi})$  as input, emergency vehicles are selected to rescue the accidents. (3) At the instant  $t = t_{\varphi+1}$ , a new accident occurs in the road network. The dispatching strategy, including vehicle routing and selection, is dynamically adjusted according to the updated real-time link travel speed.

### 3. Dynamic Emergency Vehicle Dispatching Modeling

At the decision-making instant  $t = t_{\varphi}$ , the shortest travel time, the required number of emergency vehicles, the upper limit of rescue time window, and the accident severity are taken as the key factors. A dynamic dispatching model with vehicle routing is built. A list of all symbols is given in Symbols.

$$\min \left\{ \sum_u \sum_l A s_u(t_{\varphi}) \times T_{lu}(t_{\varphi}) \times x_{lu}(t_{\varphi}) + \sum_u M \times z_u(t_{\varphi}) \right\},$$

$$Ev_l \in EV(t_{\varphi}), \quad (6)$$

$$\forall Ac_u \in \begin{cases} AC(t_{\varphi}) & t_{\varphi} = t_0 \\ AC^1(t_{\varphi}) & t_{\varphi} \neq t_0 \end{cases}$$

$$\text{s.t. } \sum_l x_{lu}(t_{\varphi}) = N a_u(t_{\varphi}),$$

$$Ev_l \in EV(t_{\varphi}), \quad (7)$$

$$\forall Ac_u \in \begin{cases} AC(t_{\varphi}) & t_{\varphi} = t_0 \\ AC^1(t_{\varphi}) & t_{\varphi} \neq t_0 \end{cases}$$

$$t_{\max}^u(t_{\varphi}) - T_{\max}^u(t_{\varphi}) \leq M \times z_u(t_{\varphi}),$$

$$t_{\max}^u(t_{\varphi}) = \max \{ T_{lu}(t_{\varphi}) \times x_{lu}(t_{\varphi}) \},$$

$$Ev_l \in EV(t_{\varphi}), \quad (8)$$

$$\forall Ac_u \in \begin{cases} AC(t_{\varphi}) & t_{\varphi} = t_0 \\ AC^1(t_{\varphi}) & t_{\varphi} \neq t_0 \end{cases}$$

$$\sum_u x_{lu}(t_\varphi) + \sum_l xx_l(t_\varphi) = 1, \tag{9}$$

$$Ac_u \in AC(t_\varphi),$$

$$\forall Ev_l \in EV(t_\varphi)$$

$$\sum_l x_{lu}(t_\varphi) + \sum_l xx_l(t_\varphi) = L(t_\varphi), \tag{10}$$

$$Ev_l \in EV(t_\varphi),$$

$$\forall Ac_u \in AC(t_\varphi)$$

$$x_{lu}(t_\varphi) = 0, 1,$$

$$\forall Ev_l \in EV(t_\varphi), \tag{11}$$

$$\forall Ac_u \in \begin{cases} AC(t_\varphi) & t_\varphi = t_0 \\ AC^1(t_\varphi) & t_\varphi \neq t_0 \end{cases}$$

$$z_u(t_\varphi) = 0, 1,$$

$$\forall Ac_u \in \begin{cases} AC(t_\varphi) & t_\varphi = t_0 \\ AC^1(t_\varphi) & t_\varphi \neq t_0 \end{cases} \tag{12}$$

$$xx_l(t_\varphi) = 0, 1, \tag{13}$$

$$\forall Ev_l(t_\varphi) \in EV(t_\varphi)$$

$$T_{lu}(t_\varphi) = \min_{n_i=Nv_l(t)} \sum_{n_i=Nv_l(t)}^{n_0^u} T_{i,i+1}(t_i)_{t_\varphi},$$

$$\forall Ev_l(t_\varphi) \in EV(t_\varphi), \tag{14}$$

$$\forall Ac_u \in \begin{cases} AC(t_\varphi) & t_\varphi = t_0 \\ AC^1(t_\varphi) & t_\varphi \neq t_0 \end{cases}$$

$$\text{s.t. } t_i = \begin{cases} t_{i-1} + T_{i-1,i}(t_{i-1})_{t_0} & n_i = n_1^l, \dots, n_0^u \\ t_0 & n_i = Nv_l(t) \end{cases} \tag{15}$$

$$(Nv_l(t), n_1^l), (n_1^l, n_2^l), \dots, (n_i, n_{i+1}), \dots, (n_{-1}^u, n_0^u), (n_0^u, Nc_u(t)) \in E \tag{16}$$

$$Nv_l(t), n_1^l, \dots, n_i, \dots, n_0^u, Nc_u(t) \in N \tag{17}$$

$$n_1^l \neq \dots \neq n_i \neq \dots \neq n_0^u \tag{18}$$

$$(n_1^l, n_2^l), \dots, (n_i, n_{i+1}), \dots, (n_{-1}^u, n_0^u) \neq (n_0^u, n_1^u), \tag{19}$$

$$Ac_u \neq Ac_{u'}.$$

Formula (6) is the objective function of emergency vehicle dispatching. It consists of two parts, the total travel time for emergency vehicles  $Ev_l$  to arrive at accidents  $Ac_u$  and the

punishment caused by the exceeding rescue time. In which,  $M$  is a huge constant. At the initial decision-making instant  $t_\varphi = t_0$ , every accident  $Ac_u \in AC(t_0)$  needs to be rescued. At

other decision-making instants, accident set  $AC^1(t_\varphi)$  includes new accidents that happened at  $t_\varphi$  and the accidents whose rescue paths at  $t_{\varphi-1}$  are affected by new accidents. Emergency vehicle set  $EV(t_\varphi)$  contains all emergency vehicles.

Formulas (7) are the emergency vehicle requirements constraints. They make sure that the vehicle requirements of each accident can be satisfied.

Formulas (8) are time window constraints of accidents. They guarantee that the latest arrival time  $t_{\max}^u(t_\varphi)$  of emergency vehicle does not exceed the upper limit of the time window  $T_{\max}^u(t_\varphi)$ .

Formulas (9) are constraints for the state of emergency vehicles. The emergency vehicle  $Ev_l$  can only be dispatched to an accident  $Ac_u$  or in an idle state.

Formula (10) is the constraint for the total number of emergency vehicles. It ensures that the total number of emergency vehicles dispatched to the accidents and in the idle state is  $L$ .

Formulas (11) are constraints for the state of variables  $x_{lu}(t_\varphi)$ . At the decision-making instant  $t_\varphi$ , if the emergency vehicle  $Ev_l$ ,  $l = 1, 2, \dots, L$ , is dispatched to the accident  $Ac_u$ ,  $u = 1, 2, \dots, U$ , then  $x_{lu}(t_\varphi) = 1$ ; otherwise,  $x_{lu}(t_\varphi) = 0$ .

Formulas (12) are constraints for the state of variables  $z_u(t_\varphi)$ . At the decision-making instant  $t_\varphi$ , if the latest rescue time for accident  $Ac_u$ ,  $u = 1, 2, \dots, U$ , exceeds the upper limit of rescue time window, then  $z_u(t_\varphi) = 1$ ; otherwise,  $z_u(t_\varphi) = 0$ .

Formulas (13) are constraints for the state of variables  $xx_l(t_\varphi)$ . At the decision-making instant  $t_\varphi$ , if the emergency vehicle  $Ev_l$ ,  $l = 1, 2, \dots, L$ , is in the idle state, then  $xx_l(t_\varphi) = 1$ ; otherwise,  $xx_l(t_\varphi) = 0$ .

Formula (14) is the objective function of rescue path for the accident  $Ac_u(t_\varphi)$ . It minimizes the travel time of the emergency vehicle from node  $Nv_l(t_\varphi)$  to the destination  $Nc_u(t_\varphi)$ , where road sections  $(n_i, n_{i+1})$  and  $(n_{i+1}, n_{i+2})$  are connected.

Formula (15) calculates the instant  $t_i$  when the emergency vehicle enters the road section  $(n_i, n_{i+1})$ ,  $n_i = Nv_l(t), \dots, n_0^u$ .

Formulas (16)–(18) are connectivity constraints of the path. They ensure that there are no loops in the path sequence.

Formulas (19) ensure that the emergency vehicles do not pass road sections with accidents.

#### 4. Solution for the Emergency Vehicle Dispatching Model

The shuffled frog leaping algorithm (SFLA) is a kind of metaheuristic algorithm that imitates the frog population's behavior in obtaining food. The initial frog population is generated and divided into several memplexes. Then the frogs search for the optimal solution within each memplex for the defined number of times. Then frogs in different memplex are shuffled so that the information can be exchanged globally. The group optimization and global information exchange alternate with each other until the convergence condition is satisfied. The mathematical model of SFLA is as follows.

(1) *Initialization.*  $H$  frogs are randomly generated to compose the initial population IP. The position of the  $h$ th frog is

encoded as  $X_h = [x_{h1}, x_{h2}, \dots, x_{hd}, \dots, x_{hD}]$ ,  $h = 1, \dots, H$ , in which,  $D$  is the dimension of the optimization problem. Each  $X_h$  represents a feasible solution to the optimization problem. And each feasible solution corresponds to a performance function  $f(X_h)$  associated with the optimization objective.

(2) *Rank and Grouping.*  $H$  frogs are ranked in descending order of performance function value. Position  $Px = [px_1, px_2, \dots, px_d, \dots, px_D]$  of the optimal frog in the population is marked. The population IP is divided into  $a$  memplexes, and there are  $c$  frogs in each memplex according to

$$M_{o_1} = \{X_{o_1+a(o_2-1)} \in \text{IP} \mid 1 \leq o_2 \leq c\} \quad (1 \leq o_1 \leq a). \quad (20)$$

(3) *Local Search.* Within each memplex, the local optimization process is repeated for the specified number of iterations  $It$ .

(3.1) Positions of the frogs in the memplex, the best and the worst, are marked as  $Pb = [pb_1, pb_2, \dots, pb_d, \dots, pb_D]$  and  $Pw = [pw_1, pw_2, \dots, pw_d, \dots, pw_D]$ , respectively.  $Pw$  is renewed along with  $Pb$  according to

$$Ds_d = \begin{cases} \min [\text{INT}(r \times (pb_d - pw_d)), D_d^{\max}] & pb_d - pw_d \geq 0 \\ \max [\text{INT}(r \times (pb_d - pw_d)), -D_d^{\max}] & pb_d - pw_d < 0, \end{cases} \quad (21)$$

$$d = 1, 2, \dots, D,$$

$$pw'_d = pw_d + Ds_d, \quad d = 1, 2, \dots, D \quad (22)$$

$$pw'_d = \begin{cases} Z_d^{\max} & pw'_d > Z_d^{\max} \\ pw'_d & Z_d^{\min} \leq pw'_d \leq Z_d^{\max} \\ Z_d^{\min} & pw'_d < Z_d^{\min}, \end{cases} \quad d = 1, 2, \dots, D \quad (23)$$

in which,  $r$  is a random number,  $r \in [0, 1]$ .  $Ds_d$  is the adjustment of the  $d$ th decision variable,  $d = 1, 2, \dots, D$ .  $D_d^{\max}$  is the maximum adjustment of the  $d$ th decision variable.  $pw'_d$  is the renewed position of the  $d$ th decision variable.  $Z_d^{\max}$  and  $Z_d^{\min}$  are the upper and lower limits of the position of the  $d$ th decision variable.

(3.2) If the performance value of  $Pw' = [pw'_1, pw'_2, \dots, pw'_d, \dots, pw'_D]$  is better than  $Pw$ , then  $Pw = Pw'$ ; otherwise,  $Pb$  in formula (21) is replaced with  $Px$ , and the position updating is executed repeatedly.

(3.3) If the performance value of  $Pw$  is still better than  $Pw'$ , then  $Pw$  is substituted by a random frog location.

(4) *Mix and Global Search.* After a local search, all memplexes are shuffled to form a new population. Frogs are ranked and the best frog  $Px$  is marked. Then the next grouping and local search are performed until the specified number of global iterations  $IT$  is completed.

4.1. *Solution for the Emergency Vehicle Dispatching Model Based on Improved Shuffled Frog Leaping Algorithm.* In order to solve the dynamic emergency vehicle dispatching model, an improved shuffled frog leaping algorithm (ISFLA) is

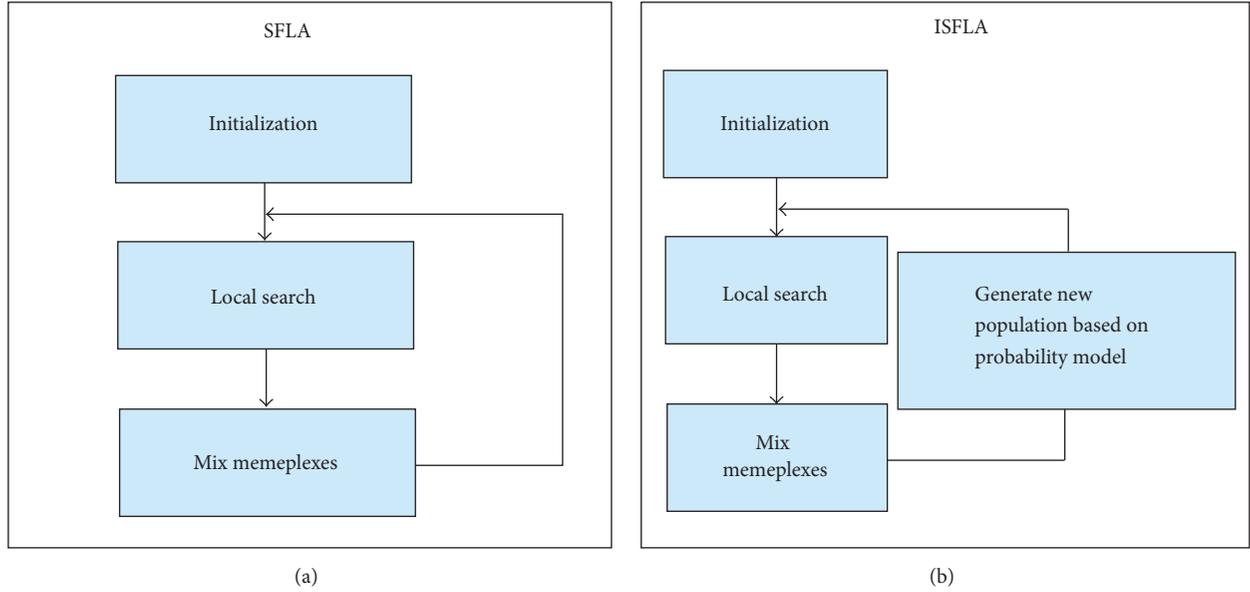


FIGURE 3: Principle of the ISFLA.

proposed. As shown in Figure 3, the improved algorithm is guided by the idea of a base SFLA. The local search uses the optimization strategy of SFLA, and the global search uses the probability model of estimation of distribution algorithms (EDA) to avoid the partial optimal solution.

**4.1.1. Encoding and Decoding.** Decision variables of the emergency vehicle dispatching are 0-1 numerical variables  $x_{lu}(t_\varphi)$ , which express whether the vehicle  $Ev_l(t_\varphi)$  should be dispatched to the accident  $Ac_u(t_\varphi)$ . The working object of SFLA is integer vector, so it is necessary to convert  $x_{lu}(t_\varphi)$  to integer variable  $x_l(t_\varphi)$ .  $x_l(t_\varphi)$  represents the dispatching strategy of  $Ev_l(t_\varphi)$ ,  $l = 1, 2, \dots, L$  at the decision-making instant  $t_\varphi$ . Each vehicle can go to one of the accidents  $Ac_u(t_\varphi)$ ,  $u = 1, 2, \dots, U$  or in the idle state. The feasible set of  $x_l(t_\varphi)$  is  $x_l(t_\varphi) = \{0, 1, 2, \dots, U\}$ . The emergency vehicle is an idle vehicle while  $x_l(t_\varphi) = 0$ . In this case, the encoding of frog position can be expressed as a row vector matrix.

$$X(t_\varphi) = [x_1(t_\varphi), x_2(t_\varphi), \dots, x_l(t_\varphi), \dots, x_L(t_\varphi)]. \quad (24)$$

Similarly,  $x_l(t_\varphi)$  should be decoded in the reverse way of encoding, and we can get the emergency vehicle dispatching strategy.

**4.1.2. Performance Function.** According to the objective function of the dynamic dispatching model of emergency vehicles, the performance function is defined as

$$f(X(t_\varphi)) = - \left\{ \sum_u \sum_l As_u \times T_{lu}(t_\varphi) \times x_{lu}(t_\varphi) + \sum_u M \times y_u(t_\varphi) + \sum_u M \times z_u(t_\varphi) \right\}$$

$$y_u(t_\varphi) = \left| Na_u(t_\varphi) - \sum_u x_{lu}(t_\varphi) \right|$$

$$z_u(t_\varphi) = \max \{0, t_{\max}^u(t_\varphi) - T_{\max}^u(t_\varphi)\},$$

$$t_{\max}^u(t_\varphi) = \max \{T_{lu}(t_\varphi) \times x_{lu}(t_\varphi)\}. \quad (25)$$

Using the quantification method in the literature [30], the accident severity  $As_u$  is divided into four levels, and the value is shown in Table 1.

**4.1.3. Process of the Improved Shuffled Frog Leaping Algorithm.** Process of the ISFLA is shown in Figure 4, and the specific process is as follows.

(1) *Build Probability Model.* The decision vector of emergency vehicle dispatching is  $X(t_\varphi) = [x_1(t_\varphi), x_2(t_\varphi), \dots, x_l(t_\varphi), \dots, x_L(t_\varphi)]$ ,  $x_l(t_\varphi) = \{0, 1, 2, \dots, U\}$ .  $U \times L$  dimensional matrix  $B_{U \times L}$  is used to describe the probability distribution of the frog population.  $b_{u \times l} \in [0, 1]$  is the element of the matrix  $B_{U \times L}$ . It represents the probability that the  $l$ th decision variable  $x_l(t_\varphi)$  is valued as  $u$ .

(2) *Generate the Initial Population.* The value of the element  $b_{u \times l}$  is set as  $1/U$ . According to a uniform distribution,  $H$  frogs are generated randomly to form the initial population IP. Each frog's position  $X_h = [x_{h,1}, x_{h,2}, \dots, x_{h,L}]$  represents a feasible decision vector.

(3) *Local Search.*  $H$  frogs are ranked in descending order of performance function value  $f(X_h)$ . Location  $P_x$  of the optimal frog in the population is marked. The population IP is divided into  $a$  memeplexes, and there are  $c$  frogs in each memeplex according to formula (20). Within each memeplex,

TABLE 1: Value of accident severity.

Accident level		Description of the accident			Property loss (L)	Accident severity
		Death toll	The number of serious injuries (SI)	The number of slight injuries		
Minor accident	Situation 1	-	-	1-2	-	40
	Situation 2	-	-	-	Motor vehicle accidents: <1,000 Nonmotor vehicle accidents: <200	
Ordinary accident	Situation 1	-	1-2	-	-	60
	Situation 2	-	-	$\geq 3$	-	
	Situation 3	-	-	-	<30,000	
Major accident	Situation 1	1-2	-	-	-	80
	Situation 2	-	$3 \leq SI < 10$	-	-	
	Situation 3	-	-	-	$30,000 \leq L < 60,000$	
Extra serious accident	Situation 1	$\geq 3$	-	-	-	100
	Situation 2	-	$\geq 11$	-	-	
	Situation 3	1	$\geq 8$	-	-	
	Situation 4	2	$\geq 5$	-	-	
	Situation 5	-	-	-	$\geq 60,000$	

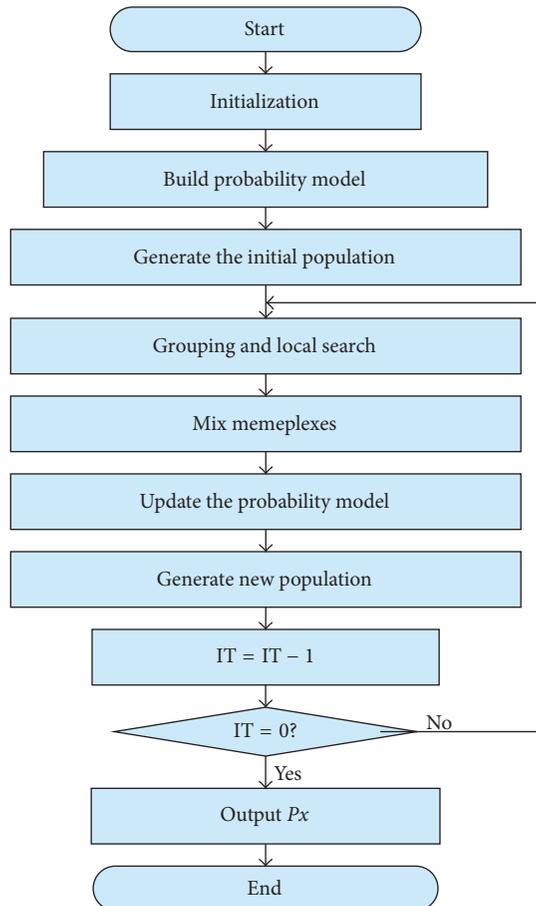


FIGURE 4: Process of the improved SFLA.

the local optimization process is repeated for the specified number of iterations  $It$  according to formula (21)–(23).

(4) *Select the Superior Individual and Update the Probability Model.* The performance function value  $f(X_h)$  of each frog in the population is calculated.  $H$  frogs are reordered in descending order of performance function value. The first  $H/2$  frogs are selected to calculate the distribution of decision variables, and then the probability matrix  $B'$  is obtained. The updated probability model is

$$B = \theta \cdot B' + (1 - \theta) \cdot B, \quad (26)$$

in which,  $\theta$  is the forgetting factor.

(5) *Generate New Population.* According to the updated probability model  $B_{U \times L}$ ,  $H$  frogs are generated to form a new population IP.

## 5. Illustrative Examples

Beijing expressway network shown as in Figure 1 was taken as an example to demonstrate the efficacy of the model and the improved algorithm. The interested time horizon  $Q$  was 9:30 to 10:30 on December 1, 2016. The minimum time interval  $\kappa$  was 5 minutes. Real link travel speeds at the instant  $t \in \{9:30, 9:35, 9:40, \dots, 10:30\}$  are given in Table 2. The upper limit of the time window  $T_{\max}^u$  was 50 minutes. Firstly, based on real link travel speeds at the instant  $t \in \{9:30, 9:35, 9:40, \dots, 10:30\}$ , five examples with different scale variable were used to test the performance of the emergency vehicle dispatching based on ISFLA. Secondly, based on prediction link travel speeds with different error ranges, two examples were used to test the performance of the dynamic emergency vehicle dispatching based on the polygonal-shaped travel speed function. Thirdly, we simulated a scene of a new accident to demonstrate the efficacy of adjustment mechanism of dispatching strategy.

### 5.1. Illustrative Examples of Emergency Vehicle Dispatching Based on the Improved Shuffled Frog Leaping Algorithm

(1) *Calculating Link Travel Time Time-Dependent Function.* Real link travel speeds  $v_{ij}^1(t)$  at instant  $t \in \{9:30, 9:35, 9:40, \dots, 10:30\}$  were used to build the polygonal-shaped travel speed function. Then the travel time function  $T_{ij}(t)$ ,  $t \in [9:30, 10:30]$ , of each road section was calculated according to formula (5). Figure 5 shows travel speed time-dependent functions of road sections  $(n_1, n_2)$ ,  $(n_2, n_1)$ ,  $(n_{18}, n_{32})$ ,  $(n_{32}, n_{18})$ , and Figure 6 shows corresponding travel time time-dependent functions of these road sections.

(2) *Parameters of Illustrative Examples of Emergency Vehicle Dispatching.* We assume that there are 15 emergency vehicles at most in the expressway network. At 9:30, 7 accidents at most occurred simultaneously. Tables 3 and 4 list the parameters of the example.

(3) *Calculating the Shortest Travel Time for Each Emergency Vehicle to Each Accident.* We assume 5 emergency vehicle

dispatching examples with different complexity. The first example has 8 emergency vehicles and 3 accidents. Then, the numbers of emergency vehicles and accidents gradually increase, and the other four examples are formed. Emergency vehicles and accidents in each example are shown in Table 5.

The dynamic shortest path algorithm of emergency vehicles in expressway network proposed in our previous research [18] was used to calculate the shortest travel time and path, and the results are shown in Tables 6 and 7.

The expressway network is a directed network. When the emergency vehicle  $Ev_l$ ,  $l = 1, 2, \dots, L$ , goes to the accident  $Ac_u$ ,  $u = 1, 2, \dots, U$ , it must run along the direction of the road section to arrive at the node  $n_1^l$  and then choose the next road section. In which,  $n_1^l$  is the end of the road section  $(n_0^l, n_1^l)$  where the emergency vehicle  $Ev_l$  is located. Road section from the emergency vehicle's location  $Nv_l$  to node  $n_1^l$  is a necessary way for emergency vehicle  $Ev_l$ . In the same way, the emergency vehicle must arrive at the node  $n_0^u$  and then run along the direction of the road section to reach the accident node  $Nc_u$ . In which,  $n_0^u$  is the start of the road section  $(n_0^u, n_1^u)$  where the accident  $Ac_u$  is located. Road section from the node  $n_0^u$  to the accident's location  $Nc_u$  is a necessary way for emergency vehicle  $Ev_l$  too. We only need to calculate the optimal path from the node  $n_1^l$  to the node  $n_0^u$ , so the optimal path from the node  $n_1^l$  to the node  $n_0^u$  is given in Table 7.

(4) *Calculating the Optimal Dispatching Strategy.* ISFLA was used to find the optimal dispatching strategy of the five examples. The improved algorithm was compared with SFLA. After some testing, parameters of ISFLA were defined as in Table 8. After ten runs, the best solutions of the five examples obtained by using these two algorithms are summarized in Table 9. Figure 7 shows the evolutionary processes of the two algorithms of example I and example V.

(5) *Result Analysis.* The applicability of the model and the improved algorithm for solving the emergency vehicle dispatching problems with different complexity were verified by the five illustrative examples. Under the constraints of emergency vehicle requirements and time windows, the optimal dispatching strategy takes accident severity as key factor to optimize the total travel times of emergency vehicles. The results correspond to the decision rules of emergency vehicle dispatching.

In example I, there are  $4^8$  possible emergency vehicle selection strategies. Analyzing Table 9 shows that SFLA and ISFLA get the same optimal selection strategies. Their objective function values are 7987.

In example II, there are  $5^{10}$  possible emergency vehicle selection strategies. Analyzing Table 9 shows that the objective function of optimal selection strategy obtained by using ISFLA is 8113. The SFLA is unwanted early convergence. Its objective function value of optimal selection strategies is 13.14% larger than that of ISFLA.

In example III, there are  $6^{12}$  possible emergency vehicle selection strategies. Analyzing Table 9 shows that the objective function of optimal selection strategy obtained by using ISFLA is 10247. The SFLA is unwanted early convergence.

TABLE 2: The table gives travel speeds of  $(n_1, n_2)$  at 9:30, 9:35, 9:40, 9:45, 9:50, 9:55, 10:00, 10:05, 10:10, 10:15, 10:20, 10:25, 10:30 on December 1, 2016. In which,  $(n_1, n_2)$  is the road section of Beijing expressway network. They are the basic data of illustrative examples in Section 5.

Road section	9:30	9:35	9:40	9:45	9:50	9:55	10:00	10:05	10:10	10:15	10:20	10:25	10:30
$(n_1, n_2)$	35.8215	20.3765	28.2279	31.8677	35.7043	38.8095	28.6015	17.5589	22.8039	20.5689	33.2365	23.6908	26.1908
$(n_1, n_6)$	22.5825	33.7016	14.5643	26.573	27.6807	19.0503	22.7254	23.13	22.2	21.6732	11.8164	23.3327	26.3327
$(n_1, n_7)$	19.7129	20.9999	15.7227	20.0905	17.6481	23.1212	26.4123	24.617	21.1975	17.4778	29.8401	32.2002	27.2002
$(n_1, n_9)$	15.355	15.528	11.5561	20.1183	19.7576	17.6415	22.0847	14.5041	14.1869	15.4142	16.9405	20.3074	22.5378
$(n_2, n_1)$	23.1756	27.1385	29.4663	30.5464	19.6817	24.3868	17.9294	18.4325	23.9658	31.6632	20.1517	25.8569	24.6569
$(n_2, n_3)$	26.4831	27.1754	21.7814	24.0784	17.1708	23.5244	17.2602	26.6113	35.9275	19.6655	23.9367	22.5875	17.5875
$(n_2, n_{10})$	16.693	13.9778	11.2268	18.5879	15.5855	17.1857	13.8282	10.7979	18.6863	17.7514	22.7961	21.2068	16.2068
$(n_3, n_2)$	25.0379	30.4431	22.1758	18.9887	28.441	27.0197	27.0197	33.0966	27.4579	26.9891	33.9385	21.9777	26.9777
$(n_3, n_4)$	21.4602	30.1089	16.8812	11.7818	11.621	26.9056	21.581	39.804	35.43943	28.224	21.4316	34.3545	29.3545
$(n_3, n_{12})$	23.7186	13.5768	22.6461	28.554	22.1646	13.8365	19.1874	18.0001	21.5321	21.6481	21.622	18.5449	23.5449
$(n_4, n_3)$	15.4424	20.2254	30.8515	25.626	23.4888	22.7928	25.596	25.40945	41.652	34.4304	25.5643	19.6756	22.6436
$(n_4, n_5)$	30.2678	19.3306	39.0199	32.8014	27.6075	30.1831	27.904	39.059	11.088	15.9983	27.7148	25.7315	28.6734
$(n_4, n_{13})$	20.139	21.9476	18.3451	21.4544	12.4372	24.3021	13.8022	9.6058	25.6725	18.4364	20.9946	16.9337	11.9337
$(n_5, n_4)$	9.3128	23.4184	15.3372	48.2641	31.6018	36.0318	16.5346	28.1376	28.1376	22.659	23.2064	25.022	27.0654
$(n_5, n_6)$	35.7092	42.1002	34.1108	37.9314	35.7767	28.5644	32.7456	25.1776	36.9073	22.659	23.2064	25.022	27.0654
$(n_5, n_{15})$	18.1438	22.007	25.7119	21.4686	15.3884	22.5308	15.1383	16.1952	17.7313	14.853	19.6452	16.5002	11.5002
$(n_6, n_1)$	15.6426	36.723	22.8957	30.8007	25.6438	32.4282	25.1974	49.986	33.9197	35.9149	21.4228	37.6876	32.6876
$(n_6, n_5)$	25.1075	31.5886	11.1249	26.73	24.013	15.8807	20.2512	24.1323	40.3112	36.2664	15.6697	26.0462	21.0462
$(n_6, n_{18})$	24.2849	18.1288	17.2682	25.7382	30.8254	14.5333	20.753	22.5785	13.04	27.2062	20.9267	32.5609	27.5609
$(n_7, n_1)$	33.1513	25.6162	28.3083	24.1654	28.8115	17.0718	25.6522	24.2943	26.9422	30.11469	29.6986	38.7776	33.7776
$(n_7, n_8)$	45.289	42.7567	39.8016	45.4383	43.9181	54.312	36.4771	36.7876	46.5368	53.676	29.1054	41.8043	36.8043
$(n_7, n_{18})$	24.2919	26.4158	27.8619	15.4707	24.595	9.864	24.684	7.38	22.212	32.076	21.8051	19.9445	23.3452
$(n_8, n_7)$	45.2188	42.6165	48.6611	48.0486	34.7927	42.0564	32.445	55.548	40.0004	29.4257	47.3986	39.5606	34.5606
$(n_8, n_9)$	24.8046	34.816	22.4496	28.5366	24.6233	16.668	12.8682	7.38	19.1072	28.6751	24.6707	18.2876	23.2876
$(n_8, n_{19})$	23.6014	17.8158	18.7145	22.1506	22.0417	18.6709	23.9033	29.326	21.7442	21.9771	11.9924	19.1274	17.1434
$(n_9, n_1)$	25.5525	22.7526	26.5187	19.5367	34.5426	25.6112	30.2352	23.8654	25.1875	29.4503	28.7788	20.3987	15.3987
$(n_9, n_8)$	31.4897	36.8907	32.8745	42.5121	33.336	34.4962	42.436	38.276	31.005	29.628	26.215	31.2124	26.2124
$(n_9, n_{10})$	12.4945	15.9981	11.0279	18.8998	12.4372	25.544	14.796	22.2355	10.6805	15.7968	12.1055	28.9159	23.9159
$(n_9, n_{20})$	26.3711	18.6249	20.6505	13.2397	22.2702	19.9457	12.22	18.1987	15.747	18.5462	16.7472	17.0505	12.0505
$(n_{10}, n_2)$	23.482	19.0404	27.7911	20.6294	22.1208	27.5455	18.4591	16.173	30.329	29.8818	23.7257	29.2399	34.2399
$(n_{10}, n_9)$	19.6502	21.1314	19.5799	16.8456	17.5475	29.628	21.7406	12.8068	26.6456	22.2932	27.195	24.6179	22.3426
$(n_{10}, n_{11})$	11.6789	25.0173	23.1734	15.8519	10.4692	15.7769	17.2203	19.3803	17.6942	21.2171	20.9595	12.7466	17.7466
$(n_{10}, n_{21})$	16.6906	23.4181	21.1797	23.4837	21.5237	16.9955	16.4308	22.9818	15.644	24.9333	22.9535	20.4386	15.4386
$(n_{11}, n_{10})$	21.1883	19.3399	21.2697	30.4105	18.6214	20.5809	19.44	17.1289	23.0388	22.6296	31.6504	22.7549	27.7549
$(n_{11}, n_{12})$	29.7455	23.1488	36.2439	41.5484	32.6593	48.132	30.6737	32.5309	32.5309	26.2273	31.0559	33.5541	28.5541
$(n_{11}, n_{22})$	19.6753	16.1579	22.8854	19.7345	28.2388	23.4119	33.845	32.5529	25.8201	23.5862	30.7608	19.2675	23.2453
$(n_{12}, n_3)$	31.2132	24.0045	23.2547	21.9237	29.9919	39.1563	30.9103	24.9077	32.3465	24.4867	23.3021	31.8698	29.4568
$(n_{12}, n_{11})$	33.088	35.2858	30.6669	28.0342	25.4681	38.1852	29.059	32.6558	38.6695	32.2736	27.1765	33.8246	28.8246
$(n_{12}, n_{13})$	15.7673	18.1255	24.3937	18.8885	22.8062	14.7927	21.5846	19.7057	16.6918	25.0319	24.1279	23.1339	25.3462
$(n_{12}, n_{23})$	15.5967	22.1842	30.6917	11.3833	18.6417	10.7881	24.5125	25.0496	9.6873	27.7184	12.5765	19.1341	14.1341
$(n_{13}, n_4)$	22.9506	21.4434	25.3072	15.7694	23.0083	22.2823	20.4405	28.2635	21.3286	27.7032	29.8792	24.2156	29.2156
$(n_{13}, n_{12})$	21.9029	30.2519	25.8323	13.7023	33.4922	27.0706	25.3517	29.1525	26.6141	14.5851	11.252	22.3766	17.3766

TABLE 2: Continued.

Road section	9:30	9:35	9:40	9:45	9:50	9:55	10:00	10:05	10:10	10:15	10:20	10:25	10:30
(n <sub>13</sub> , n <sub>14</sub> )	11.1064	25.6402	15.2103	19.3209	26.4223	30.3127	13.0589	32.0853	27.2811	25.506	21.4571	20.4382	15.4382
(n <sub>13</sub> , n <sub>24</sub> )	15.4208	17.5974	24.8324	25.0641	21.4277	24.1301	23.3536	17.0912	21.1445	17.5043	14.5007	22.1737	27.1737
(n <sub>14</sub> , n <sub>13</sub> )	23.6147	38.1495	28.9343	24.2838	40.8111	24.048	28.2018	45.36	25.2878	27.65	23.2166	21.039	26.039
(n <sub>14</sub> , n <sub>15</sub> )	25.379	22.6816	16.5716	19.1937	21.2507	25.92	30.8738	16.8351	26.708	23.8517	20.5968	15.6582	20.6582
(n <sub>14</sub> , n <sub>27</sub> )	21.3262	21.9084	24.3039	28.8116	24.2594	30.2419	11.8582	19.7537	23.18	17.8241	15.6202	25.0852	20.0852
(n <sub>15</sub> , n <sub>5</sub> )	21.8971	11.6351	30.1314	25.5666	26.0133	29.028	19.9033	23.3528	32.7366	29.9893	20.5936	15.9837	21.2351
(n <sub>15</sub> , n <sub>14</sub> )	22.5953	26.6376	28.3237	25.4259	21.0793	19.422	15.9731	25.92	26.6161	46.296	29.4048	34.293	36.3256
(n <sub>15</sub> , n <sub>29</sub> )	24.8356	29.7005	35.0005	13.6867	36.468	22.2797	20.34	24.7755	11.088	33.676	30.3817	27.5295	32.5295
(n <sub>16</sub> , n <sub>15</sub> )	23.5872	31.4391	20.8336	9.2175	28.447	26.4114	17.0868	21.1131	23.8973	21.9235	23.4818	19.4364	14.4364
(n <sub>16</sub> , n <sub>17</sub> )	28.7868	28.331	27.7535	22.1266	39.8588	36.296	19.422	31.4111	25.0175	26.1084	44.1895	19.9332	25.3457
(n <sub>16</sub> , n <sub>30</sub> )	23.5312	20.7604	22.3994	35.63	34.2775	19.7144	21.6727	27.5377	26.1173	20.34	25.3507	30.9571	35.9571
(n <sub>17</sub> , n <sub>16</sub> )	26.7026	16.7753	19.1615	25.0103	22.2606	30.4504	30.7058	21.096	28.3757	30.9772	25.6049	32.1169	27.1169
(n <sub>17</sub> , n <sub>18</sub> )	16.9842	34.7854	32.1204	28.8798	29.7914	26.69	25.211	36.1121	26.7128	39.1083	29.6514	27.4658	22.4658
(n <sub>17</sub> , n <sub>18</sub> )	26.443	28.7525	32.2085	27.9107	33.5474	42.424	33.1352	44.132	29.628	21.285	38.88	22.4163	27.4163
(n <sub>17</sub> , n <sub>31</sub> )	20.0635	18.4469	18.9766	12.6472	14.5116	16.4666	13.5609	18.6594	28.8261	14.5888	20.2372	11.8279	6.8279
(n <sub>18</sub> , n <sub>6</sub> )	28.5716	27.6879	36.1665	27.9511	27.4823	33.0515	30.2954	22.6653	21.871	20.1762	43.026	34.8619	38.3462
(n <sub>18</sub> , n <sub>7</sub> )	17.1967	25.6923	22.2066	30.8889	40.716	15.114	12.96	5.544	25.92	13.0093	18.7247	23.7387	28.7387
(n <sub>18</sub> , n <sub>17</sub> )	32.0776	31.2593	49.3007	31.5098	56.466	46.3468	24.048	24.048	41.652	23.13	23.3579	31.7037	36.7037
(n <sub>18</sub> , n <sub>32</sub> )	21.765	13.6177	24.1532	20.6376	22.2606	17.6057	20.226	26.3626	7.6245	10.8335	21.0773	15.7262	17.5275
(n <sub>19</sub> , n <sub>8</sub> )	30.0455	34.0459	26.2879	28.8757	26.1564	33.9629	38.5092	29.0318	34.1819	29.4782	32.5105	37.9813	36.3864
(n <sub>19</sub> , n <sub>20</sub> )	9.971	19.224	11.088	24.048	14.796	24.048	21.3077	28.6925	20.7057	10.2956	16.9843	15.5002	10.5002
(n <sub>19</sub> , n <sub>33</sub> )	37.9415	31.3179	23.2697	28.7727	26.4532	38.88	27.6162	26.532	46.296	32.088	34.3308	27.6593	22.6593
(n <sub>19</sub> , n <sub>34</sub> )	24.5436	17.0723	22.9667	30.4918	19.9406	17.1968	15.9998	23.2988	27.0237	31.7541	19.3243	19.9782	20.2351
(n <sub>20</sub> , n <sub>9</sub> )	16.0336	21.0688	25.6279	30.6163	31.9925	22.4156	21.399	27.0876	25.4508	26.8263	26.8914	15.5985	20.5985
(n <sub>20</sub> , n <sub>19</sub> )	26.5774	22.212	17.5736	27.2318	15.442	13.6652	26.5167	17.5446	24.1079	30.6512	24.6187	27.79	32.79
(n <sub>20</sub> , n <sub>21</sub> )	31.4044	31.4625	47.808	37.008	40.1212	35.172	42.588	23.882	33.7562	36.6223	31.0834	29.3702	24.3702
(n <sub>21</sub> , n <sub>10</sub> )	19.8354	23.3835	28.8326	33.7234	19.1974	36.8871	32.6304	25.8425	23.494	21.5067	29.1639	21.527	26.527
(n <sub>21</sub> , n <sub>20</sub> )	28.3482	29.4714	19.5844	21.8987	25.765	28.5483	33.6437	26.5502	35.3078	31.3771	34.8706	29.6347	32.5326
(n <sub>21</sub> , n <sub>22</sub> )	26.0216	27.5469	21.8179	32.4478	33.4188	21.0689	23.4394	28.1423	24.1956	32.7357	29.0167	20.4584	29.5815
(n <sub>21</sub> , n <sub>35</sub> )	22.5705	28.1672	18.9582	31.5342	35.7092	23.1775	24.9478	28.0398	32.2195	28.1321	27.3823	22.1031	32.4264
(n <sub>22</sub> , n <sub>11</sub> )	32.0161	33.8281	49.6334	31.4378	24.217	41.5755	35.0492	28.7604	45.0867	34.7548	33.471	28.1108	23.1108
(n <sub>22</sub> , n <sub>21</sub> )	21.5587	24.7503	13.8207	28.1799	36.1223	28.797	22.4211	26.1431	19.8201	34.3893	29.7035	28.9367	33.9367
(n <sub>22</sub> , n <sub>23</sub> )	12.8279	35.172	20.358	10.17	23.1605	21.7582	22.824	29.3252	17.6377	28.4508	16.513	25.1189	30.1189
(n <sub>22</sub> , n <sub>36</sub> )	16.8102	19.7036	20.046	17.7061	18.0843	19.2021	21.7497	26.6898	22.7205	17.28	22.5881	17.5268	22.5268
(n <sub>23</sub> , n <sub>12</sub> )	26.3685	19.6544	20.9984	19.641	25.8781	19.1095	17.6822	25.7649	20.8829	24.8079	21.2406	24.3477	29.3477
(n <sub>23</sub> , n <sub>22</sub> )	16.8402	20.5886	16.65	29.8227	21.1968	19.2456	17.586	16.668	30.4934	20.9667	24.2786	25.0395	20.0395
(n <sub>23</sub> , n <sub>24</sub> )	29.2047	24.7494	18.3338	26.4002	14.7043	28.692	24.4668	11.088	20.34	25.92	27.8506	24.2783	19.2783
(n <sub>23</sub> , n <sub>37</sub> )	23.8144	22.3499	18.5552	20.8287	21.6968	11.4653	20.4559	26.4973	18.5754	24.9654	27.9486	28.4628	33.4628
(n <sub>24</sub> , n <sub>13</sub> )	30.4946	25.7359	36.4783	38.7628	25.6916	15.4647	23.175	20.7594	21.1762	19.3615	25.3909	35.5331	30.5331
(n <sub>24</sub> , n <sub>23</sub> )	36.1943	31.331	37.8856	28.1583	27.6649	37.044	39.9093	25.92	24.432	27.98	34.4391	29.0487	24.0487
(n <sub>24</sub> , n <sub>25</sub> )	27.5556	25.9729	24.751	29.2513	28.6489	25.1988	20.5482	25.7422	30.9605	16.9408	24.8777	15.2558	20.2558
(n <sub>24</sub> , n <sub>38</sub> )	26.3805	23.4738	24.2851	21.887	18.8264	21.3668	22.3002	12.6113	22.0732	26.5655	20.3389	30.4344	35.4344

TABLE 2: Continued.

Road section	9:30	9:35	9:40	9:45	9:50	9:55	10:00	10:05	10:10	10:15	10:20	10:25	10:30
(n <sub>25</sub> , n <sub>24</sub> )	25.331	28.2618	26.6353	30.4132	31.0304	22.7455	27.2853	21.2781	20.3911	31.6241	27.0002	26.7254	21.7254
(n <sub>25</sub> , n <sub>26</sub> )	16.8309	27.0856	23.9089	19.3628	24.066	34.794	20.89029	46.908	29.004	19.89	25.92	24.7748	29.7748
(n <sub>25</sub> , n <sub>39</sub> )	19.7948	24.4553	22.3961	13.2497	13.9135	29.1571	18.4512	35.4701	9.6681	25.6926	23.0038	18.4626	13.4626
(n <sub>26</sub> , n <sub>25</sub> )	27.6015	29.85	28.5529	18.1117	23.5015	44.424	33.3481	23.598	18.504	34.254	31.464	25.7385	20.7385
(n <sub>26</sub> , n <sub>27</sub> )	14.2224	19.9524	16.5509	16.6665	31.572	24.3207	20.976	20.5671	7.38	12.96	21.9918	24.4198	19.4198
(n <sub>26</sub> , n <sub>40</sub> )	33.5522	19.6369	25.2266	22.8186	25.8973	13.3329	34.1434	28.2749	37.0596	33.5	25.0689	19.3657	19.2819
(n <sub>26</sub> , n <sub>14</sub> )	30.6965	22.8707	28.1807	28.9659	20.5081	25.7812	32.9863	25.5622	23.4569	27.2833	23.6245	27.2157	32.2157
(n <sub>27</sub> , n <sub>26</sub> )	29.3327	37.7083	34.5393	35.3089	27.3833	33.152	55.548	36.1578	25.9193	31.8867	18.8907	25.4497	25.3257
(n <sub>27</sub> , n <sub>28</sub> )	34.5428	33.1044	26.3141	38.216	36.9792	19.828	37.179	26.652	21.296	28.5949	40.3886	37.2857	42.2857
(n <sub>27</sub> , n <sub>41</sub> )	22.6554	18.3384	20.6647	27.7224	20.0416	21.7215	15.3789	14.2179	21.1639	30.3106	17.1591	12.9237	14.3215
(n <sub>28</sub> , n <sub>27</sub> )	24.7294	13.4349	23.6613	48.204	18.504	23.672	35.172	25.92	15.732	7.38	23.5587	20.7311	25.7311
(n <sub>28</sub> , n <sub>29</sub> )	27.7343	45.2357	46.278	48.204	18.504	23.672	35.172	25.92	15.732	7.38	23.5587	20.7311	25.7311
(n <sub>28</sub> , n <sub>42</sub> )	32.757	35.1901	29.7974	36.8838	39.0422	31.2206	43.6835	27.4834	34.6492	28.0919	34.9322	29.0565	24.0565
(n <sub>29</sub> , n <sub>15</sub> )	18.7413	22.9066	30.6725	31.2796	26.8011	27.437	29.1645	18.7048	23.0915	16.8493	23.7545	30.7464	32.7321
(n <sub>29</sub> , n <sub>28</sub> )	16.5841	23.1357	17.9721	5.544	24.984	26.1818	31.1185	28.4255	21.7963	16.2167	34.7601	26.723	21.723
(n <sub>29</sub> , n <sub>28</sub> )	35.7481	41.2172	50.004	58.632	23.6078	36.4782	55.548	46.296	34.4414	36.9133	41.7204	39.5694	44.5694
(n <sub>29</sub> , n <sub>30</sub> )	24.4832	35.7813	28.6492	24.5854	50.2455	34.5999	38.4233	43.1548	25.8914	38.4789	39.0105	36.4984	31.4984
(n <sub>30</sub> , n <sub>16</sub> )	32.8059	33.4404	29.0517	40.1148	33.0317	36.4324	28.7253	27.6238	32.5452	31.7361	38.9741	36.0303	41.0303
(n <sub>30</sub> , n <sub>29</sub> )	33.3402	32.9426	26.4765	40.716	32.813	22.212	29.628	27.7627	32.9475	25.4733	23.6324	29.8145	24.8145
(n <sub>30</sub> , n <sub>31</sub> )	18.0093	23.7247	25.822	28.7387	24.6348	33.8815	23.5953	27.6376	30.3237	27.4259	31.4048	28.6161	23.6161
(n <sub>30</sub> , n <sub>47</sub> )	29.9271	28.0907	21.013	30.969	20.7049	37.5447	31.1589	29.5889	25.6384	14.9421	20.8261	30.192	35.192
(n <sub>31</sub> , n <sub>17</sub> )	27.3122	23.0705	26.384	22.972	19.9255	23.7923	22.1811	20.1768	30.8551	23.1064	35.6519	20.5192	25.5192
(n <sub>31</sub> , n <sub>17</sub> )	13.3395	29.7268	20.8545	17.9808	23.7303	17.8568	24.8957	22.1967	36.8889	31.6923	13.5156	28.2066	26.3212
(n <sub>31</sub> , n <sub>30</sub> )	29.9233	27.5431	24.8307	20.7151	25.5793	26.5665	19.653	27.9609	23.8885	27.5288	29.6811	23.4767	28.4767
(n <sub>31</sub> , n <sub>32</sub> )	18.2443	21.0437	26.6463	22.7714	18.1332	14.5623	21.7784	22.6731	20.7887	22.4282	17.4275	27.2954	22.2954
(n <sub>31</sub> , n <sub>47</sub> )	34.6978	23.1543	29.0436	24.907	30.5601	28.0568	27.7801	42.7455	24.8498	27.7656	41.4151	22.2665	23.066
(n <sub>32</sub> , n <sub>18</sub> )	20.0793	14.9731	33.293	20.9773	25.5168	27.8158	25.568	20.4764	23.6614	25.3756	33.3948	21.6959	21.3256
(n <sub>32</sub> , n <sub>31</sub> )	22.4128	20.5054	20.34	35.796	14.796	45.548	29.3574	26.5325	26.7919	20.8342	33.6774	31.6365	26.6365
(n <sub>32</sub> , n <sub>33</sub> )	20.3312	24.4924	24.2836	31.2118	30.8979	28.6509	25.9583	22.4449	28.0319	34.8349	30.5642	29.0862	24.0862
(n <sub>32</sub> , n <sub>45</sub> )	34.5595	27.3606	36.8804	38.7388	41.0041	30.1619	38.1456	35.7044	34.9696	43.0544	21.1344	32.4679	27.4679
(n <sub>33</sub> , n <sub>19</sub> )	28.0225	19.644	36.4511	20.716	38.88	20.34	13.672	25.7334	31.2822	23.6385	24.025	29.1403	34.1403
(n <sub>33</sub> , n <sub>32</sub> )	16.0562	29.5635	23.2929	36.0355	39.4106	26.7698	36.1572	34.7219	20.5097	28.7999	30.5667	32.138	28.3215
(n <sub>33</sub> , n <sub>46</sub> )	28.3858	34.4957	41.3754	26.8214	20.2584	29.8203	17.0806	23.3569	26.6312	33.9373	33.6228	24.8346	19.8346
(n <sub>34</sub> , n <sub>19</sub> )	38.7342	42.1347	43.8738	47.8653	39.3315	48.2212	46.5461	55.2707	45.5078	46.0586	38.8005	37.4032	42.4032
(n <sub>34</sub> , n <sub>35</sub> )	45.4	34.8313	33.05829	51.8472	28.38	42.5736	49.98857	43.7484	47.011	41.9231	58.2984	34.9144	32.4512
(n <sub>34</sub> , n <sub>46</sub> )	28.2038	41.6719	31.5473	29.1606	45.1054	36.8593	36.2981	25.9666	34.6039	35.1209	25.4465	30.6452	25.6452
(n <sub>35</sub> , n <sub>21</sub> )	38.3401	35.6249	48.5614	38.4442	40.0985	32.3709	22.9813	29.4015	27.6955	29.0522	36.886	29.1026	24.1026
(n <sub>35</sub> , n <sub>34</sub> )	45.1644	46.6339	42.4113	35.9576	59.1703	28.4833	41.4302	35.7486	41.5471	48.225	46.9663	41.0249	46.0249
(n <sub>35</sub> , n <sub>36</sub> )	22.611	27.6803	34.0971	27.4104	16.7638	26.5984	23.4473	22.5357	22.6849	18.7214	22.0752	19.2935	21.2924
(n <sub>36</sub> , n <sub>22</sub> )	17.8407	29.7715	34.5095	28.9844	30.6518	27.3529	19.6213	29.6933	37.7258	43.1677	27.9886	19.6087	24.6087
(n <sub>36</sub> , n <sub>35</sub> )	43.7953	54.4549	61.5546	31.3509	68.5008	36.09	35.0472	66.11143	49.50514	41.346	35.172	57.6846	45.4344
(n <sub>37</sub> , n <sub>37</sub> )	19.3156	15.9166	19.7184	28.5772	26.2822	25.679	30.2439	26.0461	26.2169	33.6578	34.694	44.6195	49.6195

TABLE 2: Continued.

Road section	9:30	9:35	9:40	9:45	9:50	9:55	10:00	10:05	10:10	10:15	10:20	10:25	10:30
( $n_{37}, n_{36}$ )	23.8205	32.8456	23.8767	28.0697	27.765	25.64229	23.42025	25.82571	30.67714	23.35015	32.391	32.8346	37.8346
( $n_{37}, n_{38}$ )	34.1043	48.0765	36.7424	37.1687	44.4156	40.54	41.9454	47.7629	52.3747	42.4163	46.6214	42.0936	47.0936
( $n_{38}, n_{24}$ )	32.6611	14.441	24.6137	33.4089	30.3681	28.0291	30.6215	34.7804	28.9226	23.4482	26.919	26.7533	31.7533
( $n_{38}, n_{37}$ )	44.8318	51.211	52.254	60.2107	65.5371	60.4039	55.0154	64.5239	65.3911	59.9616	63.57	62.3024	67.3024
( $n_{38}, n_{39}$ )	37.1429	49.9001	44.5609	36.0501	40.989	47.9961	38.0558	53.7996	41.8743	51.4709	38.3611	48.5935	45.4235
( $n_{39}, n_{25}$ )	34.3629	28.4779	30.0763	29.3488	32.4467	35.5281	29.2485	33.7507	41.5317	36.5889	35.9516	26.1112	31.1112
( $n_{39}, n_{38}$ )	47.4719	62.2148	68.7616	56.3855	66.4986	64.3643	51.5869	63.1797	57.5995	65.7155	60.4977	44.8243	49.8243
( $n_{39}, n_{40}$ )	51.8907	47.8745	58.5121	49.4962	62.436	42.215	52.2124	45.918	53.8011	31.5118	53.7514	38.3837	42.3321
( $n_{40}, n_{26}$ )	35.4259	28.2369	38.1587	33.4012	23.1637	18.4515	31.297	39.8748	34.087	33.3255	37.1265	24.6396	19.6396
( $n_{40}, n_{39}$ )	48.7455	42.1488	55.2439	60.5484	52.6593	50.6737	52.5309	46.2273	52.0559	54.5541	56.3667	55.4897	50.4897
( $n_{40}, n_{41}$ )	40.428	41.73709	51.84	36.22629	44.79943	56.88655	51.228	40.9608	48.7695	43.925	45.6818	52.9536	52.9536
( $n_{41}, n_{27}$ )	21.5665	32.1018	33.315	38.2749	26.2941	36.7688	32.4286	27.3579	36.645	33.5939	30.5894	43.2638	38.2638
( $n_{41}, n_{40}$ )	49.7784	60.5336	44.939	57.9558	64.3915	59.1353	62.1333	60.4039	47.4822	54.6687	67.7975	57.6513	59.6323
( $n_{41}, n_{42}$ )	46.8069	54.2979	43.1809	44.939	43.9223	39.9223	43.9772	51.3589	46.1985	56.2295	53.273	47.3326	46.3216
( $n_{42}, n_{28}$ )	23.5687	22.0959	23.0359	27.8531	24.22	21.4339	22.3374	17.0648	19.8429	16.522	21.9041	18.4894	17.4394
( $n_{42}, n_{41}$ )	41.7558	50.9238	45.4815	42.6664	58.9132	52.9074	55.1869	59.0718	54.4421	57.4068	58.4637	62.3649	59.3321
( $n_{42}, n_{43}$ )	22.452	37.3327	45.7083	42.5393	44.3089	36.3833	42.152	27.8907	35.9193	35.4497	46.1578	41.8867	36.8867
( $n_{43}, n_{29}$ )	26.551	25.5558	22.89	17.6108	33.3904	42.0009	43.34	27.0469	29.724	27.2055	30.4557	24.4874	19.4874
( $n_{43}, n_{42}$ )	50.6882	50.4938	70.0271	61.6079	51.544	60.1915	52.1194	60.6471	56.3798	69.9701	57.2431	46.103	51.103
( $n_{43}, n_{44}$ )	51.9058	24.5301	43.5541	43.2413	65.1582	32.2113	45.666	53.676	42.588	29.616	47.6023	31.0086	36.0086
( $n_{44}, n_{43}$ )	41.0666	47.6639	50.3153	57.3912	49.8147	42.576	56.672	57.4133	55.8927	59.5123	69.7826	64.2306	69.2306
( $n_{44}, n_{45}$ )	28.6551	34.8394	33.4949	33.7306	45.8684	35.6184	47.1494	33.747	46.7805	34.822	40.4126	46.7659	41.7659
( $n_{44}, n_{47}$ )	19.9468	15.8323	23.3384	13.9004	14.2217	23.352	29.1193	16.6816	15.9695	21.4484	25.4341	24.8079	29.8079
( $n_{45}, n_{32}$ )	24.7969	26.8204	20.6999	26.4952	22.3768	23.7876	20.971	24.3051	47.4192	46.4452	32.8815	30.7235	25.7235
( $n_{45}, n_{44}$ )	37.7425	56.6497	42.6488	44.8774	54.1986	44.5991	55.2349	44.2395	56.1024	53.0347	46.2959	41.0744	40.0421
( $n_{45}, n_{46}$ )	33.3609	41.1878	35.4696	36.3263	38.1141	41.0098	49.226	44.7845	31.1314	26.2193	38.532	25.257	30.257
( $n_{46}, n_{33}$ )	30.9449	30.5055	17.3013	27.1126	24.4442	14.7958	19.0287	25.0807	31.9526	25.6242	28.5735	21.6121	26.6121
( $n_{46}, n_{34}$ )	35.3327	29.628	26.838	32.908	37.332	45.66	44.424	45.1164	37.9687	39.0396	37.2611	40.5312	35.5312
( $n_{46}, n_{45}$ )	44.8368	46.52	38.0651	45.1987	48.8088	44.7385	57.8387	39.1315	39.1035	31.001	52.2839	51.9367	46.9367
( $n_{47}, n_{30}$ )	11.9523	25.8677	29.0142	19.907	24.6575	26.7575	21.3123	24.9181	27.4727	19.3771	30.0109	31.5672	31.5672
( $n_{47}, n_{31}$ )	30.0607	30.0464	30.0704	24.1245	28.2755	12.4826	18.9899	33.2118	22.0356	27.9654	35.0488	20.6665	22.6325
( $n_{47}, n_{44}$ )	23.6209	18.1936	25.7786	21.5246	30.2617	13.5507	22.8474	29.7383	19.0304	18.2511	16.3982	20.3277	25.3277

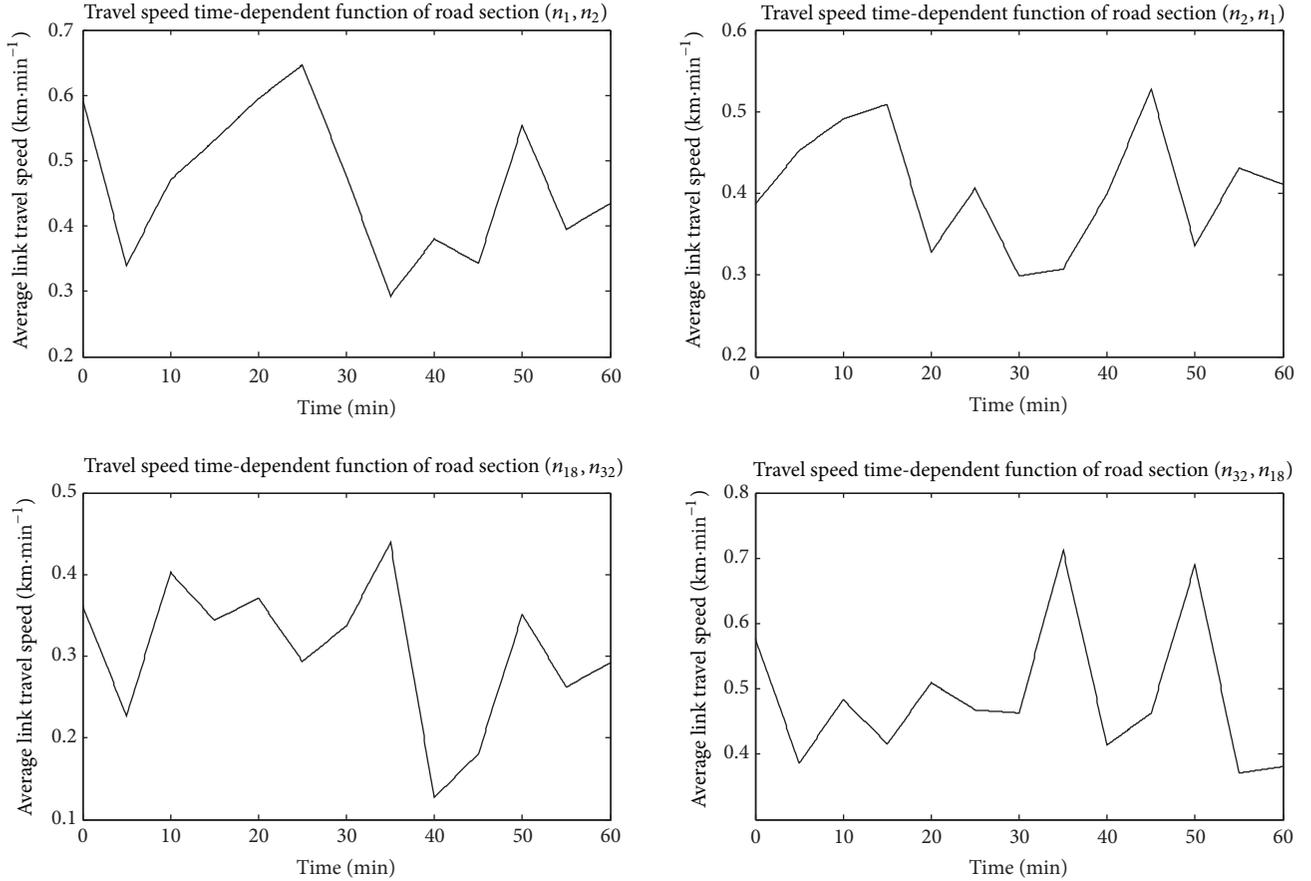


FIGURE 5: Link travel speed time-dependent functions.

TABLE 3: Accident parameters of the illustrative example.

Accident no.	Accident node	Road section of accident	Location of accident	Accident level	Accident severity	Accident demand
1	$Nc_1/n_{48}$	$(n_{21}, n_{22})$	$L_{21,48} = L_{48,22}$	Minor	40	1
2	$Nc_2/n_{49}$	$(n_6, n_5)$	$L_{6,49} = L_{49,5}$	Ordinary	60	2
3	$Nc_3/n_{50}$	$(n_{27}, n_{14})$	$L_{27,50} = L_{50,14}$	Major	80	2
4	$Nc_4/n_{51}$	$(n_{16}, n_{15})$	$L_{16,51} = L_{51,15}$	Minor	40	1
5	$Nc_5/n_{52}$	$(n_9, n_{20})$	$L_{9,52} = L_{52,20}$	Minor	40	2
6	$Nc_6/n_{53}$	$(n_{38}, n_{39})$	$L_{38,53} = L_{53,39}$	Ordinary	60	2
7	$Nc_7/n_{54}$	$(n_{47}, n_{44})$	$L_{47,54} = L_{54,44}$	Minor	40	1

TABLE 4: Emergency vehicle parameters of the illustrative example.

Vehicle no.	1	2	3	4	5
Vehicle node	$Nv_1/n_{55}$	$Nv_2/n_{56}$	$Nv_3/n_{57}$	$Nv_4/n_{58}$	$Nv_5/n_{59}$
Road section of vehicle	$(n_2, n_3)$	$(n_9, n_8)$	$(n_{12}, n_{13})$	$(n_{18}, n_{17})$	$(n_{15}, n_{29})$
Location of vehicle	$L_{2,55} = L_{55,3}$	$L_{9,56} = L_{56,8}$	$L_{12,57} = L_{57,13}$	$L_{18,58} = L_{58,17}$	$L_{15,59} = L_{59,29}$
Vehicle no.	6	7	8	9	10
Vehicle node	$Nv_6/n_{60}$	$Nv_7/n_{61}$	$Nv_8/n_{62}$	$Nv_9/n_{63}$	$Nv_{10}/n_{64}$
Road section of vehicle	$(n_{39}, n_{25})$	$(n_{30}, n_{31})$	$(n_{33}, n_{32})$	$(n_{19}, n_{33})$	$(n_{24}, n_{13})$
Location of vehicle	$L_{39,60} = L_{60,25}$	$L_{30,61} = L_{61,31}$	$L_{33,62} = L_{62,32}$	$L_{19,63} = L_{63,33}$	$L_{24,64} = L_{64,13}$
Vehicle no.	11	12	13	14	15
Vehicle node	$Nv_{11}/n_{65}$	$Nv_{12}/n_{66}$	$Nv_{13}/n_{67}$	$Nv_{14}/n_{68}$	$Nv_{15}/n_{69}$
Road section of vehicle	$(n_{28}, n_{29})$	$(n_{12}, n_{23})$	$(n_{27}, n_{28})$	$(n_2, n_{10})$	$(n_{18}, n_{32})$
Location of vehicle	$L_{28,65} = L_{65,29}$	$L_{12,66} = L_{66,23}$	$L_{27,67} = L_{67,28}$	$L_{2,68} = L_{68,10}$	$L_{18,69} = L_{69,32}$

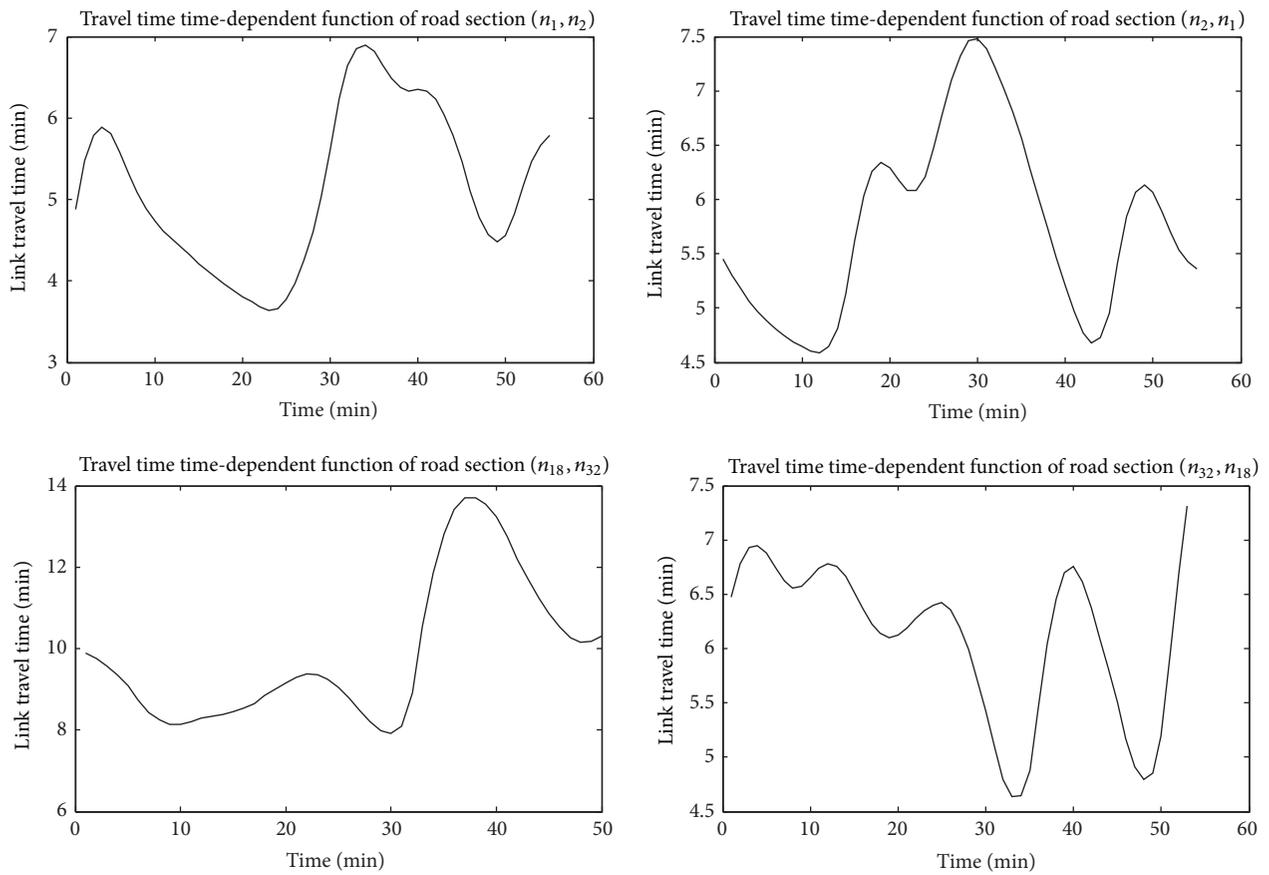


FIGURE 6: Link travel time time-dependent functions.

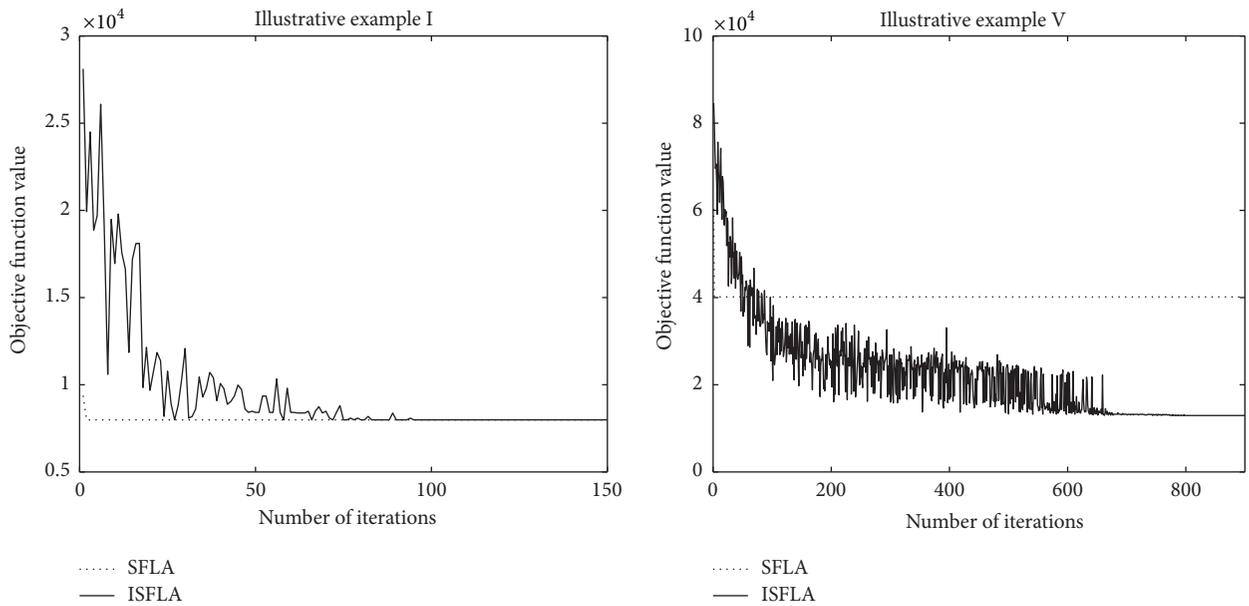


FIGURE 7: Evolutionary processes of the two algorithms for solving the illustrative examples.

TABLE 5: Emergency vehicles and accidents in each example.

Example no.	Vehicle no.	Accident no.
I	1, 2, 3, 4, 5, 6, 7, 8	1, 2, 3
II	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	1, 2, 3, 4
III	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	1, 2, 3, 4, 5
IV	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14	1, 2, 3, 4, 5, 6
V	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	1, 2, 3, 4, 5, 6, 7

Its objective function value of optimal selection strategies is 34.91% larger than that of ISFLA.

In example IV, there are  $7^{14}$  possible emergency vehicle selection strategies. Analyzing Table 9 shows that the objective function of optimal selection strategy obtained by using ISFLA is 12061. The SFLA is unwanted early convergence. Its objective function value of optimal selection strategies is 50.04% larger than that of ISFLA.

In example V, there are  $8^{15}$  possible emergency vehicle selection strategies. Analyzing Table 9 shows that the objective function of optimal selection strategy obtained by using ISFLA is 12947. The SFLA is unwanted early convergence. Its objective function value of optimal selection strategies is 210.10% larger than that of ISFLA.

Therefore, compared with SFLA, the optimization performance of ISFLA is getting better and better with the increase of the number of decision variables. ISFLA preceded SFLA in realizing the optimal resolution for the complicated emergency vehicle selection problem.

**5.2. Illustrative Examples of Dynamic Emergency Vehicle Dispatching Based on the Prediction Link Travel Speed.** Examples I and II in Section 5.1 were used to test the performance of dynamic emergency vehicle dispatching based on the prediction link travel speed. Polygonal-shaped travel speed functions  $v_{ij}(t)$ ,  $t \in [9:30, 10:30]$ ,  $\forall (n_i, n_j) \in E$ , were built using prediction travel speeds with different absolute error intervals ( $\pm 3$  km/h,  $\pm 5$  km/h, and  $\pm 10$  km/h). The present prediction methods can control the maximum absolute error of the prediction travel speed within  $\pm 10$  km/h [31, 32], so the maximum prediction error was assumed as  $\pm 10$  km/h. The prediction travel speed  $v_{ij}^2(\phi)_{t_0}$  was randomly generated within the interval  $[v_{ij}^1(t_\phi) - 10, v_{ij}^1(t_\phi) + 10]$ , in which,  $v_{ij}^1(t_\phi)$  is the real vehicle speed at the instant  $t_\phi$ . In order to analyze the effect of the decrease of prediction error on the initial emergency vehicle dispatching strategy, the two prediction errors  $\pm 3$  km/h and  $\pm 5$  km/h were assumed. Then the travel time functions  $T_{ij}(t)$ ,  $t \in [9:30, 10:30]$ , of each road section were calculated according to formula (5). The dynamic shortest path algorithm of emergency vehicles in expressway network proposed in our previous research [18] was used to calculate the shortest travel time and path, and the results are shown in Tables 10 and 11.

ISFLA was used to find the optimal dispatching strategy of the two examples under three prediction errors, and the best solutions are shown in Table 12.

In example I, the optimal emergency vehicle dispatching strategy (including vehicle routing and selection), which is

based on the prediction link travel speed with prediction errors of  $\pm 3$ ,  $\pm 5$ , is the same as the actual optimal strategy. However, when the prediction error is  $\pm 10$ , the rescue vehicle for the accident  $Ac_2$  is different from the actual optimal strategy. The emergency vehicle  $Ev_7$  instead of the vehicle  $Ev_8$  is dispatched to the accident  $Ac_2$ . This led to a  $23.4594 - 21.7823 = 1.6771$  minutes extension of the rescue time for the accident  $Ac_2$ . The objective function value based on the prediction travel speed is 101 more than that based on the real travel speed. It can be seen that the prediction error of the travel speed affects the accuracy of the initial emergency vehicle dispatching, but when the absolute prediction error is within  $\pm 10$ , the effect is not obvious.

In example II, the optimal emergency vehicle dispatching strategy (including vehicle routing and selection), which is based on the prediction link travel speed with prediction errors of  $\pm 3$ ,  $\pm 5$ , and  $\pm 10$ , is the same as the actual optimal strategy.

It is known from the two examples that the prediction error of  $\pm 10$  can basically meet the requirements of the initial dispatching.

**5.3. Illustrative Example of Dynamic Adjustment of Dispatching Strategy.** Example I in Section 5.2 was used to test the performance of dynamic adjustment of dispatching strategy. Suppose that the prediction error of link travel speed was  $\pm 5$ . At 9:30, emergency vehicles were dispatched to accidents according to the optimal dispatching strategy obtained in Section 5.2. Emergency vehicle  $Ev_2$  was dispatched to accident  $Ac_1$ . Emergency vehicle  $Ev_3$  was dispatched to accident  $Ac_3$ . Emergency vehicle  $Ev_4$  was dispatched to accident  $Ac_2$ . Emergency vehicle  $Ev_6$  was dispatched to accident  $Ac_3$ . Emergency vehicle  $Ev_8$  was dispatched to accident  $Ac_2$ . Emergency vehicles  $Ev_1$ ,  $Ev_5$ , and  $Ev_7$  were idle vehicles. The optimal dispatching strategy is shown in Figure 8.

At 9:35, a new accident  $Ac_4$  happens on the road section (9, 20). According to the real link travel speed, location of each emergency vehicle was calculated. They are shown in Figure 9.

Due to the influence of the accident  $Ac_4$ , road conditions of the accident section (9, 20) and its upstream sections (8, 9) and (1, 9) changed. In order to consider the impact of accident  $Ac_4$  on upstream sections, a weakening factor  $\alpha_{ij}(t)$  to the travel speed of upstream section  $(n_i, n_j)$  was introduced. Therefore, the travel speed functions of sections (8, 9) and (1, 9) were  $v'_{8,9}(t) = \alpha_{8,9}(t) \cdot v_{8,9}(t)$ ,  $v'_{1,9}(t) = \alpha_{1,9}(t) \cdot v_{1,9}(t)$ , respectively. Suppose that  $\alpha_{ij}$  decreases linearly with time  $t$ . The initial value  $\alpha_{ij}(t = t_\phi) = 0.7$ . When  $t = t_\phi + 30$  min,  $\alpha_{ij}(t) = 0.53$ .

The travel speed of each road section was recollected at 9:35, and the prediction travel speed function was calculated. We suppose that travel speed prediction error of road sections (8, 9) and (1, 9) was  $\pm 10$  and that of other road sections was  $\pm 5$ . Then, travel paths of emergency vehicles were replanned, and the dispatching strategy was adjusted. The optimal dispatching strategy at 9:35 is shown in Figure 10. The dynamic emergency vehicle dispatching process is shown in Table 13.

TABLE 6: The shortest travel time of the illustrative examples based on the real link travel speed.

Vehicle no.	Example no.	Accident no.						
		1	2	3	4	5	6	7
1	I	31.2925	38.9665	54.9495	--	--	--	--
	II	31.2925	38.9665	54.9495	54.2331	--	--	--
	III	31.2925	38.9665	54.9495	54.2331	34.8666	--	--
	IV	31.2925	38.9665	54.9495	54.2331	34.8666	45.2505	--
	V	31.2925	38.9665	54.9495	54.2331	34.8666	45.2505	$\infty$
2	I	22.1482	31.9124	$\infty$	--	--	--	--
	II	22.1482	31.9124	$\infty$	28.8721	--	--	--
	III	26.9763	31.9124	$\infty$	28.8721	15.2541	--	--
	IV	26.9763	31.9124	$\infty$	28.8721	15.2541	$\infty$	--
	V	26.9763	31.9124	$\infty$	28.8721	15.2541	$\infty$	38.6391
3	I	48.3611	47.2027	33.8637	--	--	--	--
	II	48.3611	47.2027	33.8637	59.0018	--	--	--
	III	48.3611	47.2027	33.8637	59.0018	54.0921	--	--
	IV	48.3611	47.2027	33.8637	59.0018	54.0921	28.6649	--
	V	48.3611	47.2027	33.8637	59.0018	54.0921	28.6649	$\infty$
4	I	41.0760	16.3454	53.4742	--	--	--	--
	II	41.0760	16.3454	$\infty$	10.1881	--	--	--
	III	41.0760	16.3454	$\infty$	10.1881	35.6330	--	--
	IV	41.0760	16.3454	$\infty$	10.1881	35.6330	$\infty$	--
	V	41.0760	16.3454	$\infty$	10.1881	35.6330	$\infty$	19.4956
5	I	49.9989	25.1126	38.7917	--	--	--	--
	II	49.9989	25.1126	38.7917	19.8843	--	--	--
	III	49.9989	25.1126	38.7917	19.8843	43.5353	--	--
	IV	49.9989	25.1126	38.7917	19.8843	43.5353	45.7355	--
	V	49.9989	25.1126	38.7917	19.8843	43.5353	45.7355	23.4806
6	I	44.4116	$\infty$	26.3079	--	--	--	--
	II	44.4116	$\infty$	26.3079	$\infty$	--	--	--
	III	44.4116	$\infty$	26.3079	$\infty$	54.4887	--	--
	IV	44.4116	$\infty$	26.3079	$\infty$	54.4887	21.4419	--
	V	44.4116	$\infty$	26.3079	$\infty$	54.4887	21.4419	$\infty$
7	I	43.6867	23.4594	56.5536	--	--	--	--
	II	43.6867	23.4594	56.5536	14.4705	--	--	--
	III	43.6867	23.4594	56.5536	14.4705	41.8074	--	--
	IV	43.6867	23.4594	56.5536	14.4705	41.8074	$\infty$	--
	V	43.6867	23.4594	56.5536	14.4705	41.8074	$\infty$	12.3566
8	I	42.6146	21.7823	$\infty$	--	--	--	--
	II	42.6146	21.7823	$\infty$	21.3312	--	--	--
	III	42.6146	21.7823	$\infty$	21.3312	40.7481	--	--
	IV	42.6146	21.7823	$\infty$	21.3312	40.7481	$\infty$	--
	V	42.6146	21.7823	$\infty$	21.3312	40.7481	$\infty$	20.0396
9	II	37.8754	33.6324	$\infty$	30.5210	--	--	--
	III	37.8754	33.6324	$\infty$	30.5210	42.3188	--	--
	IV	37.8754	33.6324	$\infty$	30.5210	42.3188	$\infty$	--
	V	37.8754	33.6324	$\infty$	30.5210	42.3188	$\infty$	29.6284
	10	II	42.3397	40.7709	28.2020	52.5731	--	--
III		42.3397	40.7709	28.2020	52.5731	47.7124	--	--
IV		42.3397	40.7709	28.2020	52.5731	47.7124	24.0304	--
V		42.3397	40.7709	28.2020	52.5731	47.7124	24.0304	$\infty$
11		III	$\infty$	27.4119	40.8674	21.9920	45.8012	--
	IV	$\infty$	27.4119	40.8674	21.9920	45.8012	48.8987	--
	V	$\infty$	27.4119	40.8674	21.9920	45.8012	48.8987	25.3691

TABLE 6: Continued.

Vehicle no.	Example no.	Accident no.						
		1	2	3	4	5	6	7
12	III	25.3758	49.7617	48.7390	$\infty$	33.5190	--	--
	IV	25.3758	49.7617	48.7390	$\infty$	33.5190	32.6388	--
	V	25.3758	49.7617	48.7390	$\infty$	33.5190	32.6388	$\infty$
13	IV	$\infty$	34.0944	21.9164	31.4456	54.1145	37.1084	--
	V	$\infty$	34.0944	21.9164	31.4456	54.1145	37.1084	33.8540
14	IV	10.1994	32.9815	$\infty$	48.1107	13.8316	54.7018	--
	V	10.1994	32.9815	$\infty$	48.1107	13.8316	54.7018	$\infty$
15	V	43.5208	22.7684	$\infty$	22.3348	41.7501	$\infty$	20.7021

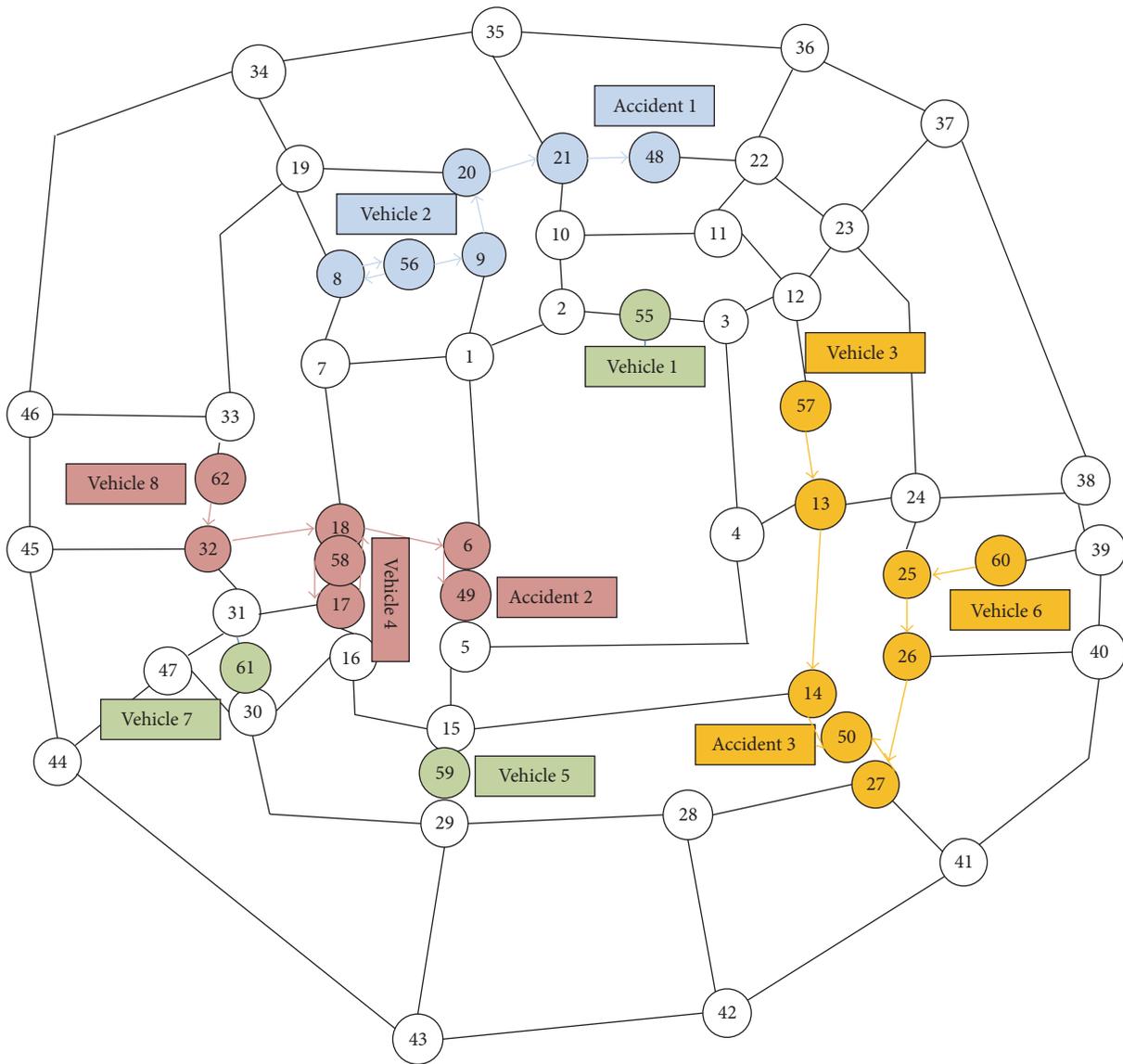


FIGURE 8: Optimal emergency vehicle dispatching strategy at 9:30.

TABLE 7: The shortest travel time path of the illustrative examples based on the real link travel speed.

Vehicle no.	Example no.	Accident no.						
		1	2	3	4	5	6	7
1	I	3, 2, 10, 21	3, 2, 1, 6	3, 12, 13, 14, 27	--			
	II	3, 2, 10, 21	3, 2, 1, 6	3, 12, 13, 14, 27	3, 2, 1, 6, 18, 17, 16			
	III	3, 2, 10, 21	3, 2, 1, 6	3, 12, 13, 14, 27	3, 2, 1, 6, 18, 17, 16	3, 2, 1, 9	--	--
	IV	3, 2, 10, 21	3, 2, 1, 6	3, 12, 13, 14, 27	3, 2, 1, 6, 18, 17, 16	3, 2, 1, 9	3, 12, 23, 37, 38	
	V	3, 2, 10, 21	3, 2, 1, 6	3, 12, 13, 14, 27	3, 2, 1, 6, 18, 17, 16	3, 2, 1, 9	3, 12, 23, 37, 38	BL
2	I	8, 9, 20, 21	8, 7, 18, 6	BL	--			
	II	8, 9, 20, 21	8, 7, 18, 6	BL	8, 7, 18, 17, 16			
	III	8, 9, 10, 21	8, 7, 18, 6	BL	8, 7, 18, 17, 16	8, 9	--	--
	IV	8, 9, 10, 21	8, 7, 18, 6	BL	8, 7, 18, 17, 16	8, 9	BL	
	V	8, 9, 10, 21	8, 7, 18, 6	BL	8, 7, 18, 17, 16	8, 9	BL	8, 7, 18, 17, 16, 30, 47
3	I	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	--			
	II	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	13, 4, 5, 15, 16			
	III	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	13, 4, 5, 15, 16	13, 12, 11, 10, 9	--	--
	IV	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	13, 4, 5, 15, 16	13, 12, 11, 10, 9	13, 24, 38	--
	V	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	13, 4, 5, 15, 16	13, 12, 11, 10, 9	13, 24, 38	BL
4	I	17, 18, 6, 1, 2, 10, 21	17, 18, 6	17, 16, 15, 14, 27	--			
	II	17, 18, 6, 1, 2, 10, 21	17, 18, 6	BL	17, 16			
	III	17, 18, 6, 1, 2, 10, 21	17, 18, 6	BL	17, 16	17, 18, 7, 8, 9	--	--
	IV	17, 18, 6, 1, 2, 10, 21	17, 18, 6	BL	17, 16	17, 18, 7, 8, 9	BL	--
	V	17, 18, 6, 1, 2, 10, 21	17, 18, 6	BL	17, 16	17, 18, 7, 8, 9	BL	17, 31, 47
5	I	29, 15, 5, 6, 1, 2, 10, 21	29, 15, 5, 6	29, 28, 27	--			
	II	29, 15, 5, 6, 1, 2, 10, 21	29, 15, 5, 6	29, 28, 27	29, 15, 16			
	III	29, 15, 5, 6, 1, 2, 10, 21	29, 15, 5, 6	29, 28, 27	29, 15, 16	29, 15, 5, 6, 1, 9	--	--
	IV	29, 15, 5, 6, 1, 2, 10, 21	29, 15, 5, 6	29, 28, 27	29, 15, 16	29, 15, 5, 6, 1, 9	29, 43, 42, 41, 40, 39, 38	--
	V	29, 15, 5, 6, 1, 2, 10, 21	29, 15, 5, 6	29, 28, 27	29, 15, 16	29, 15, 5, 6, 1, 9	29, 43, 42, 41, 40, 39, 38	29, 30, 47
6	I	25, 24, 23, 22, 21	BL	25, 26, 27	--			
	II	25, 24, 23, 22, 21	BL	25, 26, 27	BL			
	III	25, 24, 23, 22, 21	BL	25, 26, 27	BL	25, 24, 23, 12, 11, 10, 9	--	--
	IV	25, 24, 23, 22, 21	BL	25, 26, 27	BL	25, 24, 23, 12, 11, 10, 9	25, 24, 38	--
	V	25, 24, 23, 22, 21	BL	25, 26, 27	BL	25, 24, 23, 12, 11, 10, 9	25, 24, 38	BL

TABLE 7: Continued.

Vehicle no.	Example no.	Accident no.						
		1	2	3	4	5	6	7
7	I	31, 32, 33, 19, 20, 21	31, 17, 18, 6	31, 30, 29, 28, 27	--			
	II	31, 32, 33, 19, 20, 21	31, 17, 18, 6	31, 30, 29, 28, 27	31, 30, 16			
	III	31, 32, 33, 19, 20, 21	31, 17, 18, 6	31, 30, 29, 28, 27	31, 30, 16	31, 17, 18, 6, 1, 9	--	--
	IV	31, 32, 33, 19, 20, 21	31, 17, 18, 6	31, 30, 29, 28, 27	31, 30, 16	31, 17, 18, 6, 1, 9	BL	--
	V	31, 32, 33, 19, 20, 21	31, 17, 18, 6	31, 30, 29, 28, 27	31, 30, 16	31, 17, 18, 6, 1, 9	BL	31, 47
8	I	32, 33, 19, 20, 21	32, 18, 6	BL	--			
	II	32, 33, 19, 20, 21	32, 18, 6	BL	32, 18, 17, 16			
	III	32, 33, 19, 20, 21	32, 18, 6	BL	32, 18, 17, 16	32, 18, 6, 1, 9	--	--
	IV	32, 33, 19, 20, 21	32, 18, 6	BL	32, 18, 17, 16	32, 18, 6, 1, 9	BL	--
	V	32, 33, 19, 20, 21	32, 18, 6	BL	32, 18, 17, 16	32, 18, 6, 1, 9	BL	32, 31, 47
9	II	33, 19, 20, 21	33, 32, 18, 6	BL	33, 32, 18, 17, 16			
	III	33, 19, 20, 21	33, 32, 18, 6	BL	33, 32, 18, 17, 16	33, 19, 8, 9	--	--
	IV	33, 19, 20, 21	33, 32, 18, 6	BL	33, 32, 18, 17, 16	33, 19, 8, 9	BL	--
	V	33, 19, 20, 21	33, 32, 18, 6	BL	33, 32, 18, 17, 16	33, 19, 8, 9	BL	33, 32, 31, 47
10	II	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	13, 4, 5, 15, 16			
	III	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	13, 4, 5, 15, 16	13, 12, 11, 10, 9	--	--
	IV	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	13, 4, 5, 15, 16	13, 12, 11, 10, 9	13, 24, 38	--
	V	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	13, 4, 5, 15, 16	13, 12, 11, 10, 9	13, 24, 38	BL
11	III	BL	29, 15, 5, 6	29, 28, 27	29, 15, 16	29, 15, 5, 6, 1, 9	--	--
	IV	BL	29, 15, 5, 6	29, 28, 27	29, 15, 16	29, 15, 5, 6, 1, 9	29, 43, 42, 41, 40, 39, 38	--
	V	BL	29, 15, 5, 6	29, 28, 27	29, 15, 16	29, 15, 5, 6, 1, 9	29, 43, 42, 41, 40, 39, 38	29, 30, 47
12	III	23, 22, 21	23, 12, 3, 2, 1, 6	23, 24, 25, 26, 27	BL	23, 12, 11, 10, 9	--	--
	IV	23, 22, 21	23, 12, 3, 2, 1, 6	23, 24, 25, 26, 27	BL	23, 12, 11, 10, 9	23, 37, 38	--
	V	23, 22, 21	23, 12, 3, 2, 1, 6	23, 24, 25, 26, 27	BL	23, 12, 11, 10, 9	23, 37, 38	BL
13	IV	BL	28, 29, 15, 5, 6	28, 27	28, 29, 15, 16	28, 29, 15, 5, 6, 1, 9	28, 42, 41, 40, 39, 38	--
	V	BL	28, 29, 15, 5, 6	28, 27	28, 29, 15, 16	28, 29, 15, 5, 6, 1, 9	28, 42, 41, 40, 39, 38	28, 29, 30, 47
14	IV	10, 21	10, 2, 1, 6	BL	10, 9, 8, 7, 18, 17, 16	10, 9	10, 21, 35, 36, 37, 38	--
	V	10, 21	10, 2, 1, 6	BL	10, 9, 8, 7, 18, 17, 16	10, 9	10, 21, 35, 36, 37, 38	BL

TABLE 7: Continued.

Vehicle no.	Example no.	Accident no.						
		1	2	3	4	5	6	7
15	V	32, 33, 19, 20, 21	32, 18, 6	BL	32, 18, 17, 16	32, 18, 6, 1, 9	BL	32, 31, 47

Note. BL means that the travel time of the optimal path exceeds the upper limit of the time window.

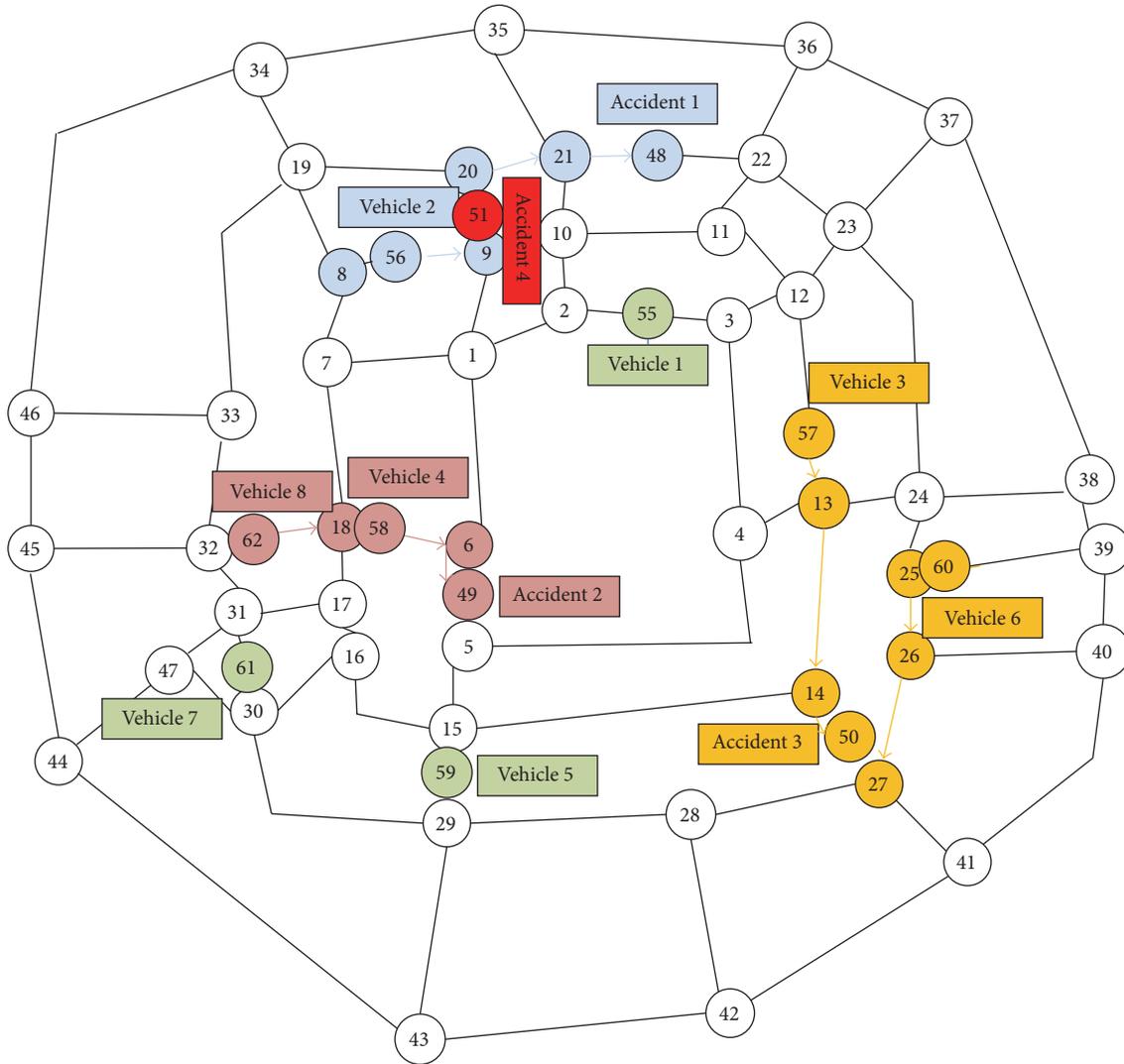


FIGURE 9: Locations of emergency vehicles and accidents at 9:35.

When the accident  $Ac_4$  happened at 9:35, the dynamical adjustment mechanism timely responded to sudden changes in road conditions of the optimal path (8, 9, 20, 21) of emergency vehicle  $Ev_2$ . After recalculating the dispatching strategy, the idle emergency vehicle  $Ev_1$  replaced the emergency vehicle  $Ev_2$  to rescue the accident  $Ac_1$ . The emergency vehicle successfully bypassed road sections (9, 20) and (8, 9) to ensure the time limit for emergency rescue. Meanwhile, the emergency vehicle  $Ev_2$  with the shortest travel time was dispatched to the new accident  $Ac_4$ . Therefore, the adjustment

of emergency vehicle dispatching strategy can effectively shorten the response time of accident.

### 6. Conclusions

The emergency vehicle dispatching in urban expressway network includes dynamic routing and emergency vehicles election. And the dispatching strategy needs to be adjusted according to the dynamic road conditions. Firstly, polygonal-shaped travel speed function based on real-time and

TABLE 8: Parameter selection of ISFLA.

Example no.	IP	$a$	$c$	IT	$D_d^{\max}$ $d = 1, \dots, D$	$Z_d^{\max}$ $d = 1, \dots, D$	$Z_d^{\min}$ $d = 1, \dots, D$	$D$	$\theta$
I	150	15	10	150	3	4	1	8	0.05
II	150	15	10	350	4	5	1	10	0.05
III	300	20	15	600	5	6	1	12	0.05
IV	374	22	17	800	5	7	1	14	0.05
V	500	25	20	900	6	8	1	15	0.05

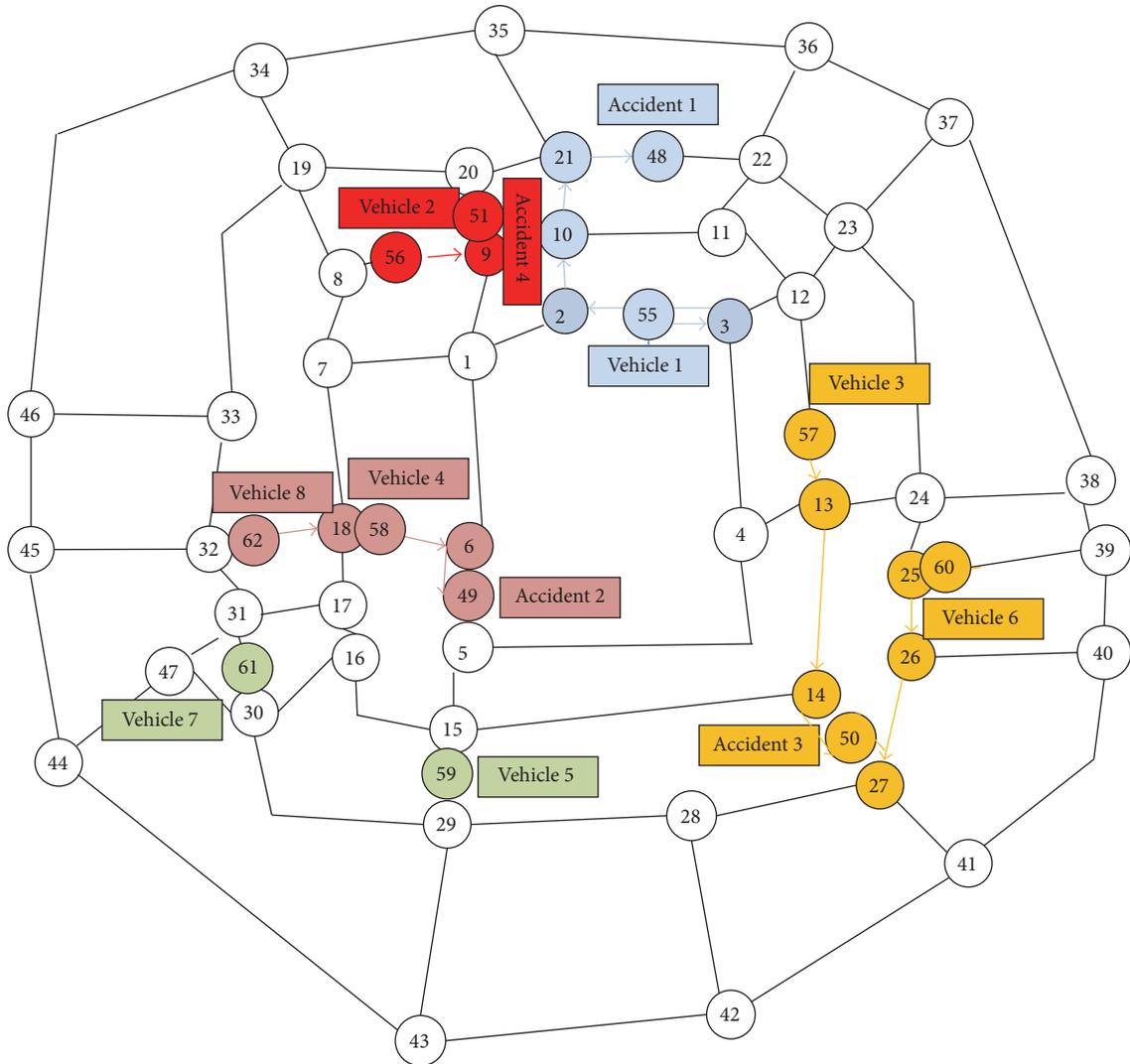


FIGURE 10: Optimal emergency vehicle dispatching strategy at 9:35.

prediction link travel speed is set up, and the dynamic emergency vehicle dispatching process is illustrated. Secondly, taking the accident severity as the key factor and the travel time as the objective function, a dynamic dispatching model considering the vehicle routing was established. The model consists of two stages: initial dispatching and dynamic adjustment. Thirdly, in order to avoid unwanted early convergence of SFLA in solving complex dispatching problems, the basic

SFLA was combined with the probabilistic model of EDA, and the SFLA was improved. Finally, Beijing expressway network was taken as an example to test the efficacy of the model and the algorithm from three aspects. First, based on the real link travel speed, 5 emergency vehicle dispatching problems with different scale variable were modeled and solved by the improved shuffled frog leaping algorithm. Second, based on the prediction link travel speed and the

TABLE 9: Optimal emergency vehicle dispatching strategy of the illustrative examples based on the real link travel speed.

Example no.	Example parameters	Algorithm	The optimal solution	Optimal emergency vehicle dispatching strategy	Travel time (min)	
I	Total number of vehicles	8		$Ev_2$ to $Ac_1$ ; $Nv_2 \rightarrow n_8 \rightarrow n_9 \rightarrow n_{20} \rightarrow n_{21} \rightarrow Nc_1$	22.1482	
	Total number of accidents	3	0,1,3,2,0,3,0,2	$Ev_4$ to $Ac_2$ ; $Nv_4 \rightarrow n_{17} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	16.3454	
	Accident severity	$As_1 = 40$	ISFLA	0,1,3,2,0,3,0,2	$Ev_8$ to $Ac_2$ ; $Nv_8 \rightarrow n_{32} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	21.7823
		$As_2 = 60$			$Ev_3$ to $Ac_3$ ; $Nv_3 \rightarrow n_{13} \rightarrow n_{14} \rightarrow n_{27} \rightarrow Nc_3$	33.8637
		$As_3 = 80$			$Ev_6$ to $Ac_3$ ; $Nv_6 \rightarrow n_{25} \rightarrow n_{26} \rightarrow n_{27} \rightarrow Nc_3$	26.3079
	Accident demand	$Na_1 = 1$	SFLA	0,1,3,2,0,3,0,2	$Ev_2$ to $Ac_1$ ; $Nv_2 \rightarrow n_8 \rightarrow n_9 \rightarrow n_{20} \rightarrow n_{21} \rightarrow Nc_1$	22.1482
		$Na_2 = 2$			$Ev_4$ to $Ac_2$ ; $Nv_4 \rightarrow n_{17} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	16.3454
		$Na_3 = 2$			$Ev_8$ to $Ac_2$ ; $Nv_8 \rightarrow n_{32} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	21.7823
	Total number of vehicles	10		$Ev_3$ to $Ac_3$ ; $Nv_3 \rightarrow n_{13} \rightarrow n_{14} \rightarrow n_{27} \rightarrow Nc_3$	33.8637	
	Total number of accidents	4	0,1,3,2,0,3,0,2	$Ev_6$ to $Ac_3$ ; $Nv_6 \rightarrow n_{25} \rightarrow n_{26} \rightarrow n_{27} \rightarrow Nc_3$	26.3079	
II	Total number of vehicles	10		$Ev_2$ to $Ac_1$ ; $Nv_2 \rightarrow n_8 \rightarrow n_9 \rightarrow n_{20} \rightarrow n_{21} \rightarrow Nc_1$	22.1482	
	Total number of accidents	4	0,1,0,2,0,3,4,2,0,3	$Ev_4$ to $Ac_2$ ; $Nv_4 \rightarrow n_{17} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	16.3454	
	Accident severity	$As_1 = 40$	ISFLA	0,1,0,2,0,3,4,2,0,3	$Ev_8$ to $Ac_2$ ; $Nv_8 \rightarrow n_{32} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	21.7823
		$As_2 = 60$			$Ev_6$ to $Ac_3$ ; $Nv_6 \rightarrow n_{25} \rightarrow n_{26} \rightarrow n_{27} \rightarrow Nc_3$	26.3079
		$As_3 = 80$			$Ev_{10}$ to $Ac_3$ ; $Nv_{10} \rightarrow n_{13} \rightarrow n_{14} \rightarrow n_{27} \rightarrow Nc_3$	28.2020
	Accident demand	$As_4 = 40$	SFLA	0,1,0,4,2,3,0,0,2,3	$Ev_7$ to $Ac_4$ ; $Nv_7 \rightarrow n_{31} \rightarrow n_{30} \rightarrow n_{16} \rightarrow Nc_4$	14.4705
		$Na_1 = 1$			$Ev_2$ to $Ac_1$ ; $Nv_2 \rightarrow n_8 \rightarrow n_9 \rightarrow n_{20} \rightarrow n_{21} \rightarrow Nc_1$	22.1482
		$Na_2 = 2$			$Ev_5$ to $Ac_2$ ; $Nv_5 \rightarrow n_{29} \rightarrow n_{15} \rightarrow n_5 \rightarrow n_6 \rightarrow Nc_2$	25.1126
	Accident demand	$Na_3 = 2$	SFLA	0,1,0,4,2,3,0,0,2,3	$Ev_9$ to $Ac_3$ ; $Nv_9 \rightarrow n_{33} \rightarrow n_{32} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	33.6324
		$Na_4 = 1$			$Ev_6$ to $Ac_3$ ; $Nv_6 \rightarrow n_{25} \rightarrow n_{26} \rightarrow n_{27} \rightarrow Nc_3$	26.3079
		$Ev_{10}$ to $Ac_3$ ; $Nv_{10} \rightarrow n_{13} \rightarrow n_{14} \rightarrow n_{27} \rightarrow Nc_3$			28.2020	
				$Ev_4$ to $Ac_4$ ; $Nv_4 \rightarrow n_{17} \rightarrow n_{16} \rightarrow Nc_4$	10.1881	

TABLE 9: Continued.

Example no.	Example parameters	Algorithm	The optimal solution	Optimal emergency vehicle dispatching strategy	Travel time (min)
III	Total number of vehicles			$Ev_1$ to $Ac_5$ ; $Nv_1 \rightarrow n_3 \rightarrow n_2 \rightarrow n_1 \rightarrow n_9 \rightarrow Nc_5$	34.8666
				$Ev_2$ to $Ac_5$ ; $Nv_2 \rightarrow n_8 \rightarrow n_9 \rightarrow Nc_5$	15.2541
				$Ev_4$ to $Ac_5$ ; $Nv_4 \rightarrow n_{17} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	16.3454
				$Ev_6$ to $Ac_5$ ; $Nv_6 \rightarrow n_{25} \rightarrow n_{26} \rightarrow n_{27} \rightarrow Nc_3$	26.3079
	Total number of accidents	ISFLA	5,5,0,2,0,3,4,2,0,3,0,1	$Ev_7$ to $Ac_4$ ; $Nv_7 \rightarrow n_{31} \rightarrow n_{30} \rightarrow n_{16} \rightarrow Nc_4$	14.4705
				$Ev_8$ to $Ac_5$ ; $Nv_8 \rightarrow n_{32} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	21.7823
				$Ev_{10}$ to $Ac_5$ ; $Nv_{10} \rightarrow n_{13} \rightarrow n_{14} \rightarrow n_{27} \rightarrow Nc_3$	28.2020
	Accident severity			$Ev_{12}$ to $Ac_1$ ; $Nv_{12} \rightarrow n_{23} \rightarrow n_{22} \rightarrow n_{21} \rightarrow Nc_1$	25.3758
				$Ev_2$ to $Ac_2$ ; $Nv_2 \rightarrow n_8 \rightarrow n_7 \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	31.9124
				$Ev_3$ to $Ac_3$ ; $Nv_3 \rightarrow n_{13} \rightarrow n_{14} \rightarrow n_{27} \rightarrow Nc_3$	33.8637
				$Ev_4$ to $Ac_5$ ; $Nv_4 \rightarrow n_{17} \rightarrow n_{18} \rightarrow n_7 \rightarrow n_8 \rightarrow n_9 \rightarrow Nc_5$	35.6330
				$Ev_6$ to $Ac_3$ ; $Nv_6 \rightarrow n_{25} \rightarrow n_{26} \rightarrow n_{27} \rightarrow Nc_3$	26.3079
				$Ev_7$ to $Ac_4$ ; $Nv_7 \rightarrow n_{31} \rightarrow n_{30} \rightarrow n_{16} \rightarrow Nc_4$	14.4705
	Accident demand	SFLA	0,2,3,5,0,3,4,5,0,2,0,1	$Ev_8$ to $Ac_5$ ; $Nv_8 \rightarrow n_{32} \rightarrow n_{18} \rightarrow n_6 \rightarrow n_1 \rightarrow n_9 \rightarrow Nc_5$	40.7481
				$Ev_{10}$ to $Ac_5$ ; $Nv_{10} \rightarrow n_{13} \rightarrow n_4 \rightarrow n_5 \rightarrow n_6 \rightarrow Nc_2$	40.7709
				$Ev_{12}$ to $Ac_1$ ; $Nv_{12} \rightarrow n_{23} \rightarrow n_{22} \rightarrow n_{21} \rightarrow Nc_1$	25.3758
				$Ev_2$ to $Ac_5$ ; $Nv_2 \rightarrow n_8 \rightarrow n_9 \rightarrow Nc_5$	15.2541
				$Ev_3$ to $Ac_6$ ; $Nv_3 \rightarrow n_{13} \rightarrow n_{24} \rightarrow n_{38} \rightarrow Nc_6$	28.6649
	Total number of vehicles			$Ev_4$ to $Ac_5$ ; $Nv_4 \rightarrow n_{17} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	16.3454
				$Ev_6$ to $Ac_6$ ; $Nv_6 \rightarrow n_{25} \rightarrow n_{24} \rightarrow n_{38} \rightarrow Nc_6$	21.4419
IV				$Ev_7$ to $Ac_4$ ; $Nv_7 \rightarrow n_{31} \rightarrow n_{30} \rightarrow n_{16} \rightarrow Nc_4$	14.4705
				$Ev_8$ to $Ac_2$ ; $Nv_8 \rightarrow n_{32} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	21.7823
	Total number of accidents	ISFLA	0,5,6,2,0,6,4,2,0,3,0,1,3,5	$Ev_{10}$ to $Ac_5$ ; $Nv_{10} \rightarrow n_{13} \rightarrow n_{14} \rightarrow n_{27} \rightarrow Nc_3$	28.2020
				$Ev_{12}$ to $Ac_1$ ; $Nv_{12} \rightarrow n_{23} \rightarrow n_{22} \rightarrow n_{21} \rightarrow Nc_1$	25.3758
				$Ev_{13}$ to $Ac_3$ ; $Nv_{13} \rightarrow n_{28} \rightarrow n_{27} \rightarrow Nc_3$	21.9164
				$Ev_{14}$ to $Ac_5$ ; $Nv_{14} \rightarrow n_{10} \rightarrow n_9 \rightarrow Nc_5$	13.8316
	Accident severity			$Ev_1$ to $Ac_6$ ; $Nv_1 \rightarrow n_3 \rightarrow n_{12} \rightarrow n_{23} \rightarrow n_{37} \rightarrow n_{38} \rightarrow Nc_6$	45.2505
				$Ev_4$ to $Ac_5$ ; $Nv_4 \rightarrow n_{17} \rightarrow n_{18} \rightarrow n_7 \rightarrow n_8 \rightarrow n_9 \rightarrow Nc_5$	35.6330
				$Ev_6$ to $Ac_6$ ; $Nv_6 \rightarrow n_{25} \rightarrow n_{24} \rightarrow n_{38} \rightarrow Nc_6$	21.4419
				$Ev_7$ to $Ac_5$ ; $Nv_7 \rightarrow n_{31} \rightarrow n_{17} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	23.4594
			$Ev_8$ to $Ac_2$ ; $Nv_8 \rightarrow n_{32} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	21.7823	
			$Ev_9$ to $Ac_1$ ; $Nv_9 \rightarrow n_{33} \rightarrow n_{19} \rightarrow n_{20} \rightarrow n_{21} \rightarrow Nc_1$	37.8754	
			$Ev_{10}$ to $Ac_5$ ; $Nv_{10} \rightarrow n_{13} \rightarrow n_{12} \rightarrow n_{11} \rightarrow n_{10} \rightarrow n_9 \rightarrow Nc_5$	47.7124	
Accident demand	SFLA	6,0,0,5,0,6,2,2,1,5,4,3,3,0	$Ev_{11}$ to $Ac_4$ ; $Nv_{11} \rightarrow n_{29} \rightarrow n_{15} \rightarrow n_{16} \rightarrow Nc_4$	21.9920	
			$Ev_{12}$ to $Ac_5$ ; $Nv_{12} \rightarrow n_{23} \rightarrow n_{24} \rightarrow n_{25} \rightarrow n_{26} \rightarrow n_{27} \rightarrow Nc_3$	48.7390	
			$Ev_{13}$ to $Ac_3$ ; $Nv_{13} \rightarrow n_{28} \rightarrow n_{27} \rightarrow Nc_3$	21.9164	

TABLE 9: Continued.

Example no.	Example parameters	Algorithm	The optimal solution	Optimal emergency vehicle dispatching strategy	Travel time (min)				
V	Total number of vehicles	15	ISFLA	0,5,3,2,4,6,7,2,0,6,0,1,3,5,0	$Ev_2$ to $Ac_5$ : $Nv_2 \rightarrow n_8 \rightarrow n_9 \rightarrow Nc_5$	15.2541			
					$Ev_3$ to $Ac_5$ : $Nv_3 \rightarrow n_{13} \rightarrow n_{14} \rightarrow n_{27} \rightarrow Nc_3$	33.8637			
	Total number of accidents	7	ISFLA	0,5,3,2,4,6,7,2,0,6,0,1,3,5,0	$Ev_4$ to $Ac_2$ : $Nv_4 \rightarrow n_{17} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	16.3454			
					$Ev_5$ to $Ac_4$ : $Nv_5 \rightarrow n_{29} \rightarrow n_{15} \rightarrow n_{16} \rightarrow Nc_4$	19.8843			
					$Ev_6$ to $Ac_6$ : $Nv_6 \rightarrow n_{25} \rightarrow n_{24} \rightarrow n_{38} \rightarrow Nc_6$	21.4419			
					$Ev_7$ to $Ac_7$ : $Nv_7 \rightarrow n_{31} \rightarrow n_{47} \rightarrow Nc_7$	12.3566			
					$Ev_8$ to $Ac_2$ : $Nv_8 \rightarrow n_{32} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	21.7823			
					$Ev_{10}$ to $Ac_6$ : $Nv_{10} \rightarrow n_{13} \rightarrow n_{24} \rightarrow n_{38} \rightarrow Nc_6$	24.0304			
					$Ev_{12}$ to $Ac_1$ : $Nv_{12} \rightarrow n_{23} \rightarrow n_{22} \rightarrow n_{21} \rightarrow Nc_1$	25.3758			
					$Ev_{13}$ to $Ac_5$ : $Nv_{13} \rightarrow n_{28} \rightarrow n_{27} \rightarrow Nc_5$	21.9164			
Accident severity	7	ISFLA	0,5,3,2,4,6,7,2,0,6,0,1,3,5,0	$Ev_{14}$ to $Ac_5$ : $Nv_{14} \rightarrow n_{10} \rightarrow n_9 \rightarrow Nc_5$	13.8316				
				$Ev_1$ to $Ac_1$ : $Nv_1 \rightarrow n_3 \rightarrow n_2 \rightarrow n_{10} \rightarrow n_{21} \rightarrow Nc_1$	31.2925				
				$Ev_2$ to $Ac_2$ : $Nv_2 \rightarrow n_8 \rightarrow n_7 \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	31.9124				
				$Ev_4$ to $Ac_7$ : $Nv_4 \rightarrow n_{17} \rightarrow n_{31} \rightarrow n_{47} \rightarrow Nc_7$	19.4956				
				$Ev_5$ to $Ac_4$ : $Nv_5 \rightarrow n_{29} \rightarrow n_{15} \rightarrow n_{16} \rightarrow Nc_4$	19.8843				
				$Ev_6$ to $Ac_6$ : $Nv_6 \rightarrow n_{25} \rightarrow n_{26} \rightarrow n_{27} \rightarrow Nc_6$	26.3079				
				$Ev_7$ to $Ac_7$ : $Nv_7 \rightarrow n_{31} \rightarrow n_{47} \rightarrow Nc_7$	12.3566				
				$Ev_8$ to $Ac_5$ : $Nv_8 \rightarrow n_{32} \rightarrow n_{18} \rightarrow n_6 \rightarrow n_1 \rightarrow n_9 \rightarrow Nc_5$	40.7481				
				$Ev_9$ to $Ac_2$ : $Nv_9 \rightarrow n_{33} \rightarrow n_{32} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	33.6324				
				$Ev_{10}$ to $Ac_6$ : $Nv_{10} \rightarrow n_{13} \rightarrow n_{24} \rightarrow n_{38} \rightarrow Nc_6$	24.0304				
Accident demand	7	ISFLA	0,5,3,2,4,6,7,2,0,6,0,1,3,5,0	$Ev_{11}$ to $Ac_5$ : $Nv_{11} \rightarrow n_{29} \rightarrow n_{28} \rightarrow n_{27} \rightarrow Nc_5$	21.9920				
				$Ev_{13}$ to $Ac_6$ : $Nv_{13} \rightarrow n_{28} \rightarrow n_{42} \rightarrow n_{41} \rightarrow n_{40} \rightarrow n_{39} \rightarrow n_{38} \rightarrow Nc_6$	37.1084				
				$Ev_{14}$ to $Ac_5$ : $Nv_{14} \rightarrow n_{10} \rightarrow n_9 \rightarrow Nc_5$	13.8316				
				$Ev_{15}$ to $Ac_5$ : $Nv_{15} \rightarrow n_{32} \rightarrow n_{18} \rightarrow n_6 \rightarrow n_1 \rightarrow n_9 \rightarrow Nc_5$	41.7501				
				The average evolution time (s)					Success rate (%) in ten runs
				I	4 <sup>8</sup>	ISFLA	7987	2,723	100
				II	5 <sup>10</sup>	SFLA	7987	2,809	30
						ISFLA	8113	15,643	100
				III	6 <sup>12</sup>	SFLA	9179	5,324	0
						ISFLA	10247	25,534	100
IV	7 <sup>14</sup>	SFLA	13824	25,163	0				
		ISFLA	12061	97,333	100				
V	8 <sup>15</sup>	SFLA	18097	51,567	0				
		ISFLA	12947	144,199	100				
		SFLA	40149	101,786	0				

TABLE 10: The shortest travel time of the illustrative examples based on the prediction link travel speed.

Vehicle no.	Example no.	Prediction error (km/h)	Accident no.			
			1	2	3	4
1	I	0	31.2925	38.9665	54.9495	--
		$\pm 3$	31.2737	38.5534	54.9460	--
		$\pm 5$	32.1368	38.4649	55.5341	--
		$\pm 10$	31.5763	42.7993	50.3388	--
	II	0	31.2925	38.9665	54.9495	54.2331
		$\pm 3$	31.2737	38.5534	54.9460	53.9916
		$\pm 5$	32.1368	38.4649	55.5341	53.4761
		$\pm 10$	31.5763	42.7993	50.3388	$\infty$
2	I	0	22.1482	31.9124	$\infty$	--
		$\pm 3$	20.8670	32.3468	$\infty$	--
		$\pm 5$	23.1840	31.3765	$\infty$	--
		$\pm 10$	18.3767	32.2828	$\infty$	--
	II	0	22.1482	31.9124	$\infty$	28.8721
		$\pm 3$	20.8670	32.3468	$\infty$	28.8694
		$\pm 5$	23.1840	31.3765	$\infty$	30.9670
		$\pm 10$	18.3767	32.2828	$\infty$	26.4442
3	I	0	48.3611	47.2027	33.8637	--
		$\pm 3$	46.8840	47.3166	33.7495	--
		$\pm 5$	46.8704	47.2581	33.8035	--
		$\pm 10$	46.4381	52.2131	33.2153	--
	II	0	48.3611	47.2027	33.8637	59.0018
		$\pm 3$	46.8840	47.3166	33.7495	59.8245
		$\pm 5$	46.8704	47.2581	33.8035	$\infty$
		$\pm 10$	46.4381	52.2131	33.2153	58.7999
4	I	0	41.0760	16.3454	53.4742	--
		$\pm 3$	41.2953	16.4075	54.8336	--
		$\pm 5$	41.6761	16.6789	52.0917	--
		$\pm 10$	40.3378	17.3145	53.4260	--
	II	0	41.0760	16.3454	$\infty$	10.1881
		$\pm 3$	41.2953	16.4075	$\infty$	10.5117
		$\pm 5$	41.6761	16.6789	$\infty$	9.5027
		$\pm 10$	40.3378	17.3145	$\infty$	9.6885
5	I	0	49.9989	25.1126	38.7917	--
		$\pm 3$	49.5269	25.4762	38.9094	--
		$\pm 5$	50.8201	24.7458	38.8310	--
		$\pm 10$	50.8567	28.2398	38.5163	--
	II	0	49.9989	25.1126	38.7917	19.8843
		$\pm 3$	49.5269	25.4762	38.9094	19.9819
		$\pm 5$	50.8201	24.7458	38.8310	18.5017
		$\pm 10$	50.8567	28.2398	38.5163	22.5535
6	I	0	44.4116	$\infty$	26.3079	--
		$\pm 3$	43.5208	51.2845	25.8029	--
		$\pm 5$	47.1051	51.4706	25.4729	--
		$\pm 10$	50.1697	54.9481	31.6198	--
	II	0	44.4116	$\infty$	26.3079	$\infty$
		$\pm 3$	43.5208	51.2845	25.8029	59.7672
		$\pm 5$	47.1051	51.4706	25.4729	$\infty$
		$\pm 10$	50.1697	54.9481	31.6198	$\infty$

TABLE 10: Continued.

Vehicle no.	Example no.	Prediction error (km/h)	Accident no.			
			1	2	3	4
7	I	0	43.6867	23.4594	56.5536	--
		±3	43.8486	23.8514	56.9816	--
		±5	42.8287	22.6855	53.4435	--
		±10	42.4865	24.9977	55.1983	--
	II	0	43.6867	23.4594	56.5536	14.4705
		±3	43.8486	23.8514	56.9816	14.7354
		±5	42.8287	22.6855	53.4435	13.9424
		±10	42.4865	24.9977	55.1983	13.5972
8	I	0	42.6146	21.7823	∞	--
		±3	42.7782	21.8997	∞	--
		±5	42.5819	21.8541	∞	--
		±10	40.9011	26.2546	∞	--
	II	0	42.6146	21.7823	∞	21.3312
		±3	42.7782	21.8997	∞	21.5001
		±5	42.5819	21.8541	∞	20.9878
		±10	40.9011	26.2546	∞	22.7096
9	II	0	37.8754	33.6324	∞	30.5210
		±3	37.3625	33.4413	∞	29.9943
		±5	39.1213	34.4820	∞	31.7022
		±10	37.5274	35.6110	∞	29.7247
10	II	0	42.3397	40.7709	28.2020	52.5731
		±3	40.9388	40.6356	28.5534	53.1496
		±5	42.4467	40.9227	28.6415	55.0761
		±10	41.3142	41.4190	26.8230	51.2051

real-time link travel speed, the dynamic emergency vehicle dispatching model and the improved SFLA were used to calculate the global optimal dispatching strategy. Third, a scene in which a new accident happened on the rescue path was simulated, and the dispatching strategy was adjusted. The results show the following: (1) The emergency vehicle dispatching model can obtain the optimal dispatching strategy under the constraints of accident demands and time windows. (2) For solving the complicated emergency vehicle selection problem, the improved SFLA has stronger search ability compared with the SFLA and can obtain more optimal dispatching strategy. (3) The optimal strategy obtained by the emergency vehicle dispatching model based on the prediction link travel speed takes into account the dynamic changes in the road conditions during the whole dispatching process. (4) The dynamic adjustment condition can timely respond to sudden changes in the road conditions and help emergency vehicles to avoid rescue delay.

**Symbols**

$N$ : The expressway network node set  
 $n_i, i = 1, 2, \dots, M$ : The  $i$ th expressway network node  
 $M$ : The total number of nodes in the expressway network

$E$ : The road section set of the expressway network  
 $(n_i, n_j) \in E$ : The road section from the node  $n_i$  to the node  $n_j$   
 $Q$ : The interested time horizon  
 $T_{ij}(t)$ : The link travel time function defined in time horizon  $Q$ . It represents a time for the emergency vehicle, leaving at an instant  $t$ , from node  $n_i$  to  $n_j$   
 $T(t)$ : The link travel time function set  
 $\kappa$ : The minimum time interval  
 $t_0$ : The initial dispatching decision-making instant  
 $t_\phi, \phi = 0, 1, \dots, \Phi$ : The start time of the  $\phi$ th time interval  
 $t_\Phi$ : The last instant of the interested time horizon  
 $v_{ij}(t)$ : The link travel speed function of the road section  $(n_i, n_j)$  at the instant  $t$   
 $v_{ij}^1(t_0)$ : The real-time link travel speed of the road section  $(n_i, n_j)$  at the instant  $t_0$   
 $v_{ij}^2(\phi)_{t_0}$ : The prediction travel speed at the instant  $t_\phi$  calculated according to  $v_{ij}^1(t_0)$

TABLE II: The shortest travel time path of the illustrative examples based on the prediction link travel speed.

Vehicle no.	Example no.	Prediction error (km/h)	Accident no.			
			1	2	3	4
1	I	0	3, 2, 10, 21	3, 2, 1, 6	3, 12, 13, 14, 27	--
		±3	3, 2, 10, 21	3, 2, 1, 6	3, 12, 13, 14, 27	--
		±5	3, 12, 11, 22, 21	3, 2, 1, 6	3, 12, 13, 14, 27	--
		±10	3, 12, 11, 22, 21	3, 2, 1, 6	3, 12, 13, 14, 27	--
	II	0	3, 2, 10, 21	3, 2, 1, 6	3, 12, 13, 14, 27	3, 2, 1, 6, 18, 17, 16
		±3	3, 2, 10, 21	3, 2, 1, 6	3, 12, 13, 14, 27	3, 2, 1, 6, 18, 17, 16
		±5	3, 12, 11, 22, 21	3, 2, 1, 6	3, 12, 13, 14, 27	3, 2, 1, 6, 18, 17, 16
		±10	3, 12, 11, 22, 21	3, 2, 1, 6	3, 12, 13, 14, 27	BL
2	I	0	8, 9, 20, 21	8, 7, 18, 6	BL	--
		±3	8, 9, 20, 21	8, 7, 18, 6	BL	--
		±5	8, 9, 20, 21	8, 7, 1, 6	BL	--
		±10	8, 9, 20, 21	8, 7, 18, 6	BL	--
	II	0	8, 9, 20, 21	8, 7, 18, 6	BL	8, 7, 18, 17, 16
		±3	8, 9, 20, 21	8, 7, 18, 6	BL	8, 7, 18, 17, 16
		±5	8, 9, 20, 21	8, 7, 1, 6	BL	8, 7, 18, 17, 16
		±10	8, 9, 20, 21	8, 7, 18, 6	BL	8, 7, 18, 17, 16
3	I	0	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	--
		±3	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	--
		±5	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	--
		±10	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	--
	II	0	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	13, 4, 5, 15, 16
		±3	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	13, 4, 5, 15, 16
		±5	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	BL
		±10	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	13, 14, 15, 16
4	I	0	17, 18, 6, 1, 2, 10, 21	17, 18, 6	17, 16, 15, 14, 27	--
		±3	17, 18, 6, 1, 2, 10, 21	17, 18, 6	17, 16, 15, 14, 27	--
		±5	17, 18, 7, 8, 19, 20, 21	17, 18, 6	17, 16, 15, 14, 27	--
		±10	17, 18, 6, 1, 2, 10, 21	17, 18, 6	17, 16, 15, 14, 27	--
	II	0	17, 18, 6, 1, 2, 10, 21	17, 18, 6	BL	17, 16
		±3	17, 18, 6, 1, 2, 10, 21	17, 18, 6	BL	17, 16
		±5	17, 18, 7, 8, 19, 20, 21	17, 18, 6	BL	17, 16
		±10	17, 18, 6, 1, 2, 10, 21	17, 18, 6	BL	17, 16
5	I	0	29, 15, 5, 6, 1, 2, 10, 21	29, 15, 5, 6	29, 28, 27	--
		±3	29, 15, 5, 6, 1, 2, 10, 21	29, 15, 5, 6	29, 28, 27	--
		±5	29, 15, 5, 6, 1, 2, 10, 21	29, 15, 5, 6	29, 28, 27	--
		±10	29, 15, 5, 6, 1, 9, 20, 21	29, 15, 5, 6	29, 28, 27	--
	II	0	29, 15, 5, 6, 1, 2, 10, 21	29, 15, 5, 6	29, 28, 27	29, 15, 16
		±3	29, 15, 5, 6, 1, 2, 10, 21	29, 15, 5, 6	29, 28, 27	29, 15, 16
		±5	29, 15, 5, 6, 1, 2, 10, 21	29, 15, 5, 6	29, 28, 27	29, 15, 16
		±10	29, 15, 5, 6, 1, 9, 20, 21	29, 15, 5, 6	29, 28, 27	29, 30, 16
6	I	0	25, 24, 23, 22, 21	BL	25, 26, 27	--
		±3	25, 24, 23, 22, 21	25, 24, 13, 4, 5, 6	25, 26, 27	--
		±5	25, 24, 23, 22, 21	25, 24, 13, 4, 5, 6	25, 26, 27	--
		±10	25, 24, 23, 22, 21	25, 24, 13, 4, 5, 6	25, 26, 27	--
	II	0	25, 24, 23, 22, 21	BL	25, 26, 27	BL
		±3	25, 24, 23, 22, 21	25, 24, 13, 4, 5, 6	25, 26, 27	25, 24, 13, 4, 5, 15, 16
		±5	25, 24, 23, 22, 21	25, 24, 13, 4, 5, 6	25, 26, 27	BL
		±10	25, 24, 23, 22, 21	25, 24, 13, 4, 5, 6	25, 26, 27	BL

TABLE 11: Continued.

Vehicle no.	Example no.	Prediction error (km/h)	Accident no.			
			1	2	3	4
7	I	0	31, 32, 33, 19, 20, 21	31, 17, 18, 6	31, 30, 29, 28, 27	--
		±3	31, 32, 33, 19, 20, 21	31, 17, 18, 6	31, 30, 29, 28, 27	--
		±5	31, 32, 33, 19, 20, 21	31, 17, 18, 6	31, 30, 29, 28, 27	--
		±10	31, 32, 33, 19, 20, 21	31, 17, 18, 6	31, 30, 29, 43, 42, 41, 27	--
	II	0	31, 32, 33, 19, 20, 21	31, 17, 18, 6	31, 30, 29, 28, 27	31, 30, 16
		±3	31, 32, 33, 19, 20, 21	31, 17, 18, 6	31, 30, 29, 28, 27	31, 30, 16
		±5	31, 32, 33, 19, 20, 21	31, 17, 18, 6	31, 30, 29, 28, 27	31, 30, 16
		±10	31, 32, 33, 19, 20, 21	31, 17, 18, 6	31, 30, 29, 43, 42, 41, 27	31, 30, 16
8	I	0	32, 33, 19, 20, 21	32, 18, 6	BL	--
		±3	32, 33, 19, 20, 21	32, 18, 6	BL	--
		±5	32, 33, 19, 20, 21	32, 18, 6	BL	--
		±10	32, 33, 19, 20, 21	32, 18, 6	BL	--
	II	0	32, 33, 19, 20, 21	32, 18, 6	BL	32, 18, 17, 16
		±3	32, 33, 19, 20, 21	32, 18, 6	BL	32, 18, 17, 16
		±5	32, 33, 19, 20, 21	32, 18, 6	BL	32, 18, 17, 16
		±10	32, 33, 19, 20, 21	32, 18, 6	BL	32, 18, 17, 16
9	II	0	33, 19, 20, 21	33, 32, 18, 6	BL	33, 32, 18, 17, 16
		±3	33, 19, 20, 21	33, 32, 18, 6	BL	33, 32, 18, 17, 16
		±5	33, 19, 20, 21	33, 32, 18, 6	BL	33, 32, 31, 30, 16
		±10	33, 19, 20, 21	33, 32, 18, 6	BL	33, 32, 18, 17, 16
10	II	0	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	13, 4, 5, 15, 16
		±3	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	13, 4, 5, 15, 16
		±5	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	13, 4, 5, 15, 16
		±10	13, 12, 11, 22, 21	13, 4, 5, 6	13, 14, 27	13, 14, 15, 16

Note. BL means that the travel time of the optimal path exceeds the upper limit of the time window.

$v_{ij}(t)_{t_0}$ :	The polygonal-shaped travel speed function at the instant $t_0$	$Nv_l(t)$ :	The location node of the emergency vehicle $Ev_l(t)$
$Ls_{ij}$ :	The length of the road section $(n_i, n_j)$	$P_{lu}(t)$ :	When the emergency vehicle $Ev_l(t) \in EV(t)$ starts traveling at an instant $t$ , the shortest time path to the accident $Ac_u(t) \in AC(t)$
$U(t)$ :	The total number of accidents at the instant $t$	$T_{lu}(t)$ :	The travel time of the shortest time path $P_{lu}(t)$ at the instant $t$
$AC(t)$ :	The accident set at the instant $t$	$t_\varphi$ :	The $\varphi$ th decision instant
$Ac_u(t) \in AC(t)$ , $u = 1, 2, \dots, U(t)$ :	The $u$ th accident at the instant $t$	$x_{lu}(t_\varphi)$ :	The decision variable; at the decision-making instant $t_\varphi$ , if the emergency vehicle $Ev_l$ , $l = 1, 2, \dots, L$ , is dispatched to the accident $Ac_u$ , $u = 1, 2, \dots, U$ , then $x_{lu}(t_\varphi) = 1$ ; otherwise, $x_{lu}(t_\varphi) = 0$
$(n_0^u, n_1^u) \in E$ :	The road section where the accident $Ac_u(t)$ occurred at the instant $t$	$z_u(t_\varphi)$ :	The decision variable; at the decision-making instant $t_\varphi$ , if the latest rescue time for accident $Ac_u$ , $u = 1, 2, \dots, U$ exceeds the upper limit of rescue time window, then $z_u(t_\varphi) = 1$ ; otherwise, $z_u(t_\varphi) = 0$
$Nc_u(t)$ :	The location node of the accident $Ac_u(t)$	$M$ :	A huge constant
$T_{max}^u(t)$ :	The upper limit of rescue time window of the accident $Ac_u(t)$	$AC^1(t_\varphi)$ :	The accident set includes new accidents that happened at $t_\varphi$ and the accidents whose rescue paths at $t_{\varphi-1}$ are affected by new accidents
$Na_u(t)$ :	The required number of emergency vehicles for the accident $Ac_u(t)$		
$As_u(t)$ :	The severity of the accident $Ac_u(t)$		
$EV(t)$ :	The emergency vehicle set at the instant $t$		
$Ev_l(t) \in EV(t)$ , $l = 1, 2, \dots, L(t)$ :	The $l$ th emergency vehicle at the instant $t$		
$L(t)$ :	The total number of emergency vehicles at the instant $t$		
$(n_0^l, n_1^l) \in E$ :	The road section where the emergency vehicle $Ev_l(t)$ is located		

TABLE 12: Optimal emergency vehicle dispatching strategy of the illustrative examples based on the prediction link travel speed.

Example no.	Prediction error (km/h)	The optimal solution	Objective function value based on the prediction travel speed	Objective function value based on the real travel speed
I	0	0, 1, 3, 2, 0, 3, 0, 2	--	7987
	±3	0, 1, 3, 2, 0, 3, 0, 2	7897	7987
	±5	0, 1, 3, 2, 0, 3, 0, 2	7981	7987
	±10	0, 1, 3, 2, 0, 3, 2, 0	8461	8088
II	0	0, 1, 0, 2, 0, 3, 4, 2, 0, 3	8113	8113
	±3	0, 1, 0, 2, 0, 3, 4, 2, 0, 3	8071	8113
	±5	0, 1, 0, 2, 0, 3, 4, 2, 0, 3	8126	8113
	±10	0, 1, 0, 2, 0, 3, 4, 2, 0, 3	8569	8113
I	0	Optimal emergency vehicle dispatching strategy	Prediction travel time (min)	Real travel time (min)
		$Ev_2$ to $Ac_1$ ; $Nv_2 \rightarrow n_8 \rightarrow n_9 \rightarrow n_{20} \rightarrow n_{21} \rightarrow Nc_1$	--	22.1482
		$Ev_4$ to $Ac_2$ ; $Nv_4 \rightarrow n_{17} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	--	16.3454
		$Ev_8$ to $Ac_2$ ; $Nv_8 \rightarrow n_{32} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	--	21.7823
		$Ev_3$ to $Ac_3$ ; $Nv_3 \rightarrow n_{13} \rightarrow n_{14} \rightarrow n_{27} \rightarrow Nc_3$	--	33.8637
		$Ev_6$ to $Ac_3$ ; $Nv_6 \rightarrow n_{25} \rightarrow n_{26} \rightarrow n_{27} \rightarrow Nc_3$	--	26.3079
		$Ev_2$ to $Ac_1$ ; $Nv_2 \rightarrow n_8 \rightarrow n_9 \rightarrow n_{20} \rightarrow n_{21} \rightarrow Nc_1$	20.8670	22.1482
		$Ev_4$ to $Ac_2$ ; $Nv_4 \rightarrow n_{17} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	16.4075	16.3454
		$Ev_8$ to $Ac_2$ ; $Nv_8 \rightarrow n_{32} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	21.8997	21.7823
		$Ev_3$ to $Ac_3$ ; $Nv_3 \rightarrow n_{13} \rightarrow n_{14} \rightarrow n_{27} \rightarrow Nc_3$	33.7495	33.8637
		$Ev_6$ to $Ac_3$ ; $Nv_6 \rightarrow n_{25} \rightarrow n_{26} \rightarrow n_{27} \rightarrow Nc_3$	25.8029	26.3079
		$Ev_2$ to $Ac_1$ ; $Nv_2 \rightarrow n_8 \rightarrow n_9 \rightarrow n_{20} \rightarrow n_{21} \rightarrow Nc_1$	23.1840	22.1482
		$Ev_4$ to $Ac_2$ ; $Nv_4 \rightarrow n_{17} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	16.6789	16.3454
		$Ev_8$ to $Ac_2$ ; $Nv_8 \rightarrow n_{32} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	21.8541	21.7823
		$Ev_3$ to $Ac_3$ ; $Nv_3 \rightarrow n_{13} \rightarrow n_{14} \rightarrow n_{27} \rightarrow Nc_3$	33.8035	33.8637
		$Ev_6$ to $Ac_3$ ; $Nv_6 \rightarrow n_{25} \rightarrow n_{26} \rightarrow n_{27} \rightarrow Nc_3$	25.4729	26.3079
II	±10	$Ev_2$ to $Ac_1$ ; $Nv_2 \rightarrow n_8 \rightarrow n_9 \rightarrow n_{20} \rightarrow n_{21} \rightarrow Nc_1$	18.3767	22.1482
		$Ev_4$ to $Ac_2$ ; $Nv_4 \rightarrow n_{17} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	17.3145	16.3454
		$Ev_7$ to $Ac_2$ ; $Nv_7 \rightarrow n_{31} \rightarrow n_{17} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	24.9977	23.4594
		$Ev_3$ to $Ac_3$ ; $Nv_3 \rightarrow n_{13} \rightarrow n_{14} \rightarrow n_{27} \rightarrow Nc_3$	33.2153	33.8637
$Ev_6$ to $Ac_3$ ; $Nv_6 \rightarrow n_{25} \rightarrow n_{26} \rightarrow n_{27} \rightarrow Nc_3$	31.6198	26.3079		

TABLE 12: Continued.

	$Ev_2$ to $Ac_1$ : $Nv_2 \rightarrow n_8 \rightarrow n_9 \rightarrow n_{20} \rightarrow n_{21} \rightarrow Nc_1$	--	22.1482
	$Ev_4$ to $Ac_2$ : $Nv_4 \rightarrow n_{17} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	--	16.3454
	$Ev_8$ to $Ac_2$ : $Nv_8 \rightarrow n_{32} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	--	21.7823
	$Ev_6$ to $Ac_3$ : $Nv_6 \rightarrow n_{25} \rightarrow n_{26} \rightarrow n_{27} \rightarrow Nc_3$	--	26.3079
	$Ev_{10}$ to $Ac_3$ : $Nv_{10} \rightarrow n_{13} \rightarrow n_{14} \rightarrow n_{27} \rightarrow Nc_3$	--	28.2020
	$Ev_7$ to $Ac_4$ : $Nv_7 \rightarrow n_{31} \rightarrow n_{30} \rightarrow n_{16} \rightarrow Nc_4$	--	14.4705
	$Ev_2$ to $Ac_1$ : $Nv_2 \rightarrow n_8 \rightarrow n_9 \rightarrow n_{20} \rightarrow n_{21} \rightarrow Nc_1$	20.8670	22.1482
	$Ev_4$ to $Ac_2$ : $Nv_4 \rightarrow n_{17} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	16.4075	16.3454
	$Ev_8$ to $Ac_2$ : $Nv_8 \rightarrow n_{32} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	21.8997	21.7823
	$Ev_6$ to $Ac_3$ : $Nv_6 \rightarrow n_{25} \rightarrow n_{26} \rightarrow n_{27} \rightarrow Nc_3$	25.8029	26.3079
	$Ev_{10}$ to $Ac_3$ : $Nv_{10} \rightarrow n_{13} \rightarrow n_{14} \rightarrow n_{27} \rightarrow Nc_3$	28.5534	28.2020
	$Ev_7$ to $Ac_4$ : $Nv_7 \rightarrow n_{31} \rightarrow n_{30} \rightarrow n_{16} \rightarrow Nc_4$	14.7354	14.4705
	$Ev_2$ to $Ac_1$ : $Nv_2 \rightarrow n_8 \rightarrow n_9 \rightarrow n_{20} \rightarrow n_{21} \rightarrow Nc_1$	23.1840	22.1482
	$Ev_4$ to $Ac_2$ : $Nv_4 \rightarrow n_{17} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	16.6789	16.3454
	$Ev_8$ to $Ac_2$ : $Nv_8 \rightarrow n_{32} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	21.8541	21.7823
	$Ev_6$ to $Ac_3$ : $Nv_6 \rightarrow n_{25} \rightarrow n_{26} \rightarrow n_{27} \rightarrow Nc_3$	25.4729	26.3079
	$Ev_{10}$ to $Ac_3$ : $Nv_{10} \rightarrow n_{13} \rightarrow n_{14} \rightarrow n_{27} \rightarrow Nc_3$	28.6415	28.2020
	$Ev_7$ to $Ac_4$ : $Nv_7 \rightarrow n_{31} \rightarrow n_{30} \rightarrow n_{16} \rightarrow Nc_4$	13.9424	14.4705
	$Ev_2$ to $Ac_1$ : $Nv_2 \rightarrow n_8 \rightarrow n_9 \rightarrow n_{20} \rightarrow n_{21} \rightarrow Nc_1$	18.3767	22.1482
	$Ev_4$ to $Ac_2$ : $Nv_4 \rightarrow n_{17} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	17.3145	16.3454
	$Ev_8$ to $Ac_2$ : $Nv_8 \rightarrow n_{32} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$	26.2546	21.7823
	$Ev_6$ to $Ac_3$ : $Nv_6 \rightarrow n_{25} \rightarrow n_{26} \rightarrow n_{27} \rightarrow Nc_3$	31.6198	26.3079
	$Ev_{10}$ to $Ac_3$ : $Nv_{10} \rightarrow n_{13} \rightarrow n_{14} \rightarrow n_{27} \rightarrow Nc_3$	26.8230	28.2020
	$Ev_7$ to $Ac_4$ : $Nv_7 \rightarrow n_{31} \rightarrow n_{30} \rightarrow n_{16} \rightarrow Nc_4$	13.5972	14.4705

TABLE 13: Dynamic emergency vehicle dispatching process.

Decision making instant	Vehicle no.	Road section of vehicle	Location of vehicle	The optimal solution	Optimal emergency vehicle dispatching strategy
9:30	1	$(n_2, n_3)$	$L_{2,55} = L_{55,3}$	0,1,3,2,0,3,0,2	$Ev_2$ to $Ac_1$ : $Nv_2 \rightarrow n_8 \rightarrow n_9 \rightarrow n_{20} \rightarrow n_{21} \rightarrow Nc_1$
	2	$(n_9, n_8)$	$L_{9,56} = L_{56,8}$		$Ev_4$ to $Ac_2$ : $Nv_4 \rightarrow n_{17} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$
	3	$(n_{12}, n_{13})$	$L_{12,57} = L_{57,13}$		$Ev_8$ to $Ac_2$ : $Nv_8 \rightarrow n_{32} \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$
	4	$(n_{18}, n_{17})$	$L_{18,58} = L_{58,17}$		$Ev_3$ to $Ac_3$ : $Nv_3 \rightarrow n_{13} \rightarrow n_{14} \rightarrow n_{27} \rightarrow Nc_3$
	5	$(n_{15}, n_{29})$	$L_{15,59} = L_{59,29}$		$Ev_6$ to $Ac_3$ : $Nv_6 \rightarrow n_{25} \rightarrow n_{26} \rightarrow n_{27} \rightarrow Nc_3$
	6	$(n_{39}, n_{25})$	$L_{39,60} = L_{60,25}$		
	7	$(n_{30}, n_{31})$	$L_{30,61} = L_{61,31}$		
	8	$(n_{33}, n_{32})$	$L_{33,62} = L_{62,32}$		
9:35	1	$(n_2, n_3)$	$L_{2,55} = L_{55,3}$	1,4,3,2,0,3,0,2	$Ev_1$ to $Ac_1$ : $Nv_1 \rightarrow n_3 \rightarrow n_2 \rightarrow n_{10} \rightarrow n_{21} \rightarrow Nc_1$
	2	$(n_8, n_9)$	$L_{8,56} = 0.12 \times L_{8,9}$ $L_{56,9} = 0.88 \times L_{8,9}$		$Ev_4$ to $Ac_2$ : $Nv_4 \rightarrow n_6 \rightarrow Nc_2$ $Ev_8$ to $Ac_2$ : $Nv_8 \rightarrow n_{18} \rightarrow n_6 \rightarrow Nc_2$
	3	$(n_{12}, n_{13})$	$L_{12,57} = 0.77 \times L_{12,13}$ $L_{57,13} = 0.23 \times L_{12,13}$		$Ev_3$ to $Ac_3$ : $Nv_3 \rightarrow n_{13} \rightarrow n_{14} \rightarrow Nc_3$ $Ev_6$ to $Ac_3$ : $Nv_6 \rightarrow n_{25} \rightarrow n_{26} \rightarrow n_{27} \rightarrow Nc_3$
	4	$(n_{18}, n_6)$	$L_{18,58} = 0.10 \times L_{18,6}$ $L_{58,6} = 0.90 \times L_{18,6}$		$Ev_2$ to $Ac_4$ : $Nv_2 \rightarrow n_9 \rightarrow Nc_4$
	5	$(n_{15}, n_{29})$	$L_{15,59} = L_{59,29}$		
	6	$n_{25}$	$n_{25}$		
	7	$(n_{30}, n_{31})$	$L_{30,61} = L_{61,31}$		
	8	$(n_{32}, n_{18})$	$L_{32,62} =$ $0.16 \times L_{32,18}$ $L_{62,18} =$ $0.84 \times L_{32,18}$		

$t_{\max}^u(t_\varphi)$ :	The latest rescue time of the accident $Ac_u(t_\varphi)$	$Pb = [pb_1, pb_2, \dots, pb_d, \dots, pb_D]$ :	The position of the best frog in the memplex
$xx_l(t_\varphi)$ :	The decision variable; at the decision-making instant $t_\varphi$ , if the emergency vehicle $Ev_l$ , $l = 1, 2, \dots, L$ is in the idle state, then $xx_l(t_\varphi) = 1$ ; otherwise, $xx_l(t_\varphi) = 0$	$Pw = [pw_1, pw_2, \dots, pw_d, \dots, pw_D]$ :	The position of the worst frog in the memplex
$T_{i,i+1}(t_i)_{t_\varphi}$ :	The link travel time function at the decision-making instant $t_\varphi$ . It represents a time for the emergency vehicle, leaving at an instant $t_i$ , from node $n_i$ to $n_{i+1}$	$Ds_d, d = 1, 2, \dots, D$ :	The adjustment of the $d$ th decision variable
$H$ :	The total number of frogs	$r \in [0, 1]$ :	A random number
IP:	The frog population	$D_d^{\max}$ :	The maximum adjustment of the $d$ th decision variable
$X_h = [x_{h1}, x_{h2}, \dots, x_{hd}, \dots, x_{hD}]$ , $h = 1, \dots, H$ :	The position of the $h$ th frog	$pw'_d$ :	The renewed position of the $d$ th decision variable
$D$ :	The dimension of the optimization problem	$Z_d^{\max}$ :	The upper limits of the position of the $d$ th decision variable
$f$ :	The performance function of the optimization problem	$Z_d^{\min}$ :	The lower limits of the position of the $d$ th decision variable
$Px = [px_1, px_2, \dots, px_d, \dots, px_D]$ :	The position of the optimal frog in the population IP	IT:	The total number of global iterations
$a$ :	The total number of memplexes	$x_l(t_\varphi)$ :	The coding of the decision variable $x_{lu}(t_\varphi)$ ; it represents the dispatching strategy of $Ev_l(t_\varphi)$ , $l = 1, 2, \dots, L$ , at decision-making instant $t_\varphi$
$c$ :	The total number of frogs in each memplex	$B_{U \times L}$ :	The probability distribution matrix of the frog population
It:	The total number of iterations within each memplex	$b_{u \times l} \in [0, 1]$ :	The element of the matrix $B_{U \times L}$ ; it represents the probability that the $l$ th decision variable $x_l(t_\varphi)$ is valued as $u$
		$\theta$ :	The forgetting factor.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

## Acknowledgments

This work is supported by Funding for Scientific Research Startup of North China University of Technology.

## References

- [1] T. Yamada, "A network flow approach to a city emergency evacuation planning," *International Journal of Systems Science*, vol. 27, no. 10, pp. 931–936, 1996.
- [2] G. M. Carter, J. M. Chaiken, and E. Ignall, "Response areas for two emergency units," *Operations Research*, vol. 20, pp. 571–594, 1972.
- [3] H. D. Sherali and S. Subramanian, "Opportunity cost-based models for traffic incident response problems," *Journal of Transportation Engineering*, vol. 125, no. 3, pp. 176–185, 1999.
- [4] K. Ozbay, C. Iyigun, M. Baykal-Gursoy, and W. Xiao, "Probabilistic programming models for traffic incident management operations planning," *Annals of Operations Research*, pp. 1–18, 2012.
- [5] X. Song, J. Wang, and C. Chang, "Nonlinear continuous consumption emergency material dispatching problem," *Systems Engineering*, vol. 32, no. 2, pp. 163–176, 2017.
- [6] W. Tang, L. Zou, and Q. Guo, "Grey multi-objective programming for materials dispatching from multiple supply points to multiple demand points," *Open Journal of Safety Science and Technology*, vol. 12, no. 11, pp. 148–152, 2016.
- [7] J. Liu and K. Xie, "Emergency materials transportation model in disasters based on dynamic programming and ant colony optimization," *Kybernetes*, vol. 46, no. 4, pp. 656–671, 2017.
- [8] X.-H. Li and Q.-M. Tan, "Particle swarm optimization algorithm for emergency resources dispatch scheduling," *Recent Patents on Mechanical Engineering*, vol. 7, no. 3, pp. 237–240, 2014.
- [9] J. Liu, L. Guo, and J. Jiang, "Emergency material allocation and scheduling for the application to chemical contingency spills under multiple scenarios," *Environmental Science Pollution Research*, vol. 24, no. 1, p. 13, 2017.
- [10] K. G. Zografos, K. N. Androutopoulos, and G. M. Vasilakis, "A real-time decision support system for roadway network incident response logistics," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 1, pp. 1–18, 2002.
- [11] A. Haghani, Q. Tian, and H. Hu, "Simulation model for real-time emergency vehicle dispatching and routing," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1882, pp. 176–183, 2004.
- [12] B. Dan, W. Zhu, H. Li, Y. Sang, and Y. Liu, "Dynamic optimization model and algorithm design for emergency materials dispatch," *Mathematical Problems in Engineering*, vol. 2013, Article ID 841458, 2013.
- [13] S. Yang, M. Hamedi, and A. Haghani, "Online dispatching and routing model for emergency vehicles with area coverage constraints," *Transportation Research Record*, no. 1923, pp. 1–8, 2005.
- [14] B. Fu, D. G. Li, and M. K. Wang, "Review and prospect on research of cloud model," *Application Research of Computers*, vol. 28, no. 2, pp. 420–426, 2011.
- [15] K. L. Cooke and E. Halsey, "The shortest route through a network with time-dependent internodal transit times," *Journal of Mathematical Analysis and Applications*, vol. 14, pp. 493–498, 1966.
- [16] D. E. Kaufman and R. L. Smith, "Minimum travel time paths in dynamic networks with application to intelligent vehicle/highway systems," *IVHS Journal*, vol. 1, no. 1, pp. 1–19, 1993.
- [17] M. Setak, M. Habibi, H. Karimi, and M. Abedzadeh, "A time-dependent vehicle routing problem in multigraph with FIFO property," *Journal of Manufacturing Systems*, vol. 35, pp. 37–45, 2015.
- [18] X.-H. Duan, J.-D. Zhao, and S.-X. Song, "Dynamic shortest paths of emergency vehicles in expressway network based on shuffled frog leaping algorithm," *Journal of Transportation Systems Engineering and Information Technology*, vol. 16, no. 3, pp. 181–186, 2016.
- [19] S. Ichoua, M. Gendreau, and J. Potvin, "Vehicle dispatching with time-dependent travel times," *European Journal of Operational Research*, vol. 144, no. 2, pp. 379–396, 2003.
- [20] Y. Yuan and D. Wang, "Path selection model and algorithm for emergency logistics management," *Computers & Industrial Engineering*, vol. 56, no. 3, pp. 1081–1094, 2009.
- [21] Y. Zhou, J. Liu, Y. Zhang, and X. Gan, "A multi-objective evolutionary algorithm for multi-period dynamic emergency resource scheduling problems," *Transportation Research Part E: Logistics and Transportation Review*, vol. 99, pp. 77–95, 2017.
- [22] Y. Zhou and J. Liu, "A multi-agent genetic algorithm for multi-period emergency resource scheduling problems in uncertain traffic network," in *Proceedings of the 2017 IEEE Congress on Evolutionary Computation, CEC 2017*, pp. 43–50, Spain, June 2017.
- [23] M. M. Eusuff and K. E. Lansey, "Optimization of water distribution network design using the shuffled frog leaping algorithm," *Journal of Water Resources Planning and Management*, vol. 129, no. 3, pp. 210–225, 2003.
- [24] M. Eusuff, K. Lansey, and F. Pasha, "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization," *Engineering Optimization*, vol. 38, no. 2, pp. 129–154, 2006.
- [25] H. Amirian and R. Sahraeian, "Solving a grey project selection scheduling using a simulated shuffled frog leaping algorithm," *Computers & Industrial Engineering*, vol. 107, pp. 141–149, 2017.
- [26] D. Lei and X. Guo, "A shuffled frog-leaping algorithm for job shop scheduling with outsourcing options," *International Journal of Production Research*, vol. 54, no. 16, pp. 4793–4804, 2016.
- [27] D. Lei, Y. Zheng, and X. Guo, "A shuffled frog-leaping algorithm for flexible job shop scheduling with the consideration of energy consumption," *International Journal of Production Research*, vol. 55, no. 11, pp. 3126–3140, 2016.
- [28] D. Lei and X. Guo, "A shuffled frog-leaping algorithm for hybrid flow shop scheduling with two agents," *Expert Systems with Applications*, vol. 42, no. 23, pp. 9333–9339, 2015.
- [29] B. Ning, Q. Gu, and Y. Wang, "Research based on effective resource allocation of improved SFLA in cloud computing," *International Journal of Grid and Distributed Computing*, vol. 9, no. 3, pp. 191–198, 2016.

- [30] X. Duan, S. Song, and J. Zhao, "Emergency vehicle dispatching and redistribution in highway network based on bilevel programming," *Mathematical Problems in Engineering*, vol. 2015, Article ID 731492, pp. 1–12, 2015.
- [31] T. Xu, *Study on Dynamic Prediction of Road Operating Speed Based on Floating Car Data*, Chongqing Jiaotong University, 2017.
- [32] J. Weng, J. Rong, F. Ren et al., "Non-parametric regression model based short-term prediction for expressway travel speed," *Journal of Highway Transportation Research and Development*, vol. 24, no. 3, pp. 93–97, 2007.

## Research Article

# A Position-Level Global Optimization Inverse Kinematic Solution Algorithm for Dual Redundant Robots Based on Motion Characteristics

Jingjie He 

AICFVE of Beijing Film Academy, Beijing, China

Correspondence should be addressed to Jingjie He; [hejingjie4@sina.com](mailto:hejingjie4@sina.com)

Received 30 October 2017; Revised 22 January 2018; Accepted 14 February 2018; Published 26 March 2018

Academic Editor: Paula L. Zabala

Copyright © 2018 Jingjie He. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Camera robot is an important tool for realizing and reproducing complex camera motion path in modern special film effects. This paper proposed an inverse kinematics optimization algorithm for PRRPR-S redundant degrees of freedom (DoF) camera robot. This paper analyzed the motion characteristics, in Genetic Mix (GM) method, from the idea of movement boundary composed of part robot axis. Then proposed Simplify Mix (SM) method which can stably converge to the global optimal solution in a shorter time.

## 1. Introduction

*1.1. Project Requirements Analysis.* The PRRPR-S robot, referred to as camera robot, is an important tool of reproducing camera movement for multilayer composite film effects and realizing complex camera motion path. It is an important interactive node in film virtual manufacturing [1]. At present, the camera shooting trajectory is obtained by teaching mode, and the camera motion data in the preview system cannot be used directly. This paper studied the inverse kinematic algorithm of camera robot with global optimization ability, which makes camera move directly and transparently in a shorter time.

*1.2. Theory Requirements Analysis.* The algebraic equation form of redundancy serial robot inverse kinematics on velocity level is linear style [2], and the position-level is expressed as a nonlinear system and usually solved by numerical iteration. Liegeois [3] proposed gradient projection method based on generalized inverse matrix, which got the local optimal solution on velocity level but there was a cumulative error. Parker et al. [4] used genetic algorithm (GA) to solve the kinematic equations of redundant DoF manipulators. Sung et al. [5] took the terminal actuator positional deviation as the optimization target on the basis of the forward kinematics

equation, which obtained the global optimal solution. The algorithm runs for a long time, and the multi-optimization objective function is hard to construct.

Masayuki et al. [6] and Singh and Claassens [7] analyzed the single redundant DoF serial robot, which determined the self-motion space, and converted the redundancy into arm angle. The method is easy to construct the objective function.

The camera robot needs to get the position and attitude of the end-effector exactly in practice. And it has 8 joints, corresponding to high dimension of solution space, in other words, low effective solution proportion space. By using GA method based on forward kinematics equations, it is difficult to find valid solutions and attach other optimization goals. In addition, the randomness of the initial population makes the existence probability of the effective individuals very low, and the algorithm convergence effect is poor. Taking the axes of  $r_1$  and  $r_4$  as redundancy, Zhang et al. [8] proposed Genetic Mix (GM) method which include motion characteristics, GA based on redundant DoF and pattern search. By presetting the initial population, GM overcomes the defects of randomness and sequence in GA and converges to the global optimal solution stably on position-level. But in literature [8], there is no discussion about key analysis of how to get redundant DoF theoretical effect solution area. This paper, first, discussed the key analysis in detail, which

TABLE 1: Robot link parameters.

Transform	$\theta$	$r$	$l$	$\alpha$
0	$\frac{\pi}{2}$	0	0	$\frac{\pi}{2}$
1	0	$r_1 + r_{10}$	0	$-\frac{\pi}{2}$
2	$\theta_2 - \frac{\pi}{2}$	$r_2$	0	$-\frac{\pi}{2}$
3	$\theta_3 - \frac{\pi}{2}$	0	$l_3$	$-\frac{\pi}{2}$
4	0	$r_4 + r_{40}$	0	$\frac{\pi}{2}$
5	$\theta_5 + \frac{\pi}{2}$	0	0	$-\frac{\pi}{2}$
6	$\theta_6$	$r_6$	0	$\frac{\pi}{2}$
7	$\theta_7 - \frac{\pi}{2}$	0	0	$-\frac{\pi}{2}$
8	$\theta_8$	0	0	0

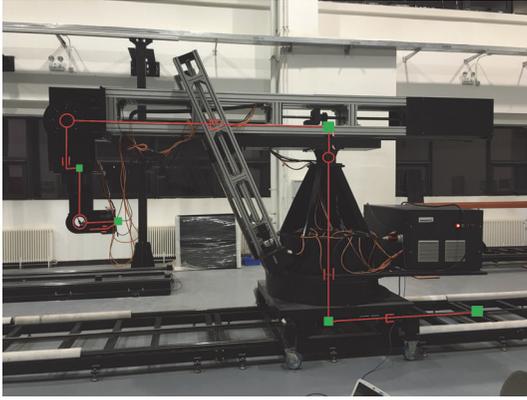


FIGURE 1: Camera robot.

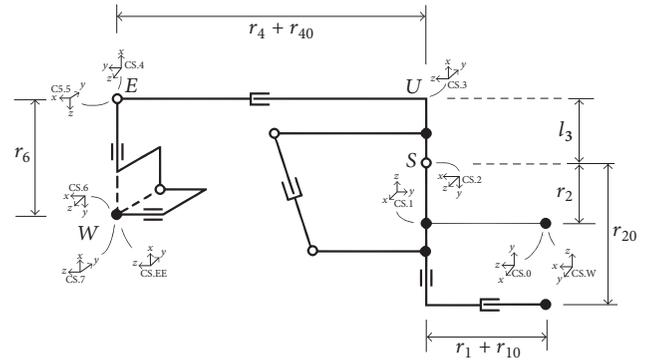


FIGURE 2: Schematic diagram of camera robot mechanism.

regarded as the most extended/contract state, and, second, proposed Simplify Mix (SM) method which reduces the time consumption further.

## 2. Motion Characteristic Analysis

**2.1. Camera Robot Workspace.** The camera robot is shown in Figure 1. There are 8 axes, including bottom linear axis  $r_1$ , ring style bottom rotate axis  $\theta_2$ , ring style top rotate axis  $\theta_3$ , top linear axis  $r_4$ , elbow axis  $\theta_5$ , rotate of wrist  $\theta_6$ , tilt of wrist  $\theta_7$ , and roll of wrist  $\theta_8$ . Modeled with D-H method, the robot "initial standard zero state" and coordinate system on the link are shown in Figure 2.

The kinematic link parameters are shown in Table 1.

$r_{10}$  and  $r_{40}$  are fixed value by mechanism. S represents the shoulder point, E represents the elbow point, W represents the wrist point, and U represents the upper arm point (U is not an axis).

The DH matrices of camera robot are as follows:

$$A_0 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & r_1 + r_{10} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} \cos\left(\theta_2 - \frac{\pi}{2}\right) & 0 & -\sin\left(\theta_2 - \frac{\pi}{2}\right) & 0 \\ \sin\left(\theta_2 - \frac{\pi}{2}\right) & 0 & \cos\left(\theta_2 - \frac{\pi}{2}\right) & 0 \\ 0 & -1 & 0 & r_2 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\begin{aligned}
 A_3 &= \begin{bmatrix} \cos\left(\theta_3 - \frac{\pi}{2}\right) & 0 & -\sin\left(\theta_3 - \frac{\pi}{2}\right) & l_3 \cdot \cos\left(\theta_3 - \frac{\pi}{2}\right) \\ \sin\left(\theta_3 - \frac{\pi}{2}\right) & 0 & \cos\left(\theta_3 - \frac{\pi}{2}\right) & l_3 \cdot \sin\left(\theta_3 - \frac{\pi}{2}\right) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 A_4 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & r_4 + r_{40} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 A_5 &= \begin{bmatrix} \cos\left(\theta_5 + \frac{\pi}{2}\right) & 0 & -\sin\left(\theta_5 + \frac{\pi}{2}\right) & 0 \\ \sin\left(\theta_5 + \frac{\pi}{2}\right) & 0 & \cos\left(\theta_5 + \frac{\pi}{2}\right) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 A_6 &= \begin{bmatrix} \cos(\theta_6) & 0 & \sin(\theta_6) & 0 \\ \sin(\theta_6) & 0 & -\cos(\theta_6) & 0 \\ 0 & 1 & 0 & r_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 A_7 &= \begin{bmatrix} \cos\left(\theta_7 - \frac{\pi}{2}\right) & 0 & -\sin\left(\theta_7 - \frac{\pi}{2}\right) & 0 \\ \sin\left(\theta_7 - \frac{\pi}{2}\right) & 0 & \cos\left(\theta_7 - \frac{\pi}{2}\right) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 A_8 &= \begin{bmatrix} \cos(\theta_8) & \sin(\theta_8) & 0 & 0 \\ -\sin(\theta_8) & \cos(\theta_8) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{1}
 \end{aligned}$$

The values of Table 1 are as follows:  $r_{10} = 1000$ ,  $r_2 = 300$ ,  $l_3 = 333$ ,  $r_{40} = 2500$ , and  $r_6 = 963$ . Then giving axes value, any posture of robot link can be calculated by transfer equations. For example, the posture of end-effector is

$$T_e = A_0 \cdot A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5 \cdot A_6 \cdot A_7 \cdot A_8. \tag{2}$$

Temporarily assume that the end-effector of the camera robot can get any posture through the wrist. As the robot moving in the bottom linear rail, the reachable space of  $W$  in the plane perpendicular to the rail represents the robot workspace. Based on the transfer matrix, as (2), the workspace coordinates of the key points and boundary equation can be calculated, as shown in Figure 3. The red curve is the outer boundary, and the green curve is the inner boundary.

**2.2. Redundancy Setting.** Considering that the industrial 6-DoF robot has mature algorithm of inverse kinematic solution, the bottom linear axis ( $r_1$ ) and the top linear axis ( $r_4$ ) are chosen as redundancy. After arbitrarily determining a set of

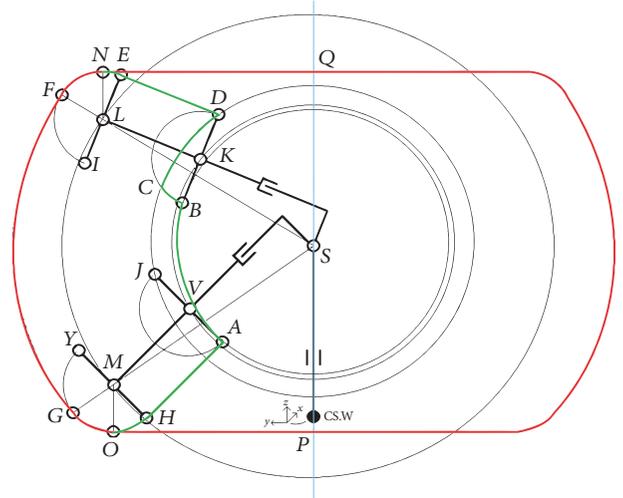


FIGURE 3: Workspace boundary of the camera robot.

$[r_1 \ r_4]$ , countable inverse solutions can be obtained. Choose the solution with the smallest value of the optimal function as the optimal solution, which reduces the 8-dimensional solution space to 2-dimensional one.

For clarity, declare the axis physical motion range as physical constraint and geometric constraint as theoretical constraint. The intersection of two constraints is the range which contains effective solution, called comprehensive constraint.

When using GA method with physical constraints of  $[r_1 \ r_4]$  for the gene range, due to the fact that the wrist attitude adjustment mechanism may block the view of camera, the effective solution share is very low and the algorithm is inefficient. In order to solve this problem, it is necessary to further analyze dual redundancy effective range. The basic idea is that, after a given posture, the robot should have two ways as “most extended state” and “the most contract state” to achieve the specified posture.

**2.3. Analysis of  $r_1$  Comprehensive Constraint.** Note the following:

- (1)  $WS$  is the distance between shoulder point ( $S$ ) and wrist point ( $W$ ).
- (2)  $pT$  is a 3-dimensional vector and represents the coordinate of target point position.
- (3)  $tS$  is the trace of shoulder point.

As for the rail can be extended to any length, the physical constraint of  $r_1$  is not in consideration.

At different heights, there are specific theoretical maximum  $WS$ ,  $TWS_{max}$ , and the theoretical minimum  $WS$ ,  $TWS_{min}$ , as shown in Figure 4.

**2.3.1.  $WS_{min}$  Determination.** Ask  $pT$  perpendicular to  $tS$ , as  $dpTtSM$ , and the foot is  $ppTtSM$ . If  $dpTtSM \leq TWS_{min}$ , refer to the lateral deviation of  $pT$  that is not big enough, as  $T_2$  shown in Figure 5. There is  $WS_{min} = TWS_{min}$ .

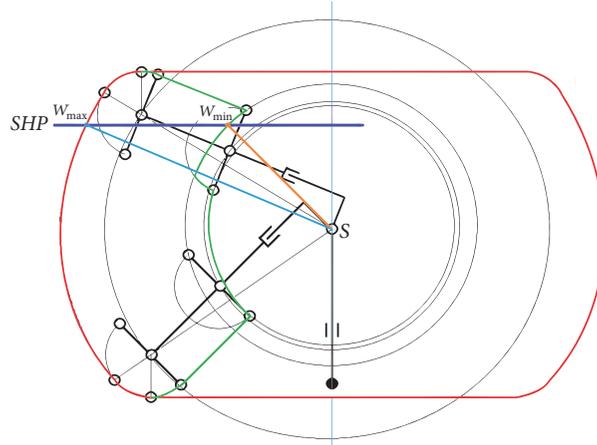
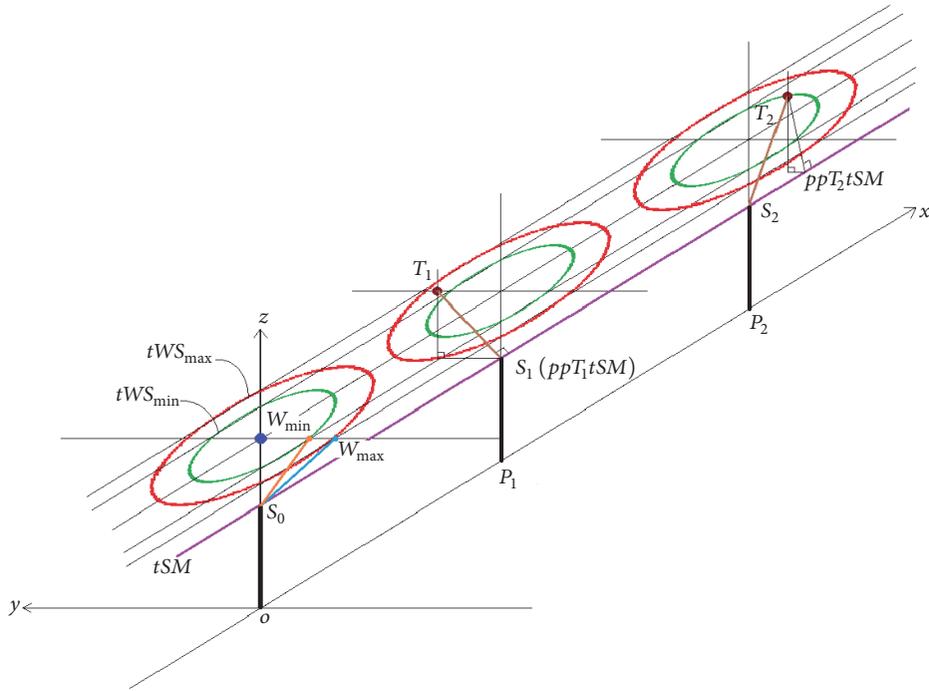


FIGURE 4: Theoretical max/min WS.

FIGURE 5: Reaching different  $pT$  at the specified height.

If  $TWS_{\min} < dpTtSM \leq TWS_{\max}$ , refer to the lateral deviation of target point that is big enough, as  $T_1$  shown in Figure 5. There is  $WS_{\min} = \|\mathbf{pT} - \mathbf{S}_1\|$ .

Thus,  $WS_{\min}$  can be determined based on the  $pT$ .

When the camera robot reaches  $pT$  in the most contract state, there are three cases according to the height of the target point and the degree of lateral deviation, as shown in Figure 6. According to  $dpTtSM$  and  $WS_{\min}$ , the limit position of  $S$  can be obtained, and then determine the limit value of  $r_1$ .

**2.3.2.  $WS_{\max}$  Determination.** When the camera robot reaches  $pT$  in the most extended state,  $W$  must be on the arc, in which

$TWS_{\max}$  is the radius and the projection of  $S$  on horizon plane passed  $pT$  is the center, as shown in Figure 7.

According to  $dpTtSM$  and  $TWS_{\max}$ , obtain the limit position of  $S$ , so as to determine the limit value of  $r_1$ . In summary, according to  $pT$  as input,  $r_1$  range is

$$r_1 \in [r_{1\text{farn}}, r_{1\text{close}}] \cup [r_{1\text{close}}, r_{1\text{farp}}]. \quad (3)$$

#### 2.4. Analysis of $r_4$ Comprehensive Constraint

**2.4.1. The Factors of  $r_4$  Determination Analysis.** After arbitrarily determining  $r_1$  in the comprehensive constraint, there are four factors that limit the value of  $r_4$ :

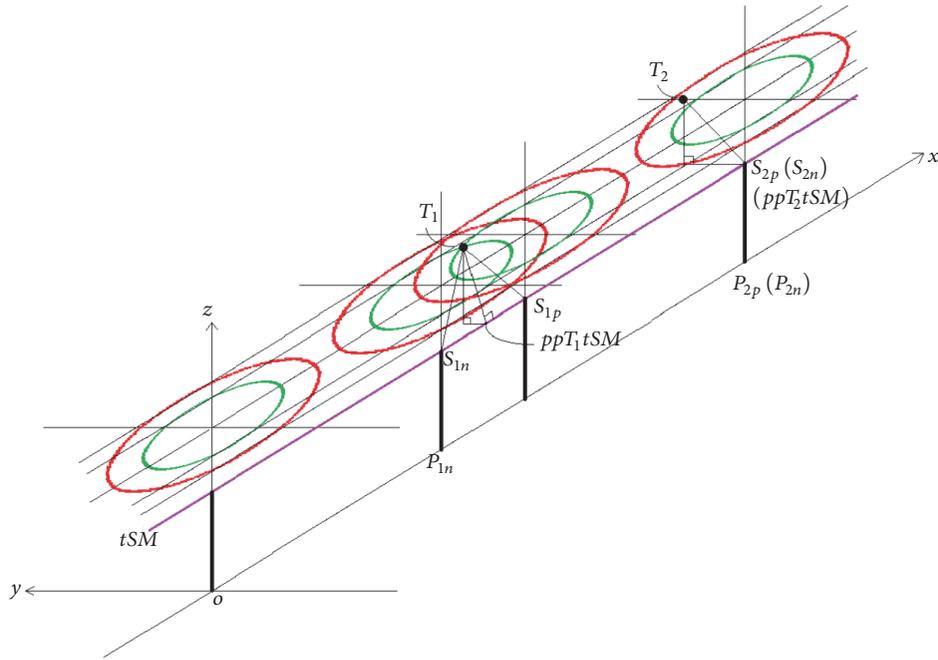


FIGURE 6: Reaching  $pT$  with  $WS_{\min}$ .

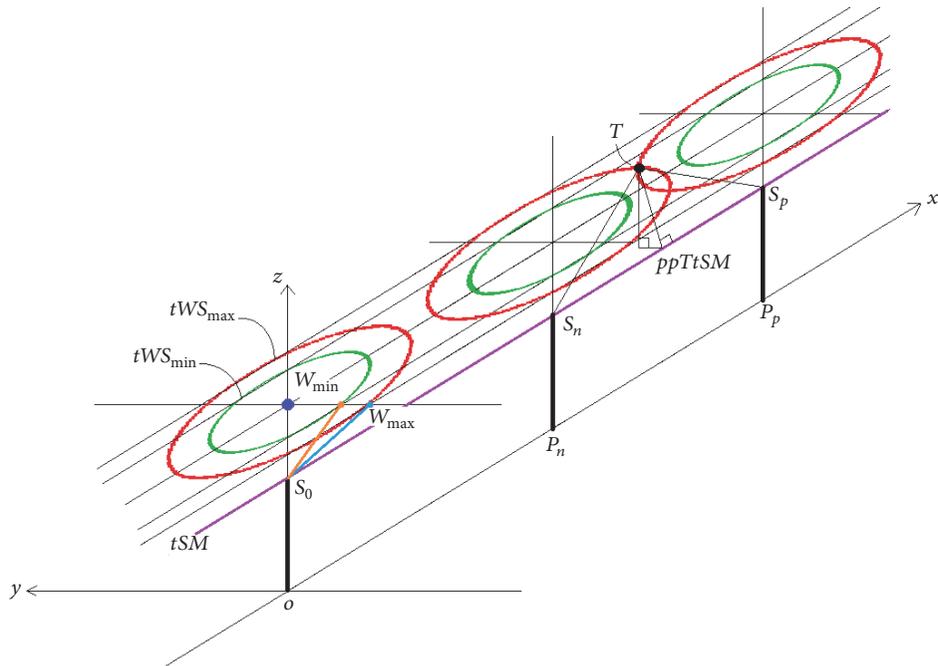


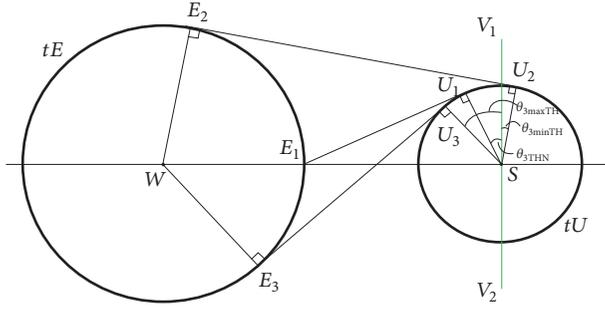
FIGURE 7: Reaching  $pT$  with  $WS_{\min}$ .

- (1)  $SE, WE, WS$  constitute a generalized triangle, as  $r_4 \in [r_{4\min f1}, r_{4\max f1}]$ .
- (2)  $UE$  moves within its physical constraint, as  $r_4 \in [r_{4\min}, r_{4\max}]$ .
- (3)  $\theta_5$  moves within its physical constraint, as  $r_4 \in [r_{4\min f3}, r_{4\max f3}]$ .

- (4)  $\theta_3$  moves within its comprehensive constraint, as  $r_4 \in [r_{4\min f4}, r_{4\max f4}]$ .

The inclusion relation of factor sets is

$$\begin{aligned}
 [r_{4\min f4}, r_{4\max f4}] &\subseteq [r_{4\min f3}, r_{4\max f3}] \\
 &\subset [r_{4\min f1}, r_{4\max f1}].
 \end{aligned}
 \tag{4}$$

FIGURE 8: The physical limit state of  $\theta_3$  when  $WS$  is horizontal.

That factor 4 included in factor 3 will be approved in the following context. Thus while  $pT$  is in the workspace and  $S$  is determined, there are

$$r_4 \in [r_{4\min}, r_{4\max}] \cap [r_{4\minsf4}, r_{4\maxsf4}]. \quad (5)$$

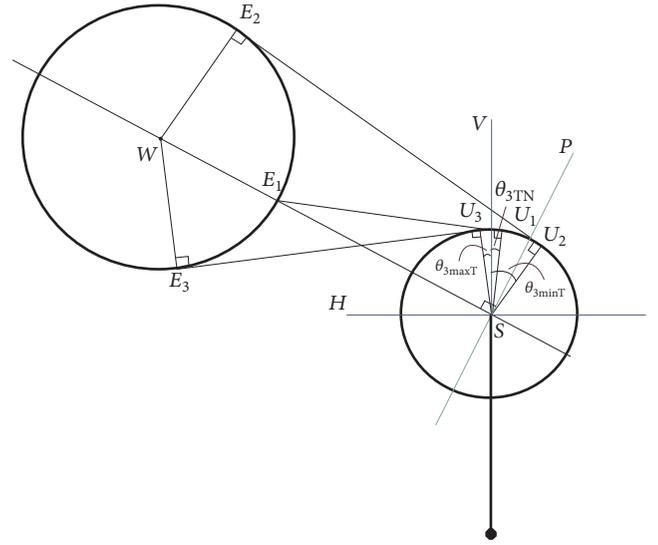
Select an arbitrary  $r_4$  in this range which can guarantee that at least one effective inverse solution is obtained. So it is important to analyze impact of factor 4 to  $r_4$ .

**2.4.2. Analysis of Factor 4.** Given the target posture,  $S$  is fixed after selecting  $r_1$ . For  $WE$  is fixed,  $E$  must be on the circle  $tE$ , whose center is  $W$  and radius is  $WE$ . Similarly,  $US$  length is fixed, so  $U$  point must be on the circle  $tU$ .  $\angle EUS$  is right angle as mechanism set.  $UE$  length is  $r_4 + r_{40}$ , as shown in Figure 8. Regardless of the height of  $W$ ,  $WS$  length is the main factor affecting the length of the  $UE$ .

Take  $WS$  in horizontal as the benchmark situation. When  $\theta_3$  rotates and  $E$  is on the connection of  $WS$ ,  $UE$  gets smallest length on the  $WS$  connection. When  $WE$  is perpendicular to  $UE$ ,  $UE$  takes the extreme value. Since  $\theta_5$  physical constraint is  $[-\pi, 0]$ , the limits of  $r_4$  by  $\theta_3$  theoretical constraint at  $WS$  in horizontal state and by  $\theta_5$  physical constraint are the same.

According to the geometric principal,  $\theta_3$  theoretical constraint is determined:

$$\begin{aligned} \theta_{3\minTH} &= \frac{\pi}{2} - \arctan\left(\frac{\sqrt{WS^2 - (r_6 - l_3)^2}}{l_3}\right) \\ &\quad - \arccos\left(\frac{WS^2 - r_6^2 + r_6 \cdot l_3}{\sqrt{WS^2 - r_6^2 + 2 \cdot r_6 \cdot l_3 \cdot WS}}\right), \\ \theta_{3\maxTH} &= \pi - \left[\left(\frac{\pi}{2} - \angle E_3SW\right) + \angle E_3SU_3\right] \\ &= \frac{\pi}{2} + \angle E_3SW - \angle E_3SU_3 \\ &= \frac{\pi}{2} \end{aligned}$$

FIGURE 9:  $WS$  in general case.

$$\begin{aligned} &+ \arccos\left(\frac{WS^2 - r_6^2 - r_6 \cdot l_3}{\sqrt{WS^2 - r_6^2 - 2 \cdot r_6 \cdot l_3 \cdot WS}}\right) \\ &- \arctan\left(\frac{\sqrt{WS^2 - (r_6 + l_3)^2}}{l_3}\right). \end{aligned} \quad (6)$$

The general case when  $WS$  at an arbitrary angle to the horizontal plane is shown in Figure 9.

$PS$  is perpendicular to  $WS$ . While specifying  $pT$ , the angle of  $WS$  and the horizontal plane  $H$  is  $\angle WSH$ . There are

$$\begin{aligned} \theta_{3\maxT} &= \theta_{3\maxTH} + \angle WSH, \\ \theta_{3\minT} &= \theta_{3\minTH} + \angle WSH, \\ \theta_{3TN} &= \theta_{3THN} + \angle WSH. \end{aligned} \quad (7)$$

In general, when specifying  $pT$  and  $S$ , three theoretical characteristic positions of  $\theta_3$  are obtained. Since  $\theta_3$  has a physical constraint, as  $\theta_{3\min} \leq \theta_3 \leq \theta_{3\max}$ . Therefore, we need to take  $\theta_3$  comprehensive constraint intervals and position of  $\theta_{3THN}$  into account.

**2.4.3. TRFour Function.** Before interpartition analysis, discuss the function:

$$r_{4T} = TRFour(W, S, \theta_3). \quad (8)$$

This function may get  $r_4$  that is outside the physical constraint. First, obtain  $\theta_{3TH}$  corresponding to the  $WS$  horizontal state:

$$\theta_{3TH} = \theta_3 - \angle WSH. \quad (9)$$

Then,

$$\angle WSU = \left| \frac{\pi}{2} - \theta_{3TH} \right|. \quad (10)$$

In  $\triangle WSU$ , by cosine theorem, there is

$$WU = \sqrt{WS^2 + US^2 - 2 \cdot WS \cdot US \cdot \cos(\angle WSU)}. \quad (11)$$

At the same time, there is

$$\begin{aligned} \angle WUS &= \arccos\left(\frac{WU^2 + US^2 - WS^2}{2 \cdot WU \cdot US}\right), \\ \angle WUE &= \left|\angle WUS - \frac{\pi}{2}\right|. \end{aligned} \quad (12)$$

In  $\triangle WUE$ , by cosine theorem, there is

$$WE^2 = WU^2 + UE^2 - 2 \cdot WU \cdot UE \cdot \cos(\angle WUE). \quad (13)$$

Considering position of line with  $UE$ , there is

$$\begin{aligned} UE &= WU \cdot \cos(\angle WUE) \\ -\sqrt{WE^2 - WU^2 \cdot \sin(\angle WUE)^2} &= r_{4T} + r_{40}. \end{aligned} \quad (14)$$

The result is

$$r_{4T} = UE - r_{40}. \quad (15)$$

**2.4.4. Analysis of  $\theta_3$  Effective Comprehensive Constraint.** After determining  $W$  and  $S$ , discuss  $\theta_3$  comprehensive constraint. Based on this, discuss  $r_4$  comprehensive constraint.

The blue sector (filled with upper right oblique line) is  $\theta_3$  theoretical constraint. The red sector (filled with upper left oblique line) is  $\theta_3$  physical constraint. The green sector (filled with cross line) is the overlapping area, which refers to comprehensive constraint. The purple line is  $\theta_{3TN}$  position. Among them, the limit position and scale of  $\theta_3$  physical constraint do not change, and there always is  $\theta_{3minT} < \theta_{3TN} < \theta_{3maxT}$ .

According to the following principles, get  $r_{4maxsf4}$  and  $r_{4minsf4}$  corresponding to the value of  $\theta_3$ :

- (A) Effective  $\theta_3$  exists in the intersection (green sector).
- (B) With  $\theta_{3TN}$  as the boundary, the theoretical constraints on both sides are monotonically increasing space.

As the standard examples, Figures 10(b) and 10(c) will be described. Figures 10(a), and 10(d)–10(j) can be treated as the same way as examples.

When  $\theta_{3max} > \theta_{3maxT}$ ,  $\theta_{3maxT} \geq \theta_{3min} > \theta_{3minT}$  and  $\theta_{3minT} < \theta_{3TN} \leq \theta_{3min}$ . There is an intersection of the lower subinterval of  $\theta_3$  physical constraint and the higher subinterval of  $\theta_3$  theoretical constraint, as shown in Figure 10(b). There are

$$\begin{aligned} \theta_{3minTP} &= \theta_{3min}, \\ \theta_{3maxTP} &= \theta_{3maxT}. \end{aligned} \quad (16)$$

And when  $\theta_3 = \theta_{3minTP}$ , obtain  $r_{4minsf4}$ .

When  $\theta_3 = \theta_{3maxTP}$ , obtain  $r_{4maxsf4}$ .

When  $\theta_3 > \theta_{3maxT}$ ,  $\theta_{3maxT} \geq \theta_{3min} > \theta_{3minT}$  and  $\theta_{3min} < \theta_{3TN} < \theta_{3maxT}$ . There is an intersection of the lower

subinterval of  $\theta_3$  physical constraint and the higher subinterval of  $\theta_3$  theoretical constraint, as shown in Figure 10(c). There are

$$\begin{aligned} \theta_{3minTP} &= \theta_{3min}, \\ \theta_{3maxTP} &= \theta_{3maxT}. \end{aligned} \quad (17)$$

And when  $\theta_3 = \theta_{3TN}$ , obtain  $r_{4minsf4}$ . There is

$$\begin{aligned} r_{4maxsf4} &= \max(TRFour(W, S, \theta_{3minTP}), \\ &TRFour(W, S, \theta_{3maxTP})). \end{aligned} \quad (18)$$

In summary, on the basis of the boundary of the camera robot workspace,  $r_1$  comprehensive constraint is determined according to  $pT$  with  $WS$  characteristic as benchmark. After selecting  $r_1$ , the influencing factors of  $r_4$  are analyzed synthetically, and  $r_4$  comprehensive constraint is obtained. And this is also the conception of sequence of the two redundancy in GA.

### 3. Optimized Method SM

**3.1. Optimized Objective Function.** The optimization goal is to improve the motion stability of the end-effector; this paper sets the motor shaft to move as little as possible with the higher load inertia. The optimization objective function  $F$  is

$$F = \mathbf{W} \cdot \mathbf{X}, \quad (19)$$

where  $\mathbf{W}$  is the weight vector and  $\mathbf{X}$  represents the movement distance vector of axis.

**3.2. Algorithm Optimization.** According to the camera robot motion characteristics,  $r_1$  can be determined first and then the theoretical effect range of  $r_4$  can be calculated. This is the sequence of two variables. GM method used the dual redundancy as genes. By setting the initial population with average distribution in effect solution area, GM can eliminate the large iterative calculation from GA nonlinear constraints caused by sequence of dual redundancy and overcome the low effective proportion of stochastic initial population. Then through the pattern search algorithm, the composite algorithm can converge to the global optimal solution stably.

In the experiment, it is found that the optimal solution of GM did not change or change little in the process of iterative computation. That is, the analysis of camera robot motion characteristics with average initial population distribution greatly weakened the global optimization effect of GA in GM. So it is desirable to solve in a shorter time by motion characteristics with pattern search. This paper proposed SM method. To verify the effect of it, set the experiment.

The robot is in the initial standard zero state. 4 to 8 meters in the positive direction of the rail, set 45 target postures evenly. Specifically, five planes are set in the space, and 9 target postures are distributed in each plane, as shown in Figure 11, where red line represents the coordinate system  $z$ -axis and black line represents the coordinate system  $x$ -axis.

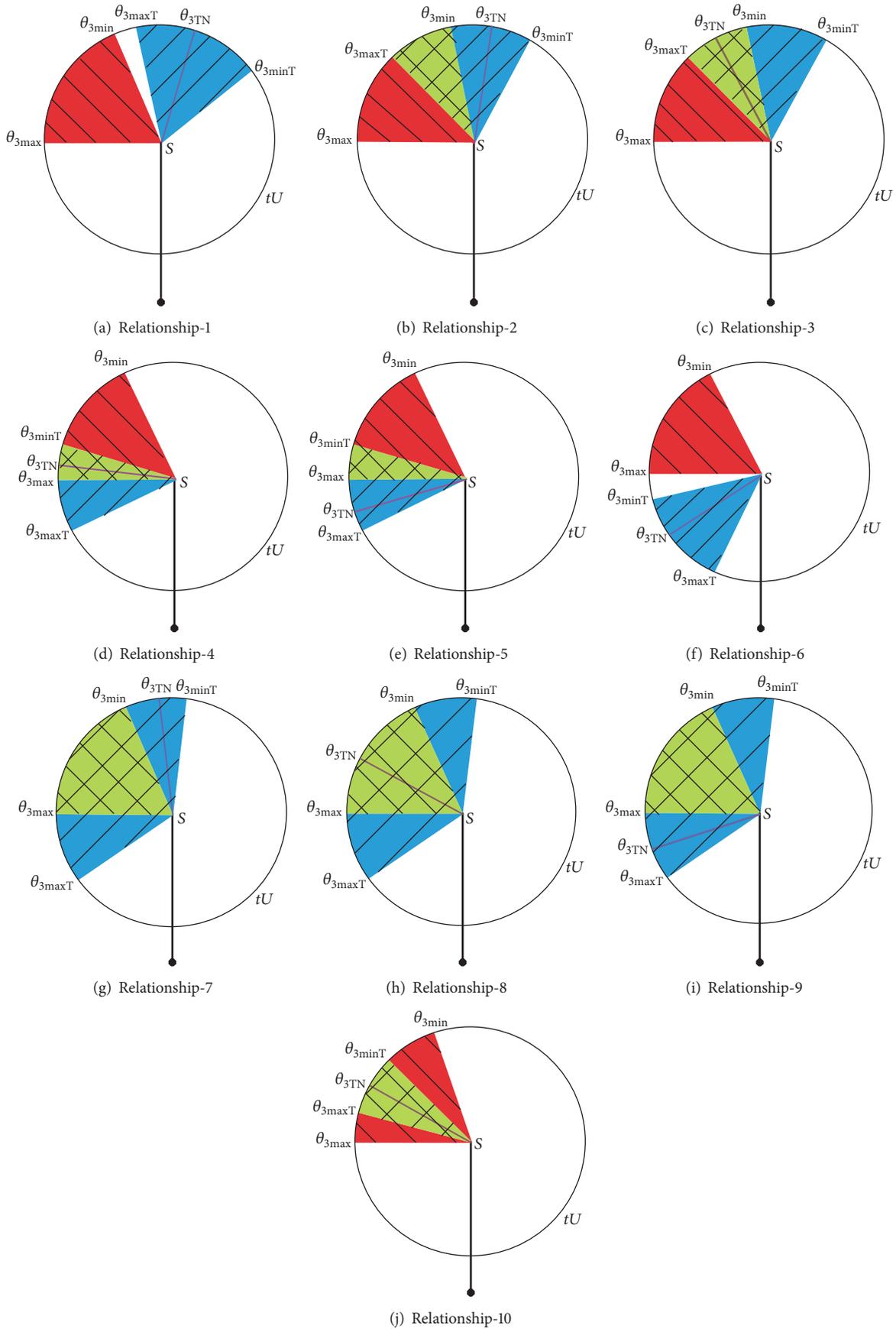


FIGURE 10:  $\theta_3$  comprehensive constraint and  $\theta_{3THN}$  relationship.

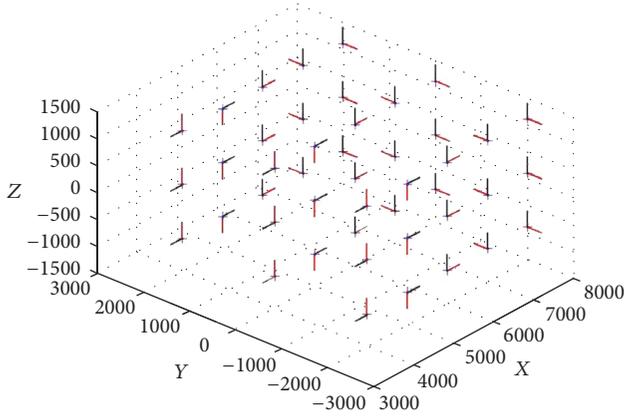


FIGURE 11: Target postures in workspace.

Assume that both  $vsGM$  and  $vsSM$  are 8-dimensional vectors and are solutions for each method, which can both achieve the target posture.

Set the physical constraints of each axis as

$$\mathbf{vAR} = [vAR_1 \ vAR_2 \ \dots \ vAR_8], \quad (20)$$

where

$$vAR_i = x_{j_{\max}} - x_{j_{\min}}, \quad (21)$$

$$i = 1, 2, \dots, 8, \quad x_j = r_1, \theta_2, \theta_3, r_4, \theta_5, \theta_6, \theta_7, \theta_{EE}.$$

Define

$$nDR = \prod_{i=1}^8 \left[ \frac{|vsGM_i - vsSM_i|}{vAR_i} \right]. \quad (22)$$

$nDR$  can be understood as the proportion of the generalized distance between different solutions in the entire workspace.

Take the experiment as follows.

Select 20 values in  $r_1$  comprehensive effective subinterval (specified  $pT$  as boundary, the robot at the same side with the current state). According to 20  $r_1$  values, the corresponding  $r_4$  comprehensive constraints are calculated. Select the middle value as  $r_4$  value, forming an individual with the corresponding  $r_1$ .

Calculate the optimal solutions by GM and SM methods, respectively. The result is shown in Figure 12. The  $x$ -axis is the target posture number, and  $y$ -axis is  $nDR$ . The solid point on the corresponding coordinate of the  $x$ -axis indicates that the GM gets a better solution, and the triangle represents that the SM gets a better solution. The cross symbol means that the two methods get the same solution.

For 45 target postures, GM costs 6328.32s and SM costs 3410.07s. There is only one posture with big distance appearing on the 30th posture, where the ratio of two values of optimal objective function is 12%, and GM method gets a better solution. The robot states are shown in Figure 13, the black body is the result of the GA, and the blue body is the result of the SM. The actual difference of two solutions is

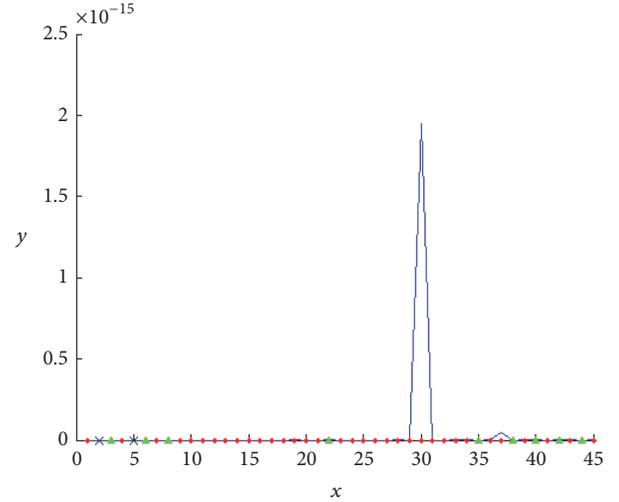


FIGURE 12: Compared result of two methods.

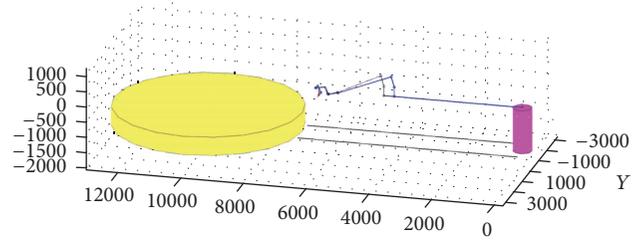


FIGURE 13: Maximum deviation of two methods.

not big. The solutions of two methods for the other 44 target postures are close enough to the global optimal solution (verified by a certain accuracy of the traversal algorithm).  $nDR$  is nearly to 0.

In summary, SM method is more convenient in practical application than GM.

## 4. Conclusion

In this paper, the motion characteristics analysis of PRRPR-S robot is discussed; GM method overcomes the defects of dual redundancy sequence and stochastic of GA. The experiment in Section 3 proved that GA in GM has little effect. SM gets good result directly and more quickly (about half the time consumed in GM).

Even though the kinematic solution method is for PRRPR-S robot, by using the idea of subworkspace and motion path, combining with pattern search, any other type of redundant robot can get the inverse kinematic solution. This is the significance of SM method.

## Conflicts of Interest

The author declares that there are no conflicts of interest related to this paper.

## References

- [1] J. He, S. Wang, and C. Wang, "Application analysis of camera robot in flim vitrual manufacture," *Advanced Motion Picture Technolgy*, vol. 10, pp. 44-45, 2017.
- [2] Z. Lu, *Theory and Application of Redundancy Robot*, China Machine Press, 2007.
- [3] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibodymechanisms," *IEEE Transactions on Systems Man & Cybernetics*, vol. 7, no. 12, pp. 868-871, 1977.
- [4] J. K. Parker, A. R. Khoogar, and D. E. Goldberg, "Inverse kinematics of redundant robots using genetic algorithms," in *Proceedings of the IEEE International Conference on Robotics and Automation, 1989*, vol. 1, pp. 271-276, Scottsdale, Ariz, USA, 2002.
- [5] Y. W. Sung, D. K. Cho, and M. J. Chung, "A constrained optimization approach to resolving manipulator redundancy," *Journal of Robotic Systems*, vol. 13, no. 5, pp. 275-288, 1996.
- [6] S. Masayuki, K. Hiromu, W. K. Yoon, K. Kosei, and K. Kazuhiro, "An analytical solution for the inverse kinematics of a redundant 7DoF Manipulator with link offsets," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1131-1142, 2008.
- [7] G. K. Singh and J. Claassens, "An analytical solution for the inverse kinematics of a redundant 7DoF manipulator with link offsets," in *Proceedings of the 23rd IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010*, pp. 2976-2982, Taipei, Taiwan, October 2010.
- [8] L. Zhang, J. He, and S. Wang, "Inverse Kinematic Solutions of Dual Redundant Camera Robot Based on Genetic Algorithm," *Mathematical Problems in Engineering*, vol. 2017, 17 pages, 2017.

## Research Article

# Generalized Beamforming Design for Cooperative MIMO Multirelay Networks with Infinite Constraints and Imperfect CSI

Haiyang Yu,<sup>1,2</sup> Wei Duan ,<sup>1</sup> Qiang Sun,<sup>1</sup> Xia Wang,<sup>1</sup> Jue Wang,<sup>1</sup> Jun Li,<sup>3</sup> and Jaeho Choi <sup>2</sup>

<sup>1</sup>School of Electronics and Information, Nantong University, Nantong 226019, China

<sup>2</sup>Department of Electronics Engineering, CAIT, Chonbuk National University, Chonju, Republic of Korea

<sup>3</sup>School of Mechanical and Electrical Engineering, Guangzhou University, Guangzhou, China

Correspondence should be addressed to Wei Duan; [sinder@ntu.edu.cn](mailto:sinder@ntu.edu.cn)

Received 29 September 2017; Accepted 17 January 2018; Published 26 March 2018

Academic Editor: Abhishek K. Gupta

Copyright © 2018 Haiyang Yu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A generalized base station-relay-user equipment (BS-Relay-UE) beamforming design is investigated for a cooperative multiple-input multiple-output (MIMO) multirelay networks with imperfect channel state information (CSI). In order to minimize the worst-case mean square error (MSE) which is subject to a semi-infinite (SI) relay power constraints, a generalized optimal beamforming structure for the relay amplifying matrix is effectively proposed, and then the SI relay power constraints are converted into linear matrix inequalities (LMIs) version. In such conversion, the objective problem recasts as a decoupled biconvex semidefinite programming (SDP) one which can be efficiently solved by the proposed alternating algorithm. The system performance has been verified in terms of worst-case MSE using a set of qualitative analyses. The results show us that the proposed beamforming method outperforms the conventional schemes and can also effectively reduce the computational complexity when it is compared to the cutting-set schemes and also to the nonrobust ones.

## 1. Introduction

Recently, cooperative multi-input multi-output (MIMO) relay network approaches are popularized to increase the system capacity and to improve the transmission reliability by leveraging spatial diversity. Since the channel estimation is important to the wireless communications, the authors in [1] investigated the main challenges faced by high-dimensional channel state information (CSI) acquisition in massive MIMO systems which was first shown that the narrow angular spread or the spatial sparsity is crucial for all low-rank approaches. Cooperative relay network technology has attracted significant interests due to the superior spectral efficiency. Various cooperative relaying schemes have been proposed, such as denoise-and-forward (DNF) [2–4], decode-and-forward (DF) [5, 6], and amplify-and-forward (AF) [7–12]. Specifically, the spatially correlated fading channels are considered in [13], which is more practical

and challenging. Training designs for estimation of spatially correlated multiple-input multiple-output (MIMO) AF two-way multirelay channels are studied in [13], where an optimal training structure is initially derived to minimize total mean square error (MSE) of the channel estimation.

Considering inaccurate channel estimation and feedback delay, the perfect channel state information (CSI), which are proposed in the above works, is usually hard to obtain in practice. In order to circumvent these problems, by taking into account the channel uncertainties, the imperfect CSI scenario has been studied in [7, 14–20]. The authors in [14] considered robust transceiver design in a multiuser MIMO multirelay cognitive radio network with interference power constraints and individual transmission power constraints. In [15], the authors have investigated the global optimal transceiver design in the MIMO link under a deterministic CSI uncertainty model. Specifically, a robust minimum mean-square-error (MMSE-) based beamforming scheme for

```

(1) Initialize:  $\xi = 10^{-3}$ ,  $N_{\max}$ ,  $\mathbf{B}^{(0)}$ ,  $\mathbf{D}^{(0)}$  set  $n = 0$ ;
(2) Repeat:
  (1) for  $n = 0$  to  $N_{\max}$  do
  (2) with fixed  $\mathbf{B}^{(n-1)}$ ,  $\mathbf{D}^{(n-1)}$  update  $\mathbf{W}_i^{(n)}$  via solving  $Q_3$ ;
  (3) for given  $\mathbf{W}_i^{(n)}$ ,  $\mathbf{B}^{(n-1)}$  update  $\mathbf{D}^{(n)}$  via solving  $Q_3$ ;
  (4) for given  $\mathbf{W}_i^{(n)}$  and  $\mathbf{D}^{(n)}$  update  $\mathbf{B}^{(n)}$  and  $\sum_{t=1}^2 \gamma_t^{(n)}$  via solving  $Q_3$ ;
  (5) if  $\sum_{t=1}^2 \gamma_t^{(n)} - \sum_{t=1}^2 \gamma_t^{(n-1)} \leq \varphi$ , then
      break;
  (6) end if
  (7) end for

```

ALGORITHM 1: The proposed generalized BS-R-UE robust beamforming scheme.

an amplify-and-forwarding multiantenna relay network is considered in [16]. Here, the authors divided the original problem into subproblems which are formulated into a convex optimization framework. By taking the imperfect channel state information into consideration, in [17], the robust AF-MIMO transceiver optimization is developed in order to combat correlated channel uncertainties, where the matrix-form conjugate gradient (MCG) method is used for optimization. Moreover, the robust joint optimization of the relay weights and the input covariance matrix of jamming signals for secrecy rate maximization is addressed in [18]. In [19], the authors investigated the worst-case robust relay precoder optimization for multiantenna AF relaying in the presence of either deterministic additive or multiplicative channel uncertainties. Specifically, the authors in [7, 20] have considered a two-way relay network consisting of multiple pairs of single-antenna users and distributed, multiple relays.

Furthermore, in [21–26], the generalization of S-Lemma to complex-valued variables and multiple semi-infinite (SI) constraints are presented. They have investigated iterative algorithms based on alternating convex search (ACS) to devise practical algorithms for solving the semi-infinite problems. In addition, in [26], the authors investigated a robust transceiver design for downlink multiuser MIMO AF-relay systems with norm-bounded channel uncertainties presenting that the cutting-set method proposed improves the performance of the worst-case MSE.

Since that, for the multiple relays scheme, not only the performance of the capacity outperforms that of the single relay one, but also the multiple relays scheme is more practical and challenging for the wireless communication scenarios. The downlink network considered here consists of a base station (BS), multiple relays, and user equipment (UE). We have named this particular AF system base BS-Relays-UE beamforming. The main contributions of this paper are summarized in the following:

- (i) We comprehensively investigate a generalized beamforming design for a cooperative MIMO multirelay networks with imperfect CSI. In the proposed scheme, the joint optimal design considers the precoding matrix at the BS, the beamforming matrix at the relays and the receiving beamforming matrix at

the UE with multiple relays, which is general and practical.

- (ii) Since the considered worst-case MSE optimization problem is not only nonconvex but also subject to the semi-infinite relay power constraints, a generalized optimal beamforming structure of the relay amplifying matrix is investigated. In addition, the SI relay power constraints are converted into linear matrix inequalities (LMIs). By this way, the objective problem recasts as a decoupled biconvex semidefinite programming (SDP) one, which can be efficiently solved by our proposed alternating algorithm.
- (iii) By means of the numerical results, the proposed beamforming design significantly reduces the computational cost and improves the performance in terms of the worst-case MSE compared with the non-robust and cutting-set cases.

In the following, a system model for the proposed cooperative MIMO multirelay networks is presented in Section 2 and a generalized BS-Relay-UE beamforming design is described in Section 3 (see Algorithm 1). In Section 4, the performance of the proposed system is evaluated using qualitative analysis and the numerical results are provided. Finally, the conclusions are made in Section 5.

*Notations.*  $\mathbf{A}^T$ ,  $\mathbf{A}^\dagger$ ,  $\text{tr}(\mathbf{A})$ ,  $\|\mathbf{A}\|_2$ , and  $\|\mathbf{A}\|_F$  denote the transpose, conjugate transpose, trace, Euclidean norm, and Frobenius norm of a matrix  $\mathbf{A}$ , respectively.  $\Re(\cdot)$  and  $\Im(\cdot)$  are operators for taking the real part and imaginary part of a complex-valued matrix, respectively. The field of  $m \times n$ -dimensional complex-valued matrix is denoted by  $\mathbb{C}^{m \times n}$ .  $I_N$  is an  $N \times N$  identity matrix.  $\odot$  and  $\otimes$  are the operators for computing Hadamard and Kronecker products.

## 2. System Model

In this section, an amplify-and-forward relaying system model is described. In the proposed system, users are away from a base station and they are isolated and out of the BS coverage. In other words, the direct link between BS and UE does not exist so that UE can get a connection to the BS only via relays in order to achieve any signal receptions. To be more

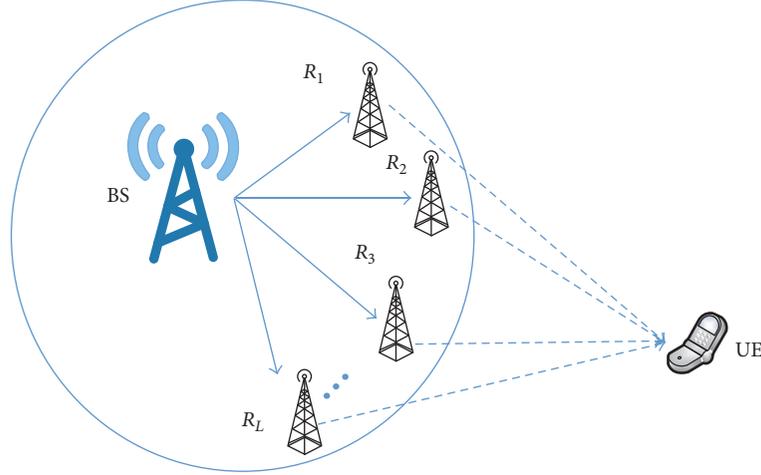


FIGURE 1: Cooperative MIMO multirelay networks.

specific, the proposed multirelay network consists of one BS,  $L$  relays, and one UE as shown in Figure 1. In this network, communications are performed using a half-duplex mode. It is further assumed that the BS and UE are equipped with  $M$  antennas while each relay node is with  $N$  antennas.

Now, concerning the communication channels related to the proposed network, there are two of them. The one is the group of estimated communication channels  $\mathbf{G}_i \in \mathbb{C}^{N \times M}$ , which are from the BS to  $L$  relays  $R_i$ , for  $i \in \{1, \dots, L\}$ ; the other one is another group of communication channels  $\mathbf{F}_i \in \mathbb{C}^{M \times N}$ , which are from  $L$  relays  $R_i$  to UE. We have also considered  $N \geq M$  to satisfy the decodeable condition.

It is assumed that the BS transmits the signals to the UE using two consecutive time slots. At the first time slot, after being linearly processed by the matrix  $\mathbf{B} \in \mathbb{C}^{M \times M}$ , which is constrained by  $\text{tr}\{\mathbf{B}\mathbf{B}^H\} \leq P_t$ , the information data  $\mathbf{x} = [x_1, \dots, x_m]^T$  with unit power is transmitted to the relay nodes. Therefore, the received signal at  $R_i$  can be expressed as

$$\mathbf{y}_{R_i} = \mathbf{G}_i \mathbf{B} \mathbf{x} + \mathbf{n}_{R_i}, \quad (1)$$

where  $\mathbf{n}_{R_i} \sim \mathcal{CN}(0, \sigma_{R_i}^2 \mathbf{I}_N)$  represents the additive white Gaussian noise (AWGN) vector with zero mean and variance  $\sigma_{R_i}^2$  at the relay node  $R_i$ .

At the second time slot, the relay node  $R_i$  linearly amplifies  $\mathbf{y}_{R_i}$  with an  $N \times N$  beamforming matrix  $\mathbf{W}_i$  and then broadcasts the amplified signal vector  $\mathbf{x}_{R_i}$  to the UE, which results in

$$\mathbf{x}_{R_i} = \mathbf{W}_i \mathbf{y}_{R_i}. \quad (2)$$

Therefore, the transmission power used by the relay node  $R_i$  is given by

$$\begin{aligned} \mathbb{E} \left\{ \|\mathbf{x}_{R_i}\|_2^2 \right\} &= \text{tr} \left\{ \mathbf{W}_i \mathbf{y}_{R_i} \mathbf{y}_{R_i}^\dagger \mathbf{W}_i^\dagger \right\} \\ &= \text{tr} \left\{ \mathbf{W}_i \left( \mathbf{G}_i \mathbf{B} \mathbf{B}^\dagger \mathbf{G}_i^\dagger + \sigma_{R_i}^2 \mathbf{I}_N \right) \mathbf{W}_i^\dagger \right\}. \end{aligned} \quad (3)$$

At the UE, the received data is linearly utilized by using the receiving beamforming matrix  $\mathbf{D} \in \mathbb{C}^{M \times M}$  with  $\text{tr}\{\mathbf{D}\mathbf{D}^\dagger\} \leq$

$P_d$ , where  $\leq P_d$  is a slack value. The received signal vectors at the UE can be finally denoted as

$$\mathbf{y} = \mathbf{D} \left( \sum_{i=1}^L \mathbf{F}_i \mathbf{W}_i \mathbf{G}_i \mathbf{B} \mathbf{x} + \sum_{i=1}^L \mathbf{F}_i \mathbf{W}_i \mathbf{n}_{R_i} + \mathbf{n}_D \right), \quad (4)$$

where  $\mathbf{n}_D$  is the received noise vector at the UE with mean zero and variance  $\sigma_D^2$ .

By taking into account the estimation error, the CSI is considered partially known at each node. With this consideration, the actual channel coefficients of the links follow that

$$\begin{aligned} \mathbf{G}_i &\triangleq \{ \tilde{\mathbf{G}}_i + \Delta_{G_i} \}, \\ \mathbf{F}_i &\triangleq \{ \tilde{\mathbf{F}}_i + \Delta_{F_i} \} \end{aligned} \quad (5)$$

with  $\{\Delta_{G_i}, \Delta_{F_i}\}$  as the channel uncertainties. For simplicity, the channel uncertainties are assumed to be norm-bounded errors (NBEs) [24] as

$$\begin{aligned} \|\Delta_{G_i}\| &\leq \alpha_i, \\ \|\Delta_{F_i}\| &\leq \beta_i, \end{aligned} \quad (6)$$

where  $0 \leq \{\alpha_i, \beta_i\} \ll 1$ .

With this observation, the worst-case MSE at the UE node can be obtained as

$$\begin{aligned} \varphi &\triangleq \left\| \sum_{i=1}^L \mathbf{D} (\tilde{\mathbf{F}}_i + \Delta_{F_i}) \mathbf{W}_i (\tilde{\mathbf{G}}_i + \Delta_{G_i}) \mathbf{B} - \mathbf{I}_M \right\|_F^2 \\ &\quad + \sigma_{R_i}^2 \left\| \sum_{i=1}^L \mathbf{D} (\tilde{\mathbf{F}}_i + \Delta_{F_i}) \mathbf{W}_i \right\|_F^2 + \sigma_D^2 \|\mathbf{D}\|_F^2 \\ &\leq \underbrace{\left\| \sum_{i=1}^L \mathbf{D} (\tilde{\mathbf{F}}_i + \Delta_{F_i}) \mathbf{W}_i (\tilde{\mathbf{G}}_i + \Delta_{G_i}) \mathbf{B} - \mathbf{I}_M \right\|_F^2}_{\mathcal{M}_1} \end{aligned}$$

$$\begin{aligned}
& + \sigma_{R_i}^2 \underbrace{\left\| \sum_{i=1}^L \mathbf{D} (\tilde{\mathbf{F}}_i + \Delta_{\mathbf{F}_i}) \mathbf{W}_i \right\|_F^2}_{\mathcal{M}_2} + \sigma_D^2 P_d \\
& = \mathcal{M}_1 + \mathcal{M}_2 + \sigma_D^2 P_d.
\end{aligned} \tag{7}$$

Since  $\mathcal{M}_1 + \mathcal{M}_2 \geq \varphi - \sigma_D^2 P_d$ , after introducing the slack value  $\hat{\varphi} = \varphi - \sigma_D^2 P_d$ , the objective problem is to minimize the worst-case MSE subjects to the BS and the relay transmission power constraints which can be formulated as follows:

$$\begin{aligned}
Q_1: \quad & \min_{\mathbf{B}, \mathbf{W}_i, \mathbf{D}} \quad \mathcal{M}_1 + \mathcal{M}_2 \\
& \text{s.t.} \quad \|\mathbf{x}_{R_i}\|^2 \leq P_{R_i}, \\
& \quad \text{tr}\{\mathbf{B}\mathbf{B}^\dagger\} \leq P_t, \\
& \quad \text{tr}\{\mathbf{D}\mathbf{D}^\dagger\} \leq P_d, \\
& \quad \hat{\varphi} \leq \mathcal{M}_1 + \mathcal{M}_2,
\end{aligned} \tag{8}$$

where  $P_{R_i}$  denotes the power constraint at the relay node. In problem  $Q_1$ , it is clear that not only the objective problem  $\mathcal{M}_1 + \mathcal{M}_2$  is nonconvex, but also the semi-infinite expressions of the optimal  $\mathbf{W}_i$ ,  $\mathbf{D}$ , and  $\mathbf{B}$  are intractable. In particular, the globally optimal solution is difficult to be obtained; in this paper, to efficiently solve  $Q_1$ , a biconvex SDP is proposed to obtain the suboptimal solution of the worst-case MSE for our proposed scheme.

$$\mathbf{F}_i^* \mathbf{W}_i \mathbf{G}_i^* = \mathbf{\Pi}_i \left[ \begin{bmatrix} \mathbf{F}_i^\diamond \end{bmatrix}_{M \times M} \quad \mathbf{0}_{M \times (N-M)} \right] \left[ \begin{bmatrix} \mathbf{W}_i^\diamond \end{bmatrix}_{M \times M} \quad \mathbf{X}_{M \times (N-M)} \right] \left[ \begin{bmatrix} \mathbf{G}_i^\diamond \end{bmatrix}_{M \times M} \quad \mathbf{0}_{(N-M) \times M} \right] \mathbf{\Omega}_i^\dagger = \mathbf{\Pi}_i \mathbf{F}_i^\diamond \mathbf{W}_i^\diamond \mathbf{G}_i^\diamond \mathbf{\Omega}_i^\dagger. \tag{12}$$

In this manner, we have

$$\begin{aligned}
& \mathcal{M}_1 + \mathcal{M}_2 \\
& = \left\| \sum_{i=1}^L \left( \mathbf{\Pi}_i \mathbf{F}_i^\diamond \mathbf{W}_i^\diamond \mathbf{G}_i^\diamond \mathbf{\Omega}_i^\dagger + \mathbf{\Phi}_i \right) - \mathbf{I}_M \right\|_F^2 \\
& \quad + \sigma_{R_i}^2 \left\| \sum_{i=1}^L \left( \mathbf{\Pi}_i \left[ \mathbf{F}_i^\diamond \mathbf{W}_i^\diamond \quad \mathbf{F}_i^\diamond \mathbf{X} \right] \mathbf{U}_i^\dagger + \mathbf{\Psi}_i \right) \right\|_F^2,
\end{aligned} \tag{13}$$

where  $\mathbf{\Psi}_i = \mathbf{D} \Delta_{\mathbf{F}_i} \mathbf{V}_i \mathbf{C}_i \mathbf{U}_i^\dagger$  and  $\mathbf{\Phi}_i = \mathbf{D} \Delta_{\mathbf{F}_i} \mathbf{V}_i \mathbf{C}_i \mathbf{F}_i + \mathbf{\Sigma}_i \mathbf{C}_i \mathbf{U}_i^\dagger \Delta_{\mathbf{G}_i} \mathbf{B}$ . It is worth noting that the term  $\mathbf{D} \Delta_{\mathbf{F}_i} \mathbf{V}_i \mathbf{C}_i \mathbf{U}_i^\dagger \Delta_{\mathbf{G}_i} \mathbf{B}$  becomes 0, because if we retain this term in  $\mathbf{\Phi}_i$ , when calculating the MSE, it will result in some terms involving 3rd and 4th order of channel uncertainties which are negligible.

### 3. Generalized BS-Relay-UE Beamforming Design

*3.1. The Optimal Relay Beamforming Design.* With the fixed BS and UE beamforming matrices  $\mathbf{B}$  and  $\mathbf{D}$ , suppose the singular value decomposition (SVD) of  $\mathbf{F}_i^* = \mathbf{D}\tilde{\mathbf{F}}_i$  and  $\mathbf{G}_i^* = \tilde{\mathbf{G}}_i \mathbf{B}$  as follows:

$$\begin{aligned}
\mathbf{F}_i^* &= \mathbf{\Pi}_i \left[ \begin{bmatrix} \mathbf{F}_i^\diamond \end{bmatrix}_{M \times M} \quad \mathbf{0}_{M \times (N-M)} \right] \mathbf{V}_i^\dagger \triangleq \mathbf{\Pi}_i \mathbf{\Sigma}_i \mathbf{V}_i^\dagger, \\
\mathbf{G}_i^* &= \mathbf{U}_i \left[ \begin{bmatrix} \mathbf{G}_i^\diamond \end{bmatrix}_{M \times M} \right] \mathbf{\Omega}_i^\dagger \triangleq \mathbf{U}_i \mathbf{F}_i \mathbf{\Omega}_i^\dagger,
\end{aligned} \tag{9}$$

where  $\mathbf{V}_i \in \mathbb{C}^{N \times N}$ ,  $\mathbf{U}_i \in \mathbb{C}^{N \times N}$ ,  $\mathbf{\Pi}_i \in \mathbb{C}^{M \times M}$ , and  $\mathbf{\Omega}_i \in \mathbb{C}^{M \times M}$  are unitary matrices.

**Theorem 1.** *Using the SVDs in (9), the optimal relay beamforming matrix as the solution to the objective problem  $Q_1$  can be obtained as*

$$\mathbf{W}_i = \mathbf{V}_i \mathbf{C}_i \mathbf{U}_i^\dagger, \tag{10}$$

where  $\mathbf{C}_i \in \mathbb{C}^{N \times N}$  is a matrix to be determined.

*Proof.* The proof is similar to [8, 25].

Without loss of generality,  $\mathbf{C}_i$  can be further partitioned as follows:

$$\mathbf{C}_i = \left[ \begin{array}{cc} \left[ \mathbf{W}_i^\diamond \right]_{M \times M} & \mathbf{X}_{M \times (N-M)} \\ \mathbf{Y}_{(N-M) \times M} & \mathbf{Z}_{(N-M) \times (N-M)} \end{array} \right]. \tag{11}$$

From (9), (11) and Theorem 1, one can have

Using  $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$  and  $\|\mathbf{A}\mathbf{B}\| \leq \|\mathbf{A}\| \|\mathbf{B}\|$ , from (13), we have

$$\begin{aligned}
\mathcal{M}_1 &\leq \left\| \sum_{i=1}^L \mathbf{\Pi}_i \mathbf{F}_i^\diamond \mathbf{W}_i^\diamond \mathbf{G}_i^\diamond \mathbf{\Omega}_i^\dagger - \mathbf{I}_M \right\|_F^2 + \sum_{i=1}^L \|\mathbf{\Phi}_i\|_F^2 \\
&\leq \sum_{i=1}^L \left( P_d P_t \alpha_i \beta_i + P_d \beta_i \|\mathbf{G}_i^\diamond\|_F^2 + P_t \alpha_i \|\mathbf{F}_i^\diamond\|_F^2 \right) \\
&\quad \cdot \left\| \left[ \begin{array}{cc} \left[ \mathbf{W}_i^\diamond \right]_{M \times M} & \mathbf{X}_{M \times (N-M)} \\ \mathbf{Y}_{(N-M) \times M} & \mathbf{Z}_{(N-M) \times (N-M)} \end{array} \right] \right\|_F^2 \\
&\quad + \left\| \sum_{i=1}^L \mathbf{F}_i^\diamond \mathbf{W}_i^\diamond \mathbf{G}_i^\diamond - \mathbf{\Pi}_i^\dagger \mathbf{\Omega}_i \right\|_F^2, \\
\mathcal{M}_2 &\leq \sigma_{R_i}^2 \left\| \sum_{i=1}^L \mathbf{\Pi}_i \left[ \mathbf{F}_i^\diamond \mathbf{W}_i^\diamond \quad \mathbf{F}_i^\diamond \mathbf{X} \right] \right\|_F^2 + \sigma_{R_i}^2 \sum_{i=1}^L \|\mathbf{\Psi}_i\|_F^2
\end{aligned}$$

$$\leq \sigma_{R_t}^2 \left\| \sum_{i=1}^L \left[ \mathbf{F}_i^\diamond \mathbf{W}_i^\diamond \quad \mathbf{F}_i^\diamond \mathbf{X} \right] \right\|_F^2 + \sigma_{R_t}^2 P_d \sum_{i=1}^L \beta_i \cdot \left\| \begin{bmatrix} \left[ \mathbf{W}_i^\diamond \right]_{M \times M} & \mathbf{X}_{M \times (N-M)} \\ \mathbf{Y}_{(N-M) \times M} & \mathbf{Z}_{(N-M) \times (N-M)} \end{bmatrix} \right\|_F^2. \quad (14)$$

Since the channel uncertainty  $\Delta_{\mathbf{G}_i}$  can be further partitioned,

$$\Delta_{\mathbf{G}_i} = \begin{bmatrix} \left[ \Delta_{\mathbf{G}_i}^\diamond \right]_{M \times M} \\ \left[ \Delta_{\mathbf{G}_i}^b \right]_{M \times (N-M)} \end{bmatrix}, \quad (15)$$

upon substituting (10), (11), and (15) into the relay power constraint (3), we have

$$\begin{aligned} & \mathbf{E} \left\{ \left\| \mathbf{x}_{R_t} \right\|_2^2 \right\} \\ &= \text{tr} \left\{ \mathbf{W}_i^\diamond \left( \mathbf{G}_i^\sharp \mathbf{B} \left( \mathbf{G}_i^\sharp \mathbf{B} \right)^\dagger + \sigma_{R_t}^2 \mathbf{I}_M \right) \left( \mathbf{W}_i^\diamond \right)^\dagger + \mathbf{X} \mathbf{X}^\dagger \right\} \\ & \quad + \text{tr} \left\{ \mathbf{Z} \mathbf{Z}^\dagger + \mathbf{Y} \left( \mathbf{G}_i^\sharp \mathbf{B} \left( \mathbf{G}_i^\sharp \mathbf{B} \right)^\dagger + \sigma_{R_t}^2 \mathbf{I}_M \right) \mathbf{Y}^\dagger \right\}, \end{aligned} \quad (16)$$

where  $\mathbf{G}_i^\sharp = \mathbf{G}_i^\diamond + \Delta_{\mathbf{G}_i}^\diamond$ . Synthesizing (14) and (16), it is clear that, for any feasible  $\mathbf{C}_i$  with  $\{\mathbf{X}, \mathbf{Y}, \mathbf{Z}\} \neq \mathbf{0}$ , one can always find

$$\mathbf{C}'_i = \begin{bmatrix} \left[ \mathbf{W}_i^\diamond \right]_{M \times M} & \mathbf{0}_{M \times (N-M)} \\ \mathbf{0}_{(N-M) \times M} & \mathbf{0}_{(N-M) \times (N-M)} \end{bmatrix}, \quad (17)$$

which can achieve the smaller relay power constraint and MSE. Thus, we conclude that if and only if  $\mathbf{X} = \mathbf{Y} = \mathbf{Z} = \mathbf{0}$ , the objective problem  $\{\min \mathcal{M}_1 + \mathcal{M}_2\}$  has the optimal solution. By this way, the optimal expression of  $\mathbf{W}_i$  can be denoted as

$$\mathbf{W}_i^\sharp = \mathbf{V}_i \begin{bmatrix} \left[ \mathbf{W}_i^\diamond \right]_{M \times M} & \mathbf{0}_{M \times (N-M)} \\ \mathbf{0}_{(N-M) \times M} & \mathbf{0}_{(N-M) \times (N-M)} \end{bmatrix} \mathbf{U}_i^\dagger. \quad (18)$$

□

**3.2. The Joint Source and Relay Beamforming Design.** From (14), it is clear that, with fixed  $\mathbf{W}_i^\diamond$ , the objective problem  $\{\min(\mathcal{M}_1 + \mathcal{M}_2)\}$  is equivalent to the problem:

$$\min \left\| \sum_{i=1}^L \Phi_i \right\|_F^2 + \left\| \sum_{i=1}^L \Psi_i \right\|_F^2. \quad (19)$$

Therefore, similar to [22], by introducing the auxiliary optimization variable  $\gamma_t$ , for  $t = 1, 2$ , problem (19) can be recast in the epigraph form [27] as  $\{\min(\gamma_1 + \gamma_2)\}$  which subjects to  $\left\| \sum_{i=1}^L \Phi_i \right\|_F^2 \leq \gamma_1$ ,  $\left\| \sum_{i=1}^L \Psi_i \right\|_F^2 \leq \gamma_2$ . In this manner, the

objective problem  $Q_1$  can be equivalently converted into the problem  $Q_2$  as follows:

$$\begin{aligned} Q_2: \min_{\mathbf{B}, \mathbf{D}} \quad & \gamma_1 + \gamma_2 \\ \text{s.t.} \quad & \left\| \mathbf{x}_{R_t} \right\|^2 \leq P_{R_t}, \\ & \text{tr} \{ \mathbf{B} \mathbf{B}^\dagger \} \leq P_t, \\ & \text{tr} \{ \mathbf{D} \mathbf{D}^\dagger \} \leq P_d, \\ & \left\| \sum_{i=1}^L \Phi_i \right\|_F^2 \leq \gamma_1, \\ & \left\| \sum_{i=1}^L \Psi_i \right\|_F^2 \leq \gamma_2. \end{aligned} \quad (20)$$

Using the identities  $\|\mathbf{A}\| = \|\text{vec}[\mathbf{A}]\|$  and  $\text{vec}[\mathbf{A}\mathbf{B}\mathbf{C}] = (\mathbf{C}^T \otimes \mathbf{A})\text{vec}[\mathbf{B}]$ , for any given matrix  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$ , for the constraint  $\left\| \sum_{i=1}^L \Phi_i \right\|_F^2 \leq \gamma_1$ , we have

$$\begin{aligned} \left\| \sum_{i=1}^L \Phi_i \right\|_F^2 &= \left\| \sum_{i=1}^L \text{vec}[\Phi_i] \right\|_F^2 \\ &= \left\| \sum_{i=1}^L \left( (\mathbf{V}_i \mathbf{C}_i \Gamma_i)^T \otimes \mathbf{D} \right) \text{vec}[\Delta_{\mathbf{F}_i}] + \sum_{i=1}^L \mathbf{B}^T \right. \\ & \quad \left. \otimes (\Sigma_i \mathbf{C}_i \mathbf{U}_i^\dagger) \text{vec}[\Delta_{\mathbf{G}_i}] \right\|_F^2. \end{aligned} \quad (21)$$

Further assuming  $\sum_{i=1}^L \text{vec}[\Phi_i] = \mathcal{N}_i$ , the constraint  $\left\| \sum_{i=1}^L \Phi_i \right\|_F^2 \leq \gamma_1$  can be represented in the term of the following LMI form as

$$\begin{bmatrix} \gamma_1 & \mathcal{N}_i^\dagger \\ \mathcal{N}_i & \mathbf{I} \end{bmatrix} \geq 0. \quad (22)$$

By employing S-Lemma [22], (22) can be recast as

$$\begin{bmatrix} \gamma_1 & \mathbf{0}_{1 \times MN} & \mathbf{0}_{1 \times MN} & \cdots & \mathbf{0}_{1 \times MN} \\ \mathbf{0}_{MN \times 1} & \mathbf{I}_{MN} & -\Xi_1^\dagger & \cdots & -\Xi_{2L}^\dagger \\ \mathbf{0}_{MN \times 1} & -\Xi_1 & \epsilon_1 \mathbf{I}_{MN} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{MN \times 1} & -\Xi_{2L} & \mathbf{0} & \cdots & \epsilon_{2L} \mathbf{I}_{MN} \end{bmatrix} \geq 0, \quad (23)$$

where  $\epsilon_i$ , for  $i \in \{1, \dots, 2L\}$ , is a slack variable, and

$$\Xi_j = \begin{cases} \left[ \alpha_j \mathbf{M}_{G_j}, \mathbf{0}_{MN \times M(N-M)} \right], & j = 1, \dots, L, \\ \left[ \beta_j \mathbf{M}_{F_j} \right], & j = L+1, \dots, 2L, \end{cases} \quad (24)$$

where

$$\begin{aligned} \mathbf{M}_{F_i} &= (\mathbf{V}_i \mathbf{C}_i \mathbf{\Gamma}_i)^T \otimes \mathbf{D}, \\ \mathbf{M}_{G_i} &= \mathbf{B}^T \otimes (\mathbf{\Sigma}_i \mathbf{C}_i \mathbf{U}_i^\dagger). \end{aligned} \quad (25)$$

On the other hand, similarly, letting  $\mathbf{\Theta}_{F_i} = (\mathbf{V}_i \mathbf{C}_i \mathbf{U}_i^\dagger)^T \otimes \mathbf{D}$ , the constraint  $\|\sum_{i=1}^L \mathbf{\Psi}_i\|_F^2$  can be equivalently converted into the LMI version as follows:

$$\begin{bmatrix} \gamma_2 & \mathbf{0}_{1 \times MN} & \mathbf{0}_{1 \times MN} & \cdots & \mathbf{0}_{1 \times MN} \\ \mathbf{0}_{MN \times 1} & \mathbf{I}_{MN} & -\mathbf{\Theta}_{F_1}^\dagger & \cdots & -\mathbf{\Theta}_{F_L}^\dagger \\ \mathbf{0}_{MN \times 1} & -\mathbf{\Theta}_{F_i} & \varsigma_1 \mathbf{I}_{MN} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{MN \times 1} & -\mathbf{\Theta}_{F_L} & \mathbf{0} & \cdots & \varsigma_L \mathbf{I}_{MN} \end{bmatrix} \succeq 0, \quad (26)$$

where  $\varsigma_i$  is a slack variable. In this observation, (23) and (26) represented the equivalent SDP formulation of the constraints  $\|\sum_{i=1}^L \mathbf{\Phi}_i\|_F^2 \leq \gamma_1$  and  $\|\sum_{i=1}^L \mathbf{\Psi}_i\|_F^2$  which are affine in the channel uncertainties.

For the relay power constraint, after introducing the slack variable  $P^\diamond$ , it can be decoupled into two parts as follows:

$$\overline{P^\diamond} + \overline{\sigma}_{R_i}^2 \|\mathbf{W}_i^\diamond\|_F^2 \leq \overline{P}_{R_i}, \quad (27)$$

$$\|\mathbf{W}_i^\diamond \mathbf{G}_i^\# \|_F^2 \leq \frac{P^\diamond}{P_t}, \quad (28)$$

where  $\overline{P^\diamond} = P^\diamond/P_t$ ,  $\overline{\sigma}_{R_i}^2 = \sigma_{R_i}^2/P_t$ , and  $\overline{P}_{R_i} = P_{R_i}/P_t$ . Defining

$$c_i = \text{vec}(\mathbf{G}_i^\diamond)^\dagger \widetilde{\mathbf{W}}_i \text{vec}(\mathbf{G}_i^\diamond) - \overline{P^\diamond}, \quad (29)$$

$$\widetilde{\mathbf{W}}_i = \left( (\mathbf{W}_i^\diamond)^\dagger \mathbf{W}_i^\diamond \right) \odot \mathbf{I},$$

the constraint (28) can be reformulated as

$$\begin{aligned} & \text{vec}(\Delta_{G_i}^\dagger) \widetilde{\mathbf{W}}_i \text{vec}(\Delta_{G_i}) \\ & + \mathfrak{R} \left\{ \text{vec}(\Delta_{G_i}^\dagger) \widetilde{\mathbf{W}}_i \text{vec}(\mathbf{G}_i^\diamond) \right\} + c_i \leq 0, \end{aligned} \quad (30)$$

which can be converted to the following LMI:

$$\begin{bmatrix} \zeta_i \mathbf{I}_N - \widetilde{\mathbf{W}}_i & -\widetilde{\mathbf{W}}_i \text{vec}(\mathbf{G}_i^\diamond) \\ -\text{vec}(\mathbf{G}_i^\diamond)^\dagger \widetilde{\mathbf{W}}_i^\dagger & c_i - \zeta_i \alpha_i \end{bmatrix} \succeq 0, \quad (31)$$

where  $\zeta_i$  is a slack variable.

Thus, the objective problem  $Q_2$  is recast as

$$\begin{aligned} Q_3: \quad & \min_{\mathbf{B}, \mathbf{D}} \quad \gamma_1 + \gamma_2 \\ \text{s.t.} \quad & \text{tr} \{ \mathbf{B} \mathbf{B}^\dagger \} \leq P_t, \\ & \text{tr} \{ \mathbf{D} \mathbf{D}^\dagger \} \leq P_d, \end{aligned} \quad (32)$$

(23), (23), (31) hold,  $\forall \zeta_i, \forall \epsilon_i, \forall \varsigma_i$ .

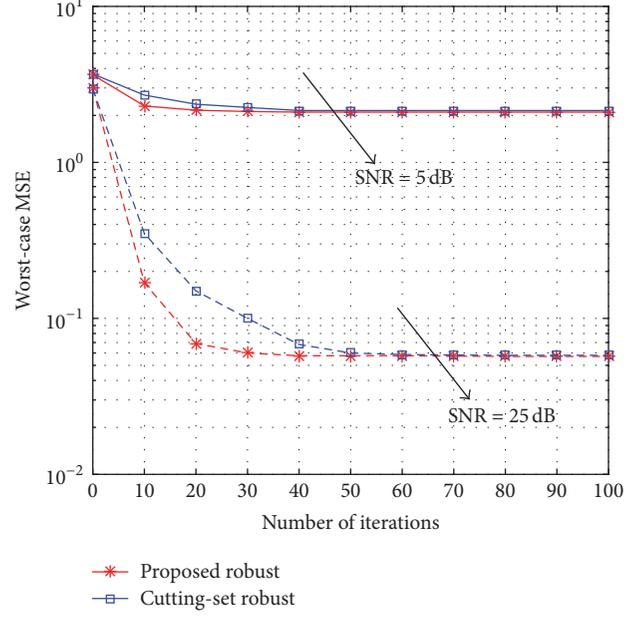


FIGURE 2: The worst-case MSE versus the number of iterations.

It is clear that the problem  $Q_3$  is a biconvex SDP which can be efficiently solved by the following alternating algorithm by using CVX [27].

By initializing the small  $\varphi$  and setting the limitation of the number of iterations  $N_{\max}$ , the suboptimal solution can be obtained when  $\sum_{t=1}^2 \gamma_t^{(n)} - \sum_{t=1}^2 \gamma_t^{(n-1)} \leq \varphi$ .

#### 4. Numerical Results

In this section, we examine the performance of the proposed robust scheme in terms of the worst-case MSE compared with cutting-set method in [26], the nonrobust case and the perfect design. All results are averaged over 5,000 channel realizations and the initial BS and the UE beamforming matrices are chosen at random. We set the number of the relay nodes  $L = 3$ , the number of antennas  $M = 4$  and  $N = 5$ , respectively. The transmission power at the BS and the relay node is  $P_t = P_r = 1$ . In the iterative algorithm,  $\varphi$  and  $N_{\max}$  are set to  $10^{-3}$  and 100, respectively.

Figure 2 depicts the convergence of the proposed algorithm with the fixed NBEs as  $\alpha_i = \beta_i = 0.01$  for the transmission SNR = 5 dB and SNR = 25 dB. Clearly, our proposed scheme has satisfactory convergent performance compared with the cutting-set method. It is because, in our proposed scheme, the objective problem converts the worst-case MSE into the one with  $\gamma_1 + \gamma_2$  which significantly reduces the computational complexity. In addition, the optimal (local) relay beamforming matrix can be efficiently derived which supports the practical utility of our design. It is also observed that, for the higher transmission SNR, the convergence rate of our proposed scheme outperforms the cutting-set one while both these two schemes provide the optimal (local) solution with more iterations.

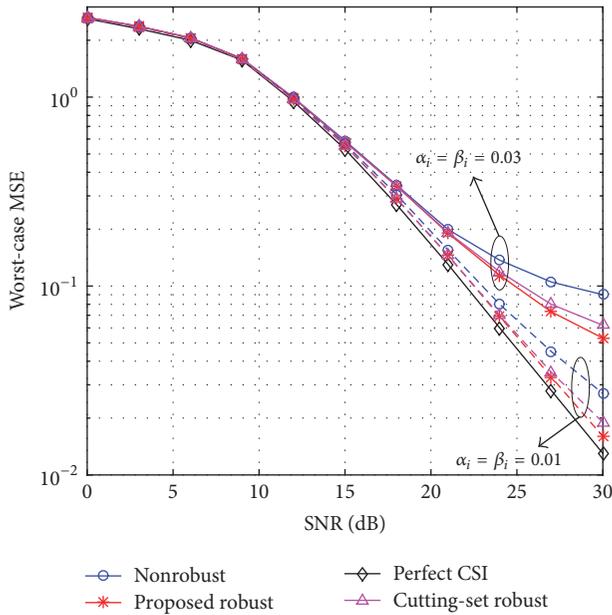


FIGURE 3: The worst-case MSE versus the transmission SNR.

Figure 3 compares the worst-case MSE in cases of our proposed scheme, the cutting-set one and the perfect one versus the transmission SNR with  $\alpha_i = \beta_i = 0.01$  and  $\alpha_i = \beta_i = 0.03$ , where the perfect one is served as the upper bound with the NBEs as  $\alpha_i = \beta_i = 0$  while the robust solutions are obtained by using the corresponding algorithms. Results reveal that our proposal has a relative advantage over the other two schemes, especially for the larger NBE case. This is reasonable, since employing the approximation may lose the performance, in our proposed scheme, the objective problem  $Q_3$  is equivalent to the original target one without using the approximations (except the terms involving the high-order channel uncertainties) which is different to the work in [26].

## 5. Conclusions

So far, we have presented an efficient way to achieve signal transmission from a BS to an isolated, out-of-coverage user by realizing a generalized BS-R-UE beamforming. In our proposed cooperative MIMO multirelay networks with imperfect CSI; first, the optimal relay beamforming is derived as a means to improve the efficiency. Then, the semi-infinite objective problem is converted into a biconvex problem, which is subject to the LMI constraints. Furthermore, the converted, biconvex problem is efficiently solved by our alternating algorithm. The system performance in terms of worst-case MSE is verified using a set of qualitative analysis; the results show us that the proposed beamforming method outperforms the conventional schemes and can also effectively reduce the computational complexity when it is compared to the cutting-set schemes and also to the nonrobust ones. Our future concerns will be the generalized optimization problems on the two-way relay networks and cooperative relaying nonorthogonal multiple access (NOMA) systems.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The authors would gratefully acknowledge the grants from the National Natural Science Foundation of China (61371113, 61401241, 61401240, and 61771264), Nantong University-Nantong Joint Research Center for Intelligent Information Technology (KFKT2016B01 and KFKT2017B01), the Open Research Fund of National Mobile Communications Research Laboratory, Southeast University (no. 2015D02), and the Brain Korea BK21 plus.

## References

- [1] H. Xie, F. Gao, and S. Jin, "An Overview of Low-Rank Channel Estimation for Massive MIMO Systems," *IEEE Access*, vol. 4, pp. 7313–7321, 2016.
- [2] Z. Zhao, M. Peng, Z. Ding, W. Wang, and H.-H. Chen, "Denoise-and-forward network coding for two-way relay MIMO systems," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 2, pp. 775–788, 2014.
- [3] J. Fan, L. Li, H. Zhang, and W. Chen, "Denoise-and-Forward Two-Path Successive Relaying with DBPSK Modulation," *IEEE Wireless Communications Letters*, vol. 6, no. 1, pp. 42–45, 2017.
- [4] M. Masjedi, A. M. Doost Hoseini, and S. Gazor, "Non-Coherent Detection and Denoise-and-Forward Two-Way Relay Networks," *IEEE Transactions on Communications*, vol. 64, no. 11, pp. 4497–4505, 2016.
- [5] H. X. Nguyen, N. N. Tran, and H. T. Nguyen, "Performance analysis of adaptive decode-and-forward relaying in noncoherent cooperative networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, no. 1, article no. 281, 2013.
- [6] Q. Cui, T. Yuan, X. Tao, A. A. Dowhuszko, and R. Jantti, "Energy efficiency analysis of two-way DF relay system with non-ideal power amplifiers," *IEEE Communications Letters*, vol. 18, no. 7, pp. 1254–1257, 2014.
- [7] C. Wang, H. Chen, Q. Yin, A. Feng, and A. F. Molisch, "Multi-user two-way relay networks with distributed beamforming," *IEEE Transactions on Wireless Communications*, vol. 10, no. 10, pp. 3460–3471, 2011.
- [8] Y. Rong, "Joint source and relay optimization for two-way linear non-regenerative MIMO relay communications," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6533–6546, 2012.
- [9] V. Havary-Nassab, S. Shahbazpanahi, and A. Grami, "Optimal distributed beamforming for two-way relay networks," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, part 1, pp. 1238–1250, 2010.
- [10] K. An, M. Lin, J. Ouyang, and H. Wei, "Performance analysis of beamforming in two-hop AF relay networks with antenna correlation and interference," *AEÜ - International Journal of Electronics and Communications*, vol. 68, no. 7, pp. 587–594, 2014.
- [11] J. Joung and J. Choi, "Linear Precoder Design for an AF Two-Way MIMO Relay Node with No Source Node Precoding," *IEEE Transactions on Vehicular Technology*, 2017.

- [12] R. Budhiraja and B. Ramamurthi, "Transceiver Design for Nonconcurrent Two-Way MIMO AF Relaying with QoS Guarantees," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9651–9661, 2016.
- [13] J.-M. Kang and H.-M. Kim, "Training Designs for Estimation of Spatially Correlated Fading Channels in MIMO Amplify-and-Forward Two-Way Multi-Relay Networks," *IEEE Communications Letters*, vol. 20, no. 4, pp. 772–775, 2016.
- [14] C. Hu and Y. Chiu, "Robust joint optimization for MIMO AF multiple-relay systems with correlated channel uncertainties," *International Journal of Electronics and Communications*, vol. 68, no. 5, pp. 453–464, 2014.
- [15] H. Tang, W. Chen, and J. Li, "Achieving Optimality in Robust Joint Optimization of Linear Transceiver Design," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1814–1819, 2016.
- [16] K. Cumanan, Z. Ding, Y. Rahulamathavan, M. M. Molu, and H.-H. Chen, "Robust MMSE beamforming for multiantenna relay networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 3900–3912, 2017.
- [17] G. Zhang, X. Li, M. Cui, G. Li, and Y. Tan, "Transceiver design for cognitive multi-user MIMO multi-relay networks using imperfect CSI," *AEÜ - International Journal of Electronics and Communications*, vol. 70, no. 5, pp. 544–557, 2016.
- [18] L. Li, Y. Xu, Z. Chen, and J. Fang, "Robust transmit design for secure AF relay networks with imperfect CSI," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, article no. 142, 2016.
- [19] H. Shen, W. Xu, J. Wang, and C. Zhao, "A worst-case robust beamforming design for multi-antenna AF relaying," *IEEE Communications Letters*, vol. 17, no. 4, pp. 713–716, 2013.
- [20] N. Lee, H. J. Yang, and J. Chun, "Achievable sum-rate maximizing af relay beamforming scheme in two-way relay channels," in *Proceedings of the ICC 2008 - 2008 IEEE International Conference on Communications Workshops*, pp. 300–305, China, May 2008.
- [21] E. A. Gharavol and E. G. Larsson, "The sign-definiteness lemma and its applications to robust transceiver optimization for multiuser MIMO systems," *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 238–252, 2013.
- [22] P. Ubaidulla and S. Aissa, "Robust two-way cognitive relaying: Precoder designs under interference constraints and imperfect CSI," *IEEE Transactions on Wireless Communications*, vol. 13, no. 5, pp. 2478–2489, 2014.
- [23] B. K. Chalise and L. Vandendorpe, "MIMO relay design for multipoint-to-multipoint communications with imperfect channel state information," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2785–2796, 2009.
- [24] S. A. Vorobyov, A. B. Gershman, and Z.-Q. Luo, "Robust adaptive beamforming using worst-case performance optimization: a solution to the signal mismatch problem," *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 313–324, 2003.
- [25] W. Duan, M. Wen, X. Jiang, Y. Yan, and M. Ho Lee, "Sum-rate maximization and robust beamforming design for MIMO two-way relay networks with reciprocal and imperfect CSI," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, article no. 157, 2016.
- [26] J. Liu, F. Gao, and Z. Qiu, "Robust transceiver design for downlink multiuser MIMO AF relay systems," *IEEE Transactions on Wireless Communications*, vol. 14, no. 4, pp. 2218–2231, 2015.
- [27] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.

## Research Article

# A Conjugate Gradient Algorithm under Yuan-Wei-Lu Line Search Technique for Large-Scale Minimization Optimization Models

Xiangrong Li <sup>1</sup>, Songhua Wang,<sup>2</sup> Zhongzhou Jin,<sup>3</sup> and Hongtruong Pham<sup>4</sup>

<sup>1</sup>College of Mathematics and Information Science, Guangxi University, Nanning, Guangxi 530004, China

<sup>2</sup>School of Mathematics and Statistics, Baishan University, Baishan, Guangxi 533000, China

<sup>3</sup>Business School, Guangxi University, Nanning, Guangxi 530004, China

<sup>4</sup>Thai Nguyen University of Economics and Business Administration, Thai Nguyen, Vietnam

Correspondence should be addressed to Xiangrong Li; xrl68@163.com

Received 17 October 2017; Accepted 27 November 2017; Published 15 January 2018

Academic Editor: Guillermo Cabrera-Guerrero

Copyright © 2018 Xiangrong Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper gives a modified Hestenes and Stiefel (HS) conjugate gradient algorithm under the Yuan-Wei-Lu inexact line search technique for large-scale unconstrained optimization problems, where the proposed algorithm has the following properties: (1) the new search direction possesses not only a sufficient descent property but also a trust region feature; (2) the presented algorithm has global convergence for nonconvex functions; (3) the numerical experiment showed that the new algorithm is more effective than similar algorithms.

## 1. Introduction

Consider the minimization optimization models defined by

$$\min \{f(x) \mid x \in \mathfrak{R}^n\}, \quad (1)$$

where the function  $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$  and  $f \in C^2$ . There exist many good algorithms for (1), such as the quasi-Newton methods [1] and the conjugate gradient methods [2–5], where the iterative formula of the conjugate gradient algorithm for (1) is designed by

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, 2, \dots, \quad (2)$$

where  $x_k$  is the  $k$ th iterative point,  $\alpha_k$  is the steplength, and  $d_k$  is the so-called conjugate gradient search direction with

$$d_{k+1} = \begin{cases} -g(x_{k+1}) + \beta_k d_k, & \text{if } k \geq 1, \\ -g(x_{k+1}), & \text{if } k = 0, \end{cases} \quad (3)$$

where  $\beta_k$  is a scalar determined from different conjugate gradient formulas and the HS method [3] is one of the most well-known conjugate gradient methods, which is

$$\beta_k^{\text{HS}} = \frac{g_{k+1}^T y_k}{d_k^T y_k}, \quad (4)$$

where  $y_k = g_{k+1} - g_k$ ,  $g_{k+1} = \nabla f(x_{k+1})$  and  $g_k = \nabla f(x_k)$ . The HS method has good numerical results for (1); however, the convergent theory is not interesting especially for the nonconvex function. At present, there exist many good conjugate gradients (see [6–8], etc.). Yuan, Wei, and Lu [9] gave a modified weak Wolfe-Powell (we called it YWL) line search for steplength  $\alpha_k$  designed by

$$\begin{aligned} & f(x_k + \alpha_k d_k) \\ & \leq f_k + \delta \alpha_k g_k^T d_k + \alpha_k \min \left[ -\delta_1 g_k^T d_k, \delta \frac{\alpha_k}{2} \|d_k\|^2 \right], \end{aligned} \quad (5)$$

$$\begin{aligned} & g(x_k + \alpha_k d_k)^T d_k \\ & \geq \sigma g_k^T d_k + \min \left[ -\delta_1 g_k^T d_k, \delta \alpha_k \|d_k\|^2 \right], \end{aligned} \quad (6)$$

where  $\delta \in (0, 1/2)$ ,  $\delta_1 \in (0, \delta)$ ,  $\sigma \in (\delta, 1)$ , and  $\|\cdot\|$  denotes the Euclidean norm. It is well known that there exist two open problems which are the global convergence of the normal BFGS method and the global convergence of the PRP method for nonconvex functions under the inexact line search technique, where the first problem is regarded as one of the most difficult one thousand mathematical problems of the 20th century [10]. Yuan et al. [9] partly solved these two open problems under the YWL technique, and the numerical performance shows that the YWL technique is more competitive than the normal weak Wolfe-Powell technique. Further study work can be found in their paper [11]. By (5), it is not difficult to see that the YWL conditions are equivalent to the weak Wolfe-Powell (WWP) conditions if  $-\delta_1 g_k^T d_k < \delta(\alpha_k/2)\|d_k\|^2$  holds, which implies that the YWL technique includes the WWP technique in some sense. Motivated by the above observations, we will make a further study and propose a new algorithm for (1). The main features of this paper are as follows:

- (i) A modified HS conjugate gradient formula is given, which has not only a sufficient descent property but also a trust region feature.

- (ii) The global convergence of the given HS conjugate gradient algorithm for nonconvex functions is established.
- (iii) Numerical results show that the new HS conjugate gradient algorithm under the YWL line search technique is better than the normal weak Wolfe-Powell technique.

This paper is organized as follows. In Section 2, a modified HS conjugate gradient algorithm is introduced. The global convergence of the given algorithm for nonconvex functions is established in Section 3 and numerical results are reported in Section 4.

## 2. Motivation and Algorithm

The nonlinear conjugate gradient algorithm is simple and has low memory requirement properties and is very effective for large-scale optimization problems, where the HS method is one of the most effective methods. However, the normal HS method has good numerical performance but fails in the convergence of nonconvex functions under the inexact line search technique. In order to overcome this shortcoming, a modified HS formula is defined by

$$d_{k+1} = \begin{cases} -g_{k+1} + \frac{g_{k+1}^T y_k d_k - d_k^T g_{k+1} y_k}{\psi_1 \|d_k\|^2 + 2\psi_2 \|d_k\| \|y_k\| + \|g_k\|^2 + \psi_3 \|y_k\|^2}, & \text{if } k \geq 1, \\ -g_{k+1}, & \text{if } k = 0, \end{cases} \quad (7)$$

where  $y_k = g_{k+1} - g_k$  and  $\psi_1, \psi_2$ , and  $\psi_3$  are positive constants. This formula is inspired by the idea of these two papers [6, 8]. In recent years, lots of scholars like to study the three-term conjugate gradient formula because of its good properties [7]. In the next section, we will prove that the new formula possesses not only a sufficient descent property but also a trust region feature. The sufficient descent property is good for the convergence and the trust region makes the convergence easy to prove. Now, we give the steps of the proposed algorithm as follows.

*Algorithm 1* (the modified three-term HS conjugate gradient algorithm (M-TT-HS-A)).

*Step 1.*  $x_1 \in \mathfrak{R}^n$ ,  $\epsilon \in (0, 1)$ ,  $\delta \in (0, 1/2)$ ,  $\delta_1 \in (0, \delta)$ ,  $\sigma \in (\delta, 1)$ ,  $\psi_1 > 0$ ,  $\psi_2 > 0$ ,  $\psi_3 > 0$ . Let  $k = 1$  and  $d_1 = -g(x_1)$ .

*Step 2.* If  $\|g_k\| \leq \epsilon$  holds, stop.

*Step 3.* Find  $\alpha_k$  by the YWL line search satisfying (5) and (6).

*Step 4.* Let  $x_{k+1} = x_k + \alpha_k d_k$ .

*Step 5.* Compute the direction  $d_{k+1}$  by (7).

*Step 6.* The algorithm stops if  $\|g_{k+1}\| \leq \epsilon$ .

*Step 7.* Let  $k = k + 1$  and go to Step 3.

## 3. Sufficient Descent Property, Trust Region Feature, and Global Convergence

This section will prove some properties of Algorithm 1.

**Lemma 2.** *The search direction  $d_k$  is designed by (7); the following two relations hold:*

$$g_k^T d_k = -\|g_k\|^2, \quad (8)$$

$$\|d_k\| \leq \psi^* \|g_k\|, \quad (9)$$

where  $\psi^* > 0$  is a constant.

*Proof.* If  $k = 1$ , it is easy to have (8) and (9). If  $k \geq 1$ , by formula (7), we have

$$\begin{aligned} g_{k+1}^T d_{k+1} &= g_{k+1}^T \left[ -g_{k+1} \right. \\ &\quad \left. + \frac{g_{k+1}^T y_k d_k - d_k^T g_{k+1} y_k}{\psi_1 \|d_k\|^2 + 2\psi_2 \|d_k\| \|y_k\| + \|g_k\|^2 + \psi_3 \|y_k\|^2} \right] \\ &= -\|g_{k+1}\|^2 \\ &\quad + \frac{g(x_{k+1})^T y_k g_{k+1}^T d_k - d_k^T g(x_{k+1}) g_{k+1}^T y_k}{\psi_1 \|d_k\|^2 + 2\psi_2 \|d_k\| \|y_k\| + \|g_k\|^2 + \psi_3 \|y_k\|^2} \end{aligned}$$

$$\begin{aligned}
 &= -\|g_{k+1}\|^2, \\
 \|d_{k+1}\| &= \left\| -g(x_{k+1}) \right. \\
 &\quad \left. + \frac{g_{k+1}^T y_k d_k - d_k^T g_{k+1} y_k}{\psi_1 \|d_k\|^2 + 2\psi_2 \|d_k\| \|y_k\| + \|g_k\|^2 + \psi_3 \|y_k\|^2} \right\| \\
 &\leq \|g_{k+1}\| \\
 &\quad + \frac{\|g(x_{k+1})\| \|y_k\| \|d_k\| + \|d_k\| \|g(x_{k+1})\| \|y_k\|}{\psi_1 \|d_k\|^2 + 2\psi_2 \|d_k\| \|y_k\| + \|g_k\|^2 + \psi_3 \|y_k\|^2} \\
 &\leq \|g_{k+1}\| + \frac{2\|g(x_{k+1})\| \|y_k\| \|d_k\|}{2\psi_2 \|d_k\| \|y_k\|} \leq \left(1 + \frac{1}{\psi_2}\right) \\
 &\quad \cdot \|g_{k+1}\|. \tag{10}
 \end{aligned}$$

Then, (8) holds as well as (9) by letting  $\psi^* \in [1 + 1/\psi_2, +\infty)$ . This completes the proof.  $\square$

Inequality (8) shows that the new formula has a sufficient descent property and inequality (9) proves that the new formula possesses a trust region feature. Both of these properties (8) and (9) are good theory characters and they play an important role in the global convergence of a conjugate gradient algorithm. The following global convergence theory will explain all this.

The following general assumptions are needed.

*Assumption A.* (i) The defined level set  $L_0 = \{x \mid f(x) \leq f(x_1)\}$  is bounded.

(ii) The objective function  $f(x)$  is bounded below, twice continuously differentiable, and is Lipschitz continuous; namely, the following inequality is true:

$$\|g(x) - g(y)\| \leq L \|x - y\|, \quad x, y \in \mathfrak{R}^n, \tag{11}$$

where  $L > 0$  is the Lipschitz constant.

By Lemma 2 and Assumption A, similar to [9], it is not difficult to show that the YWL line search technique is reasonable and Algorithm 1 is well defined. Here, we do not state it anymore. Now, we prove the global convergence of Algorithm 1 for nonconvex functions.

**Theorem 3.** *Let Assumption A hold, and the iterate sequence  $\{x_k, \alpha_k, d_k, g(x_k)\}$  is generated by M-TT-HS-A. Then, the relation*

$$\lim_{k \rightarrow \infty} \|g_k\| = 0 \tag{12}$$

is true.

*Proof.* By (5), (8), and (9), we obtain

$$\begin{aligned}
 f(x_k + \alpha_k d_k) &\leq f(x_k) + \delta \alpha_k g_k^T d_k \\
 &\quad + \alpha_k \min \left[ -\delta_1 g_k^T d_k, \delta \frac{\alpha_k}{2} \|d_k\|^2 \right] \\
 &\leq f(x_k) + \delta \alpha_k g_k^T d_k - \alpha_k \delta_1 g_k^T d_k \\
 &\leq f(x_k) + \alpha_k (\delta - \delta_1) g_k^T d_k \\
 &\leq f(x_k) - \alpha_k (\delta - \delta_1) \|g_k\|^2. \tag{13}
 \end{aligned}$$

Summing these inequalities for  $k = 0$  to  $\infty$  and using Assumption A (ii) generate

$$\sum_{k=0}^{\infty} (\gamma_1 - \gamma) \alpha_k (\delta - \delta_1) \|g_k\|^2 \leq f(x_1) - f_{\infty} < +\infty. \tag{14}$$

Inequality (14) implies that

$$\lim_{k \rightarrow \infty} \alpha_k \|g_k\|^2 = 0 \tag{15}$$

is true. By (6) and (8) again, we get

$$\begin{aligned}
 g(x_k + \alpha_k d_k)^T d_k &\geq \sigma g_k^T d_k \\
 &\quad + \min \left[ -\delta_1 g_k^T d_k, \delta \alpha_k \|d_k\|^2 \right] \\
 &\geq \sigma g_k^T d_k. \tag{16}
 \end{aligned}$$

Thus, the inequality

$$\begin{aligned}
 -(\sigma - 1) \|g_k\|^2 &\leq (\sigma - 1) g_k^T d_k \\
 &\leq [g(x_k + \alpha_k d_k) - g(x_k)]^T d_k \\
 &\leq \|g(x_k + \alpha_k d_k) - g(x_k)\| \|d_k\| \\
 &\leq \alpha_k L \|d_k\|^2 \tag{17}
 \end{aligned}$$

holds, where the first inequality follows (8) and the last inequality follows (11). Then, we have

$$\alpha_k \geq \frac{(1 - \sigma) \|g_k\|^2}{L \|d_k\|^2} \geq \frac{(1 - \sigma) \|g_k\|^2}{L (\psi^*)^2 \|g_k\|^2} = \frac{(1 - \sigma)}{L (\psi^*)^2}. \tag{18}$$

By (15) and (18), we have

$$\lim_{k \rightarrow \infty} \|g_k\|^2 = 0. \tag{19}$$

Therefore, we get (12) and the proof is complete.  $\square$

#### 4. Numerical Results Performance

This section will give numerical results of Algorithm 1 and the similar algorithms for comparing them. We will give another two algorithms for comparison; they are listed as follows.

*Algorithm 2* (the normal three-term formula [8] under the YWL technique).

*Step 1.*  $x_1 \in \mathfrak{R}^n$ ,  $\epsilon \in (0, 1)$ ,  $\delta \in (0, 1/2)$ ,  $\delta_1 \in (0, \delta)$ ,  $\sigma \in (\delta, 1)$ ,  $\psi_1 > 0$ ,  $\psi_2 > 0$ ,  $\psi_3 > 0$ . Let  $k = 1$  and  $d_1 = -g(x_1)$ .

*Step 2.* If  $\|g_k\| \leq \epsilon$  holds, stop.

*Step 3.* Find  $\alpha_k$  by the YWL line search satisfying (5) and (6).

*Step 4.* Let  $x_{k+1} = x_k + \alpha_k d_k$ .

*Step 5.* Compute the direction  $d_{k+1}$  by

$$d_{k+1} = \begin{cases} -g_{k+1} + \frac{g_{k+1}^T y_k d_k - d_k^T g_{k+1} y_k}{\|g_k\|^2}, & \text{if } k \geq 1, \\ -g_{k+1}, & \text{if } k = 0. \end{cases} \quad (20)$$

*Step 6.* The algorithm stops if  $\|g_{k+1}\| \leq \epsilon$ .

*Step 7.* Let  $k = k + 1$  and go to Step 3.

*Algorithm 3* (the normal three-term formula [8] under the WWP technique).

*Step 1.*  $x_1 \in \mathfrak{R}^n$ ,  $\epsilon \in (0, 1)$ ,  $\delta \in (0, 1/2)$ ,  $\sigma \in (\delta, 1)$ . Let  $k = 1$  and  $d_1 = -g(x_1)$ .

*Step 2.* If  $\|g_k\| \leq \epsilon$  holds, stop.

*Step 3.* Find  $\alpha_k$  by the WWP line search satisfying

$$\begin{aligned} f(x_k + \alpha_k d_k) &\leq f_k + \delta \alpha_k g_k^T d_k, \\ g(x_k + \alpha_k d_k)^T d_k &\geq \sigma g_k^T d_k. \end{aligned} \quad (21)$$

*Step 4.* Let  $x_{k+1} = x_k + \alpha_k d_k$ .

*Step 5.* Compute the direction  $d_{k+1}$  by (20).

*Step 6.* The algorithm stops if  $\|g_{k+1}\| \leq \epsilon$ .

*Step 7.* Let  $k = k + 1$  and go to Step 3.

*4.1. Problems and Experiment.* The following are some notes.

*Test Problems.* These problems and the related initial points are listed in Table 1; the detailed problems can be found in Andrei [12], and some papers also use these problems [13].

*Experiments.* Codes are run on Intel(R) Xeon(R) CPU, E5507 @2.27 GHz, and 6.00 GB memory and Windows 7 operation system and written by MATLAB R2009a.

*Parameters.*  $\delta = 0.1$ ,  $\delta_1 = 0.05$ ,  $\sigma = 0.9$ , and  $\psi_1 = \psi_2 = \psi_3 = 0.001$ .

*Dimension.* Large-scale dimensions  $n = 3000, 6000, 12000$ , and 30000.

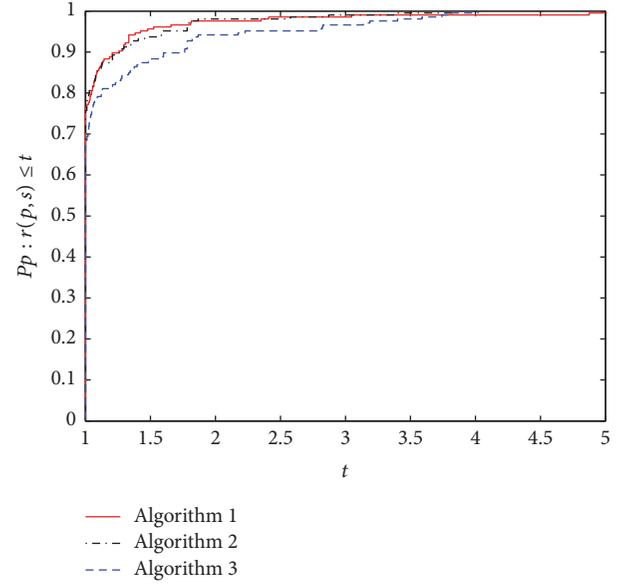


FIGURE 1: NI performance of these methods.

*Stop Rules.* *Himmelblau* stop rule: if  $|f(x_k)| > 1e - 5$ , let  $\text{stop1} = |f(x_k) - f(x_{k+1})|/|f(x_k)|$ ; otherwise, set  $\text{stop1} = |f(x_k) - f(x_{k+1})|$ . If  $\text{stop1} < 1e - 5$  or  $\|g(x_k)\| < 1e - 6$  holds, the program stops.

*Other Cases.* The line search technique accepts  $\alpha_k$  if the searching number is more than 6 and the algorithm will stop if the total iteration number is larger than 800.

The numerical results are listed in Table 2, where

“Number” is the tested problems number;

“Dim.” is the problems dimension;

“NI” is the total iteration number;

“CPU” is the system CPU time in seconds;

“NFG” is the total number of functions and gradients.

*4.2. Results and Discussion.* We use the tool of Dolan and Moré [14] to analyze the efficiency of the three given algorithms. Figures 1 and 2 show that the performance of Algorithm 1 is the best and that Algorithm 1 has the best robust property among those three methods and Algorithm 2 is better than Algorithm 3, which shows that the given formula (7) is competitive to the normal three-term conjugate gradient formula (20) and the YWL line search technique is more effective than the norm WWP technique, and all of these conclusions are coincident with the results of [9]. Algorithm 1 in Figure 3 is competitive to the other two algorithms and it has the best robust property. It is not difficult to see that Figure 3 shows that Algorithm 1 is not so good and we think the reason is formula (7) or the YWL technique since more information is needed and hence more CPU time is necessary.

TABLE 1: Test problems.

Number	Problem	$x_0$
(1)	Extended Freudenstein and Roth Function	[0.5, -2, ..., 0.5, -2]
(2)	Extended Trigonometric Function	[0.2, 0.2, ..., 0.2]
(3)	Extended Rosenbrock Function	[-1.2, 1, -1.2, 1, ..., -1.2, 1]
(4)	Extended Beale Function	[1, 0.8, ..., 1, 0.8]
(5)	Raydan 1 Function	[1, 1, ..., 1]
(6)	Raydan 2 Function	[1, 1, ..., 1]
(7)	Diagonal 1 Function	[1/n, 1/n, ..., 1/n]
(8)	Diagonal 3 Function	[1, 1, ..., 1]
(9)	Hager Function	[1, 1, ..., 1]
(10)	Generalized Tridiagonal 1 Function	[2, 2, ..., 2]
(11)	Extended Tridiagonal 1 Function	[2, 2, ..., 2]
(12)	Extended Three Exponential Terms Function	[0.1, 0.1, ..., 0.1]
(13)	Diagonal 4 Function	[1, 1, ..., 1, 1]
(14)	Diagonal 5 Function	[1.1, 1.1, ..., 1.1]
(15)	Extended Himmelblau Function	[1, 1, ..., 1]
(16)	Generalized PSC1 Function	[3, 0.1, ..., 3, 0.1]
(17)	Extended PSC1 Function	[3, 0.1, ..., 3, 0.1]
(18)	Extended Block Diagonal BD1 Function	[0.1, 0.1, ..., 0.1]
(19)	Extended Maratos Function	[1.1, 0.1, ..., 1.1, 0.1]
(20)	Extended Cliff Function	[0, -1, ..., 0, -1]
(21)	Extended Wood Function	[-3, -1, -3, -1, ..., -3, -1]
(22)	Extended Quadratic Penalty QP1 Function	[1, 1, ..., 1]
(23)	Extended Quadratic Penalty QP2 Function	[1, 1, ..., 1]
(24)	A Quadratic Function QF2 Function	[0.5, 0.5, ..., 0.5]
(25)	Extended EPI Function	[1.5, 1.5, ..., 1.5]
(26)	Extended Tridiagonal-2 Function	[1, 1, ..., 1]
(27)	BDQRTIC Function (CUTE)	[1, 1, ..., 1]
(28)	ARWHEAD Function (CUTE)	[1, 1, ..., 1]
(29)	NONDIA (Shanno-78) Function (CUTE)	[-1, -1, ..., -1]
(30)	DQDRTIC Function (CUTEr)	[3, 3, 3, ..., 3]
(31)	EG2 Function (CUTE)	[1, 1, 1, ..., 1]
(32)	DIXMAANA Function (CUTE)	[2, 2, 2, ..., 2]
(33)	DIXMAANB Function (CUTE)	[2, 2, 2, ..., 2]
(34)	DIXMAANC Function (CUTE)	[2, 2, 2, ..., 2]
(35)	Partial Perturbed Quadratic Function	[0.5, 0.5, ..., 0.5]
(36)	Broyden Tridiagonal Function	[-1, -1, ..., -1]
(37)	EDENSCH Function (CUTE)	[0, 0, ..., 0]
(38)	LIARWHD Function (CUTEr)	[4, 4, ..., 4]
(39)	DIAGONAL 6 Function	[1, 1, ..., 1]
(40)	DIXON3DQ Function (CUTE)	[-1, -1, ..., -1]
(41)	DIXMAANF Function (CUTE)	[2, 2, 2, ..., 2]
(42)	DIXMAANG Function (CUTE)	[2, 2, 2, ..., 2]
(43)	DIXMAANH Function (CUTE)	[2, 2, 2, ..., 2]
(44)	DIXMAANI Function (CUTE)	[2, 2, 2, ..., 2]
(45)	DIXMAANJ Function (CUTE)	[2, 2, 2, ..., 2]
(46)	DIXMAANK Function (CUTE)	[2, 2, 2, ..., 2]
(47)	DIXMAANL Function (CUTE)	[2, 2, 2, ..., 2]
(48)	DIXMAAND Function (CUTE)	[2, 2, 2, ..., 2]
(49)	ENGVAL1 Function (CUTE)	[2, 2, 2, ..., 2]
(50)	Extended DENSCHNB Function (CUTE)	[1, 1, ..., 1]
(51)	SINQUAD Function (CUTE)	[0.1, 0.1, ..., 0.1]
(52)	Partial Perturbed Quadratic PPQ2 Function	[0.5, 0.5, ..., 0.5]

TABLE 2: Numerical results.

Number	Dim.	Algorithm 1			Algorithm 2			Algorithm 3		
		NI	NFG	CPU	NI	NFG	CPU	NI	NFG	CPU
(1)	3000	15	38	0.093601	20	56	0.140401	20	56	0.124801
	6000	24	66	0.358802	21	59	0.280802	21	59	0.249602
	12000	22	59	0.577204	17	46	0.405603	17	46	0.405603
	30000	22	59	1.357209	17	46	0.998406	17	46	1.029607
(2)	3000	74	163	0.327602	74	163	0.343202	83	181	0.358802
	6000	81	176	0.702005	81	176	0.686404	81	176	0.655204
	12000	83	182	1.388409	83	182	1.341609	83	182	1.357209
	30000	88	194	3.572423	88	194	3.588023	89	196	3.478822
(3)	3000	88	262	0.124801	99	283	0.124801	73	216	0.093601
	6000	83	277	0.234002	112	329	0.265202	112	329	0.249602
	12000	78	272	0.390002	16	66	0.078001	16	66	0.093601
	30000	49	119	0.483603	46	182	0.561604	46	182	0.468003
(4)	3000	32	87	0.109201	21	57	0.093601	21	57	0.0624
	6000	24	65	0.171601	27	74	0.187201	27	74	0.187201
	12000	23	62	0.312002	23	63	0.265202	26	69	0.343202
	30000	24	67	0.748805	29	78	0.858005	25	66	0.717605
(5)	3000	23	51	0.0624	23	51	0.0312	23	51	0.0624
	6000	23	51	0.093601	23	51	0.093601	23	51	0.0624
	12000	23	51	0.171601	23	51	0.156001	23	51	0.124801
	30000	23	51	0.390002	23	51	0.358802	23	51	0.312002
(6)	3000	12	26	0.0156	12	26	0.0156	12	26	0.0156
	6000	12	26	0.0468	12	26	0.0468	12	26	0.0312
	12000	12	26	0.093601	12	26	0.0624	12	26	0.078
	30000	12	26	0.171601	12	26	0.171601	12	26	0.156001
(7)	3000	2	9	0.0312	2	9	0.0312	2	9	0.0312
	6000	2	9	0	2	9	0	2	9	0
	12000	2	9	0	2	9	0.0312	2	9	0.0312
	30000	2	9	0.0312	2	9	0.0468	2	9	0.0468
(8)	3000	17	36	0.0624	17	36	0.0468	17	36	0.0624
	6000	19	40	0.093601	19	40	0.109201	19	40	0.078
	12000	19	40	0.156001	19	40	0.156001	19	40	0.171601
	30000	19	40	0.405603	19	40	0.405603	19	40	0.374402
(9)	3000	24	113	0.078001	24	113	0.078	24	113	0.078
	6000	22	52	0.093601	21	50	0.109201	22	52	0.109201
	12000	2	9	0.0156	2	9	0.0156	2	9	0.0312
	30000	2	9	0.0624	2	9	0.0468	2	9	0.078
(10)	3000	6	15	0.655204	6	15	0.592804	6	15	0.592804
	6000	6	15	2.043613	6	15	1.981213	6	15	1.996813
	12000	6	15	7.207246	6	15	7.503648	6	15	7.394447
	30000	3	8	21.590538	3	8	21.528138	3	8	21.481338
(11)	3000	34	81	1.762811	35	86	1.653611	35	86	1.63801
	6000	47	114	6.520842	34	77	4.383628	36	88	4.74243
	12000	36	88	14.820095	37	93	15.241298	37	93	15.100897
	30000	43	102	92.492993	39	100	86.050152	39	100	85.83175
(12)	3000	14	30	0.0468	14	30	0.0624	14	30	0.0468
	6000	21	44	0.093601	21	44	0.109201	21	44	0.078001
	12000	24	50	0.202801	24	50	0.218401	24	50	0.218401
	30000	24	50	0.592804	24	50	0.483603	24	50	0.468003
(13)	3000	4	13	0.0312	3	10	0.0156	3	10	0
	6000	4	13	0	3	10	0	3	10	0
	12000	4	13	0.0312	3	10	0.0312	3	10	0
	30000	4	13	0.0468	3	10	0.0312	3	10	0.0312

TABLE 2: Continued.

Number	Dim.	Algorithm 1			Algorithm 2			Algorithm 3		
		NI	NFG	CPU	NI	NFG	CPU	NI	NFG	CPU
(14)	3000	3	9	0.0312	3	9	0.0312	3	9	0
	6000	3	9	0.0312	3	9	0.0312	3	9	0.0312
	12000	3	9	0.0624	3	9	0.0624	3	9	0.0312
	30000	3	9	0.078001	3	9	0.078	3	9	0.078001
(15)	3000	9	48	0.0468	9	48	0.0312	9	48	0.0312
	6000	128	292	0.312002	42	108	0.093601	42	108	0.078
	12000	24	114	0.140401	29	139	0.156001	29	139	0.156001
	30000	13	39	0.156001	16	45	0.171601	16	45	0.156001
(16)	3000	33	76	0.109201	33	76	0.093601	42	90	0.093601
	6000	32	73	0.187201	32	73	0.156001	41	88	0.187201
	12000	32	73	0.312002	32	73	0.312002	43	95	0.390003
	30000	33	78	0.811205	33	78	0.780005	37	86	0.889206
(17)	3000	9	44	0.124801	9	44	0.156001	9	43	0.124801
	6000	9	44	0.280802	9	44	0.265202	9	44	0.249602
	12000	9	44	0.499203	9	44	0.514803	9	44	0.514803
	30000	9	44	1.279208	9	44	1.232408	9	44	1.232408
(18)	3000	11	88	0.0624	11	87	0.0312	11	86	0.0312
	6000	11	88	0.093601	11	83	0.093601	11	83	0.093601
	12000	47	207	0.436803	20	115	0.202801	20	115	0.187201
	30000	39	158	0.873606	89	248	1.63801	31	134	0.733205
(19)	3000	28	56	0.0312	28	56	0.0468	28	56	0.0312
	6000	38	76	0.078	38	76	0.093601	38	76	0.0624
	12000	159	504	0.717605	170	594	0.733205	66	162	0.249602
	30000	7	16	0.078	7	16	0.0468	7	16	0.078
(20)	3000	97	212	0.234002	97	212	0.218401	97	212	0.202801
	6000	106	230	0.452403	106	230	0.421203	106	230	0.390003
	12000	93	206	0.733205	93	206	0.702005	93	206	0.686404
	30000	87	194	1.700411	87	194	1.54441	87	194	1.52881
(21)	3000	36	90	0.0468	36	90	0.0468	36	90	0.078
	6000	37	94	0.124801	37	94	0.093601	37	94	0.093601
	12000	34	88	0.202801	33	86	0.156001	33	86	0.156001
	30000	33	83	0.436803	36	92	0.405603	36	92	0.421203
(22)	3000	42	92	0.0624	42	92	0.078	42	92	0.078
	6000	41	90	0.109201	41	90	0.124801	41	90	0.093601
	12000	41	90	0.234002	41	90	0.187201	41	90	0.202801
	30000	45	98	0.514803	45	98	0.499203	45	98	0.483603
(23)	3000	42	88	0.109201	42	88	0.078001	42	88	0.078
	6000	71	146	0.296402	71	146	0.296402	71	146	0.280802
	12000	37	80	0.312002	37	80	0.265202	37	80	0.280802
	30000	52	110	1.029607	52	110	0.904806	52	110	0.920406
(24)	3000	3	7	0	3	7	0	3	7	0
	6000	3	7	0.0312	3	7	0.0312	3	7	0
	12000	2	5	0	2	5	0	2	5	0
	30000	2	5	0.0312	2	5	0.0312	2	5	0.0624
(25)	3000	4	8	0	4	8	0	4	8	0
	6000	5	10	0.0156	5	10	0.0156	5	10	0.0312
	12000	7	14	0.0312	7	14	0.0312	7	14	0.0624
	30000	10	20	0.140401	10	20	0.124801	10	20	0.124801
(26)	3000	12	24	0.0312	12	24	0.0468	12	24	0.0468
	6000	16	32	0.078001	16	32	0.0468	16	32	0.0468
	12000	21	42	0.124801	21	42	0.109201	21	42	0.109201
	30000	30	62	0.421203	30	62	0.405603	31	62	0.405603

TABLE 2: Continued.

Number	Dim.	Algorithm 1			Algorithm 2			Algorithm 3		
		NI	NFG	CPU	NI	NFG	CPU	NI	NFG	CPU
(27)	3000	18	57	1.341609	14	48	1.029607	14	48	0.998406
	6000	5	16	1.060807	8	28	1.778411	8	28	1.778411
	12000	5	16	3.494422	8	28	6.084039	8	28	6.068439
	30000	5	16	21.138135	17	55	77.9693	17	55	77.9537
(28)	3000	9	23	0.0312	9	26	0.0312	9	26	0.0156
	6000	12	30	0.0312	12	30	0.0468	12	30	0.0312
	12000	10	25	0.078	10	28	0.078001	10	28	0.0468
	30000	12	29	0.140401	11	30	0.109201	11	30	0.124801
(29)	3000	26	52	0.0468	26	52	0.0312	26	52	0.0312
	6000	27	54	0.078001	27	54	0.0468	27	54	0.0468
	12000	29	58	0.124801	29	58	0.109201	29	58	0.093601
	30000	23	46	0.249602	23	46	0.171601	23	46	0.202801
(30)	3000	49	119	0.0624	27	68	0.0624	27	68	0.0468
	6000	37	92	0.109201	26	65	0.0468	26	65	0.078
	12000	29	75	0.124801	28	69	0.124801	28	69	0.093601
	30000	32	77	0.296402	27	68	0.234002	27	68	0.280802
(31)	3000	4	21	0	4	21	0	4	21	0
	6000	4	21	0.0312	4	21	0.0156	4	21	0.0312
	12000	4	21	0.0312	4	21	0.0312	4	21	0.0468
	30000	4	21	0.093601	4	21	0.109201	4	21	0.109201
(32)	3000	22	48	0.296402	22	48	0.296402	22	48	0.265202
	6000	23	50	0.577204	23	50	0.561604	23	50	0.608404
	12000	24	52	1.201208	24	52	1.185608	24	52	1.185608
	30000	25	54	3.026419	25	54	3.04202	25	54	3.010819
(33)	3000	38	80	0.483603	38	80	0.483603	36	76	0.436803
	6000	39	82	0.951606	39	82	0.967206	37	78	0.889206
	12000	41	86	1.950013	41	86	1.965613	38	80	1.856412
	30000	42	88	5.054432	42	88	5.070032	39	82	4.69563
(34)	3000	66	136	0.842405	66	136	0.795605	66	136	0.826805
	6000	69	142	1.669211	69	142	1.669211	69	142	1.63801
	12000	72	148	3.478822	72	148	3.447622	72	148	3.478822
	30000	75	154	9.001258	75	154	8.970057	75	154	8.876457
(35)	3000	85	175	26.036567	86	177	26.364169	86	177	26.379769
	6000	45	110	57.79837	84	180	103.631464	84	180	103.693865
	12000	58	146	296.932303	35	98	186.514796	35	98	186.421195
	30000	75	201	2473.645457	96	250	3169.191515	96	250	3169.331916
(36)	3000	33	76	1.279208	48	106	1.794012	49	105	1.809612
	6000	47	106	7.020045	50	112	7.254046	50	112	7.254047
	12000	49	101	26.410969	50	103	26.785372	50	103	26.769772
	30000	40	95	154.050987	41	97	155.096194	41	97	154.940193
(37)	3000	23	48	0.124801	23	48	0.124801	23	48	0.124801
	6000	23	48	0.265202	23	48	0.265202	23	48	0.265202
	12000	23	48	0.499203	23	48	0.499203	23	48	0.483603
	30000	23	48	1.232408	23	48	1.216808	23	48	1.170007
(38)	3000	12	36	0.0468	19	59	0.0468	19	59	0.0624
	6000	47	150	0.156001	26	76	0.093601	26	76	0.0624
	12000	68	178	0.374402	13	41	0.0624	13	41	0.093601
	30000	11	30	0.140401	42	99	0.468003	42	99	0.452403
(39)	3000	21	44	0.483603	21	44	0.514803	21	44	0.483603
	6000	22	46	1.950013	22	46	1.934412	22	46	1.887612
	12000	23	48	7.441248	23	48	7.441248	23	48	7.425648
	30000	24	50	51.979533	24	50	51.121528	24	50	51.027927

TABLE 2: Continued.

Number	Dim.	Algorithm 1			Algorithm 2			Algorithm 3		
		NI	NFG	CPU	NI	NFG	CPU	NI	NFG	CPU
(40)	3000	449	906	0.468003	800	1619	0.826805	800	1615	0.780005
	6000	449	906	0.748805	800	1619	1.294808	800	1615	1.279208
	12000	449	906	1.263608	800	1619	2.371215	800	1615	2.308815
	30000	449	906	3.07322	800	1619	5.288434	800	1615	5.163633
(41)	3000	93	192	1.170007	94	194	1.201208	164	339	2.028013
	6000	73	152	1.825212	73	152	1.794012	204	419	5.038832
	12000	80	166	3.931225	80	166	3.915625	251	513	12.246079
	30000	90	186	11.029271	90	186	10.93567	323	657	39.07825
(42)	3000	154	318	1.996813	150	310	1.887612	118	242	1.51321
	6000	151	308	3.775224	151	323	3.822024	134	274	3.307221
	12000	188	386	9.40686	202	414	9.937264	156	318	7.675249
	30000	215	453	26.972573	239	492	29.390588	198	402	24.164555
(43)	3000	96	205	1.248008	71	158	0.889206	65	150	0.826805
	6000	65	151	1.669211	58	129	1.48201	81	172	1.996813
	12000	81	177	4.102826	85	185	4.212027	100	210	4.929632
	30000	89	193	11.169672	90	195	11.122871	91	205	11.310072
(44)	3000	142	294	1.856412	137	284	1.731611	128	267	1.63801
	6000	159	328	4.024826	153	316	3.837625	162	339	4.056026
	12000	178	366	9.001258	170	350	8.455254	254	531	12.682881
	30000	212	434	26.379769	199	408	24.554557	800	1612	99.060635
(45)	3000	93	192	1.170007	94	194	1.185608	164	339	2.074813
	6000	73	152	1.840812	73	152	1.825212	206	423	5.101233
	12000	80	166	3.962425	80	166	3.900025	255	521	12.44888
	30000	90	186	11.013671	89	184	10.85767	333	677	40.404259
(46)	3000	72	163	0.951606	73	176	0.982806	99	210	1.279208
	6000	119	250	2.979619	115	242	2.839218	123	258	3.026419
	12000	138	288	6.848444	147	306	7.238446	137	286	6.739243
	30000	174	360	21.574938	172	356	21.091335	170	352	20.794933
(47)	3000	302	613	3.946825	283	575	3.634823	800	1606	10.654868
	6000	398	805	10.124465	368	745	9.31326	800	1606	20.950934
	12000	359	727	18.189717	402	813	20.139729	800	1606	41.621067
	30000	437	883	54.912352	437	883	54.366348	800	1606	103.584664
(48)	3000	26	58	0.343202	26	58	0.327602	27	60	0.327602
	6000	27	60	0.686404	27	60	0.670804	27	60	0.639604
	12000	27	60	1.341609	27	60	1.326009	27	60	1.326009
	30000	27	60	3.354022	27	60	3.291621	28	62	3.416422
(49)	3000	41	86	3.276021	41	86	3.244821	42	88	3.322821
	6000	40	84	10.670468	40	84	10.654868	41	86	10.90447
	12000	40	84	37.48704	40	84	37.799042	42	88	39.655454
	30000	42	88	250.459606	42	88	249.367599	44	92	261.020873
(50)	3000	35	72	0.0624	35	72	0.0624	35	72	0.0468
	6000	36	74	0.093601	36	74	0.109201	36	74	0.078
	12000	38	78	0.171601	38	78	0.156001	38	78	0.156001
	30000	39	80	0.452403	39	80	0.390002	39	80	0.421203
(51)	3000	28	97	1.918812	51	183	3.432022	51	183	3.447622
	6000	47	164	10.608068	56	218	13.104084	56	218	13.088484
	12000	42	135	30.872598	37	160	31.964605	37	160	31.886604
(52)	3000	569	1582	193.176038	800	2260	272.783349	800	2260	272.627348
	6000	800	2184	1074.254086	800	2200	1077.140105	800	2200	1074.83129
	30000	728	2025	24653.74284	784	2183	26511.62115	784	2183	26511.71475

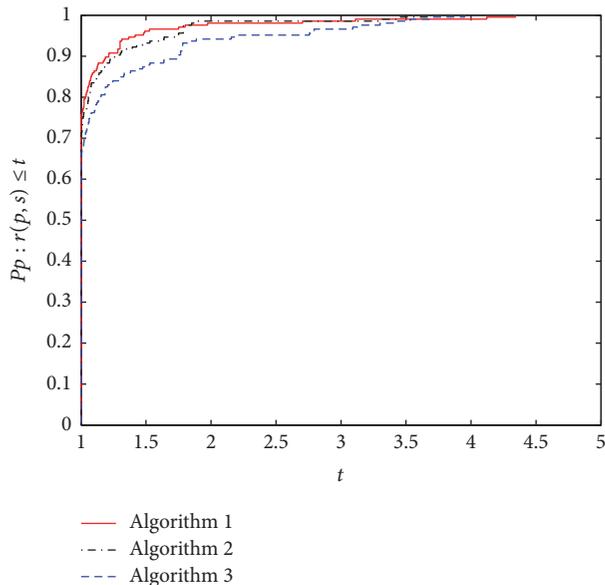


FIGURE 2: NFG performance of these methods.

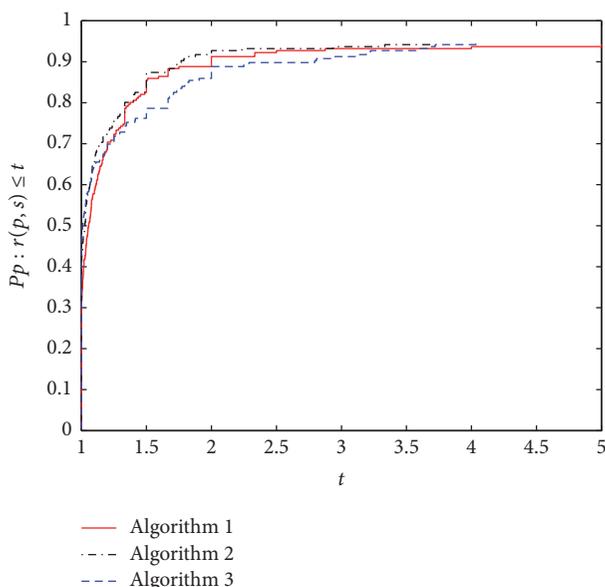


FIGURE 3: CPU time performance of these methods.

## 5. Conclusions

This paper proposes a modified HS three-term conjugate gradient algorithm for large-scale optimization problems and the given algorithm has some good features.

(1) The modified HS three-term conjugate gradient possesses a trust region property, which makes the global convergence of the general functions easy to get. However, the normal HS formula including many other conjugate gradient formulas does not have this feature, which may be the crucial point for the global convergence of the general functions.

(2) The largest dimension of the test problems is 30000 variables and the numerical results show that the presented

algorithm is competitive to other similar methods. More experiments will be done to prove the performance of the proposed algorithm in the future.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Authors' Contributions

Mr. Xiangrong Li and Dr. Songhua Wang wrote this paper in English. Dr. Zhongzhou Jin and Dr. Hongtruong Pham carried out the experiment. All the authors read and approved the final manuscript.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grants nos. 11661009 and 11261006), the Guangxi Science Fund for Distinguished Young Scholars (no. 2015GXNSFGA139001), the Guangxi Natural Science Key Fund (no. 2017GXNSFDA198046), and the Basic Ability Promotion Project of Guangxi Young and Middle-Aged Teachers (no. 2017KY0019).

## References

- [1] G. Yuan and Z. Wei, "Convergence analysis of a modified BFGS method on convex minimizations," *Computational optimization and applications*, vol. 47, no. 2, pp. 237–255, 2010.
- [2] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The Computer Journal*, vol. 7, pp. 149–154, 1964.
- [3] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436 (1953), 1952.
- [4] E. Polak and G. Ribière, "Note sur la convergence de méthodes de directions conjuguées," *Revue Française d'Informatique et de Recherche Opérationnelle*, vol. 3, no. 16, pp. 35–43, 1969.
- [5] B. T. Polyak, "The conjugate gradient method in extremal problems," *USSR Computational Mathematics and Mathematical Physics*, vol. 9, no. 4, pp. 94–112, 1969.
- [6] G. Yuan, Z. Meng, and Y. Li, "A modified Hestenes and Stiefel conjugate gradient algorithm for large-scale nonsmooth minimizations and nonlinear equations," *Journal of Optimization Theory and Applications*, vol. 168, no. 1, pp. 129–152, 2016.
- [7] G. Yuan and M. Zhang, "A three-terms Polak-Ribière-Polyak conjugate gradient algorithm for large-scale nonlinear equations," *Journal of Computational and Applied Mathematics*, vol. 286, pp. 186–195, 2015.
- [8] L. Zhang, W. Zhou, and D.-H. Li, "A descent modified Polak-Ribière-Polyak conjugate method and its global convergence," *IMA Journal of Numerical Analysis (IMAJNA)*, vol. 26, no. 4, pp. 629–640, 2006.
- [9] G. Yuan, Z. Wei, and X. Lu, "Global convergence of BFGS and PRP methods under a modified weak Wolfe-Powell line search," *Applied Mathematical Modelling: Simulation and Computation for Engineering and Environmental Systems*, vol. 47, pp. 811–825, 2017.

- [10] A. S. Nemirovsky and D. B. Yudin, *Problem complexity and method efficiency in optimization*, John Wiley & Sons, Inc., NY, New York, USA, 1983.
- [11] G. Yuan, Z. Sheng, B. Wang, W. Hu, and C. Li, "The global convergence of a modified BFGS method for nonconvex functions," *Journal of Computational and Applied Mathematics*, vol. 327, pp. 274–294, 2018.
- [12] N. Andrei, "An unconstrained optimization test functions collection," *Advanced Modeling and Optimization*, vol. 10, no. 1, pp. 147–161, 2008.
- [13] Y. Wu, "A modified three-term PRP conjugate gradient algorithm for optimization models," *Journal of Inequalities and Applications*, Paper No. 97, 14 pages, 2017.
- [14] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.