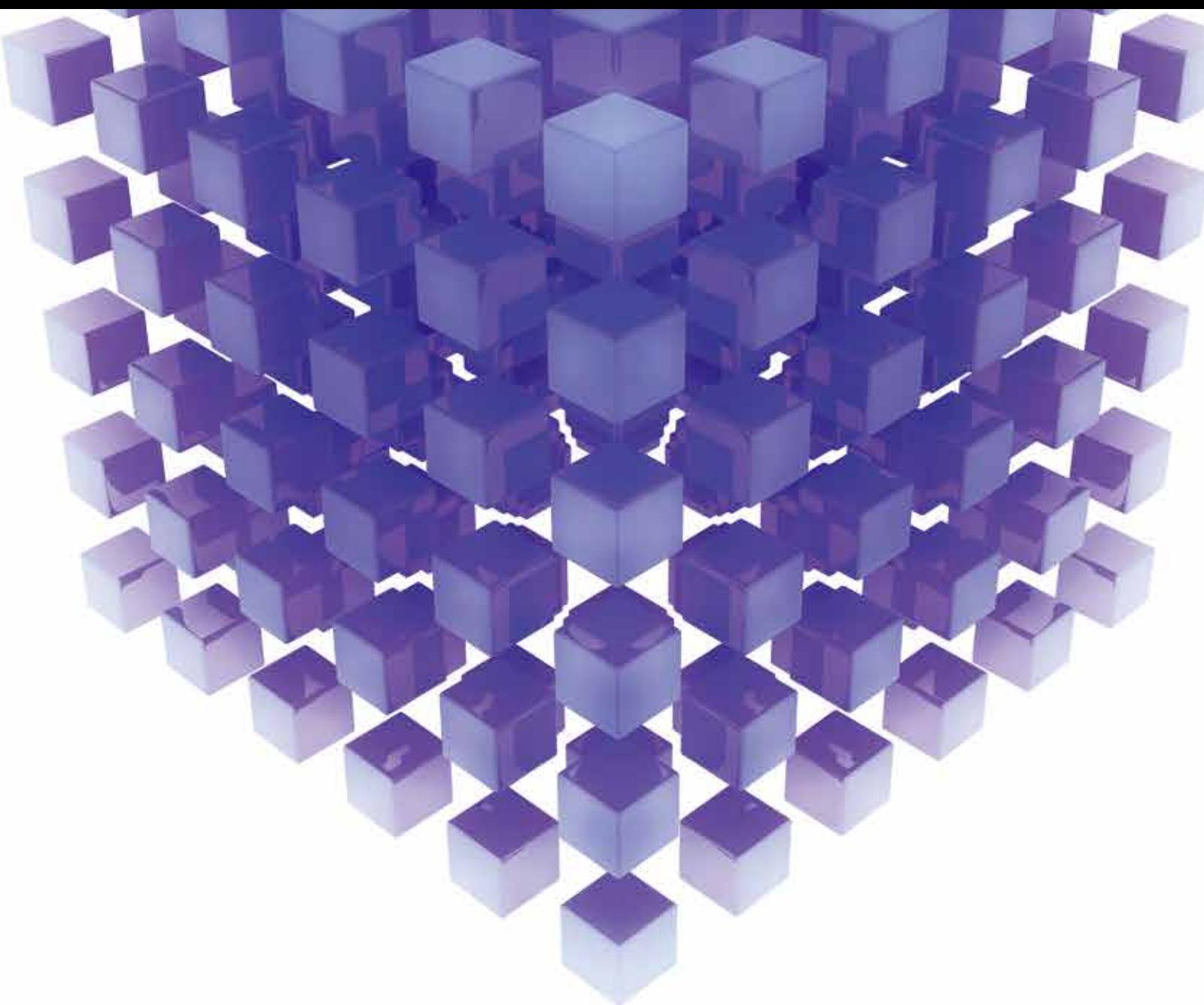


INTELLIGENT TECHNIQUES FOR SIMULATION AND MODELLING

GUEST Editors: William Guo, Chih-Cheng Hung, Ross Shepherd, Zhiqiang Ma,
and Siddhivinayak Kulkarni





Intelligent Techniques for Simulation and Modelling

Intelligent Techniques for Simulation and Modelling

Guest Editors: William Guo, Chih-Cheng Hung,
Ross Shepherd, Zhiqiang Ma, and Siddhivinayak Kulkarni



Copyright © 2014 Hindawi Publishing Corporation. 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

- M. Abd El Aziz, Egypt
E. M. Abdel-Rahman, Canada
R. K. Abu Al-Rub, USA
Sarp Adali, South Africa
Salvatore Alfonzetti, Italy
Igor Andrianov, Germany
Sebastian Anita, Romania
W. Assawinchaichote, Thailand
Erwei Bai, USA
Ezzat G. Bakhoum, USA
José M. Balthazar, Brazil
R. K. Bera, India
C. Bérenguer, France
Jonathan N. Blakely, USA
Stefano Boccaletti, Spain
Stephane P.A. Bordas, USA
Daniela Bosco, Italy
M. Boutayeb, France
Michael J. Brennan, UK
Salvatore Caddemi, Italy
Piermarco Cannarsa, Italy
Jose E. Capilla, Spain
Carlo Cattani, Italy
M. M. Cavalcanti, Brazil
Diego J. Celentano, Chile
Mohammed Chadli, France
Arindam Chakraborty, USA
Yong-Kui Chang, China
Michael J. Chappell, UK
Kui Fu Chen, China
Xinkai Chen, Japan
Kue-Hong Chen, Taiwan
Jyh-Horng Chou, Taiwan
Slim Choura, Tunisia
C. Cruz-Hernandez, Mexico
Swagatam Das, India
Filippo de Monte, Italy
Antonio Desimone, Italy
Yannis Dimakopoulos, Greece
Baocang Ding, China
Joao B. R. Do Val, Brazil
Daoyi Dong, Australia
B. Dubey, India
Horst Ecker, Austria
M. Onder Efe, Turkey
Elmetwally Elabbasy, Egypt
A. Elías-Zúñiga, Mexico
Anders Eriksson, Sweden
Vedat S. Erturk, Turkey
Moez Feki, Tunisia
Ricardo Femat, Mexico
Roberto A. Valente, Portugal
C. Fuerte-Esquivel, Mexico
Zoran Gajic, USA
Ugo Galvanetto, Italy
Furong Gao, Hong Kong
Xin-Lin Gao, USA
Behrouz Gatmiri, Iran
Oleg V. Gendelman, Israel
Didier Georges, France
P. Gonçalves, Brazil
Oded Gottlieb, Israel
Fabrizio Greco, Italy
Quang Phuc Ha, Australia
M. R. Hajj, USA
Tony S. W. Hann, Taiwan
Thomas Hanne, Switzerland
K. R. (Stevanovic) Hedrih, Serbia
M.I. Herreros, Spain
Wei-Chiang Hong, Taiwan
Jaromir Horacek, Czech Republic
Huabing Huang, China
Chuangxia Huang, China
Gordon Huang, Canada
Yi Feng Hung, Taiwan
Hai-Feng Huo, China
Asier Ibeas, Spain
Anuar Ishak, Malaysia
Reza Jazar, Australia
Zhijian Ji, China
Jun Jiang, China
J. J. Judice, Portugal
Tadeusz Kaczorek, Poland
Tamas Kalmar-Nagy, USA
Tomasz Kapitaniak, Poland
Hamid Reza Karimi, Norway
Metin O. Kaya, Turkey
Nikolaos Kazantzis, USA
Farzad Khani, Iran
K. Krabbenhoft, Australia
Ren-Jieh Kuo, Taiwan
Jurgen Kurths, Germany
Claude Lamarque, France
Usik Lee, Korea
Marek Leflik, Poland
Stefano Lenci, Italy
Roman Lewandowski, Poland
Shanling Li, Canada
Ming Li, China
Jian Li, China
Shihua Li, China
Teh-Lu Liao, Taiwan
Panos Liatsis, UK
Shueei M. Lin, Taiwan
Yi-Kuei Lin, Taiwan
Jui-Sheng Lin, Taiwan
Yuji Liu, China
Wanquan Liu, Australia
Bin Liu, Australia
Paolo Lonetti, Italy
V. C. Loukopoulos, Greece
Junguo Lu, China
Chien-Yu Lu, Taiwan
Alexei Mailybaev, Brazil
M. K. Maiti, India
O. D. Makinde, South Africa
R. Martinez-Guerra, Mexico
Driss Mehdi, France
Roderick Melnik, Canada
Xinzhu Meng, China
Yuri V. Mikhlin, Ukraine
G. Milovanovic, Serbia
Ebrahim Momoniat, South Africa
Trung Nguyen Thoi, Vietnam
Hung Nguyen-Xuan, Vietnam
Ben T. Nohara, Japan
Sotiris K. Ntouyas, Greece
Gerard Olivar, Colombia
Claudio Padra, Argentina
Bijaya Ketan Panigrahi, India
Francesco Pellicano, Italy
Matjaz Perc, Slovenia
Vu Ngoc Phat, Vietnam
M. do Rosário Pinho, Portugal
A. Pogromsky, The Netherlands

Seppo Pohjolainen, Finland	Cheng Shao, China	Moran Wang, China
Stanislav Potapenko, Canada	Bo Shen, Germany	Yijing Wang, China
Sergio Preidikman, USA	Jian-Jun Shu, Singapore	Gerhard-Wilhelm Weber, Turkey
Carsten Proppe, Germany	Zhan Shu, UK	J. S. Witteveen, The Netherlands
Hector Puebla, Mexico	Dan Simon, USA	Kwok-Wo Wong, Hong Kong
Justo Puerto, Spain	Luciano Simoni, Italy	Ligang Wu, China
Dane Quinn, USA	Grigori M. Sisoev, UK	Zhengguang Wu, China
K. R. Rajagopal, USA	Christos H. Skiadas, Greece	Gongnan Xie, China
Gianluca Ranzi, Australia	Davide Spinello, Canada	Wang Xing-yuan, China
Sivaguru Ravindran, USA	Sri Sridharan, USA	Xi Frank Xu, USA
G. Rega, Italy	Rolf Stenberg, Finland	Xuping Xu, USA
Pedro Ribeiro, Portugal	Changyin Sun, China	Jun-Juh Yan, Taiwan
J. Rodellar, Spain	Jitao Sun, China	Xing-Gang Yan, UK
R. Rodriguez-Lopez, Spain	Xi-Ming Sun, China	Suh-Yuh Yang, Taiwan
A. J. Rodriguez-Luis, Spain	Andrzej Swierniak, Poland	Mahmoud T. Yassen, Egypt
Ignacio Romero, Spain	Yang Tang, Germany	Mohammad I. Younis, USA
Hamid Ronagh, Australia	Allen Tannenbaum, USA	Bo Yu, China
Carla Roque, Portugal	Cristian Toma, Romania	Huang Yuan, Germany
R. R. García, Spain	Irina N. Trendafilova, UK	S.P. Yung, Hong Kong
Manouchehr Salehi, Iran	Alberto Trevisani, Italy	Ion Zaballa, Spain
Miguel A. Sanjuán, Spain	Jung-Fa Tsai, Taiwan	A. M. Zenkour, Saudi Arabia
Ilmar F. Santos, Denmark	K. Vajravelu, USA	Jianming Zhan, China
Nickolas S. Sapidis, Greece	Victoria Vampa, Argentina	Xu Zhang, China
E. J. Sapountzakis, Greece	Josep Vehi, Spain	Yingwei Zhang, China
Bozidar Sarler, Slovenia	Stefano Vidoli, Italy	Lu Zhen, China
Andrey V. Savkin, Australia	Xiaojun Wang, China	Liancun Zheng, China
Massimo Scalia, Italy	Dan Wang, China	Jian Guo Zhou, UK
Mohamed A. Seddeek, Egypt	Youqing Wang, China	Zexuan Zhu, China
A. P. Seyranian, Russia	Yongqi Wang, Germany	Mustapha Zidi, France
Leonid Shaikhet, Ukraine	Cheng C. Wang, Taiwan	

Contents

Intelligent Techniques for Simulation and Modelling, William Guo, Chih-Cheng Hung, Ross Shepherd, Zhiqiang Ma, and Siddhivinayak Kulkarni
Volume 2014, Article ID 179457, 3 pages

Crop Yield Forecasting Using Artificial Neural Networks: A Comparison between Spatial and Temporal Models, William W. Guo and Heru Xue
Volume 2014, Article ID 857865, 7 pages

Design and Implementation of Fuzzy Logic Controller for Online Computer Controlled Steering System for Navigation of a Teleoperated Agricultural Vehicle, Prema Kannan, Senthil Kumar Natarajan, and Subhransu Sekhar Dash
Volume 2013, Article ID 590861, 10 pages

A Sarsa(λ) Algorithm Based on Double-Layer Fuzzy Reasoning, Quan Liu, Xiang Mu, Wei Huang, Qiming Fu, and Yonggang Zhang
Volume 2013, Article ID 561026, 9 pages

Semophysical Modelling of the Nonlinear Dynamics of a Surface Craft with LS-SVM, David Moreno-Salinas, Dictino Chaos, Eva Besada-Portas, José Antonio López-Orozco, Jesús M. de la Cruz, and Joaquín Aranda
Volume 2013, Article ID 890120, 13 pages

An Ensemble Method for High-Dimensional Multilabel Data, Huawei Liu, Zhonglong Zheng, Jianmin Zhao, and Ronghua Ye
Volume 2013, Article ID 208051, 5 pages

BMCloud: Minimizing Repair Bandwidth and Maintenance Cost in Cloud Storage, Chao Yin, Changsheng Xie, Jiguang Wan, Chih-Cheng Hung, Jinjiang Liu, and Yihua Lan
Volume 2013, Article ID 756185, 11 pages

Structural Attack to Anonymous Graph of Social Networks, Tieying Zhu, Shanshan Wang, Xiangtao Li, Zhiguo Zhou, and Rimeng Zhang
Volume 2013, Article ID 237024, 8 pages

A Hyperbolic Tangent Adaptive PID + LQR Control Applied to a Step-Down Converter Using Poles Placement Design Implemented in FPGA, Marcelo Dias Pedroso, Claudinor Bitencourt Nascimento, Angelo Marcelo Tusset, and Maurício dos Santos Kaster
Volume 2013, Article ID 101650, 8 pages

New Algorithms for Bidirectional Singleton Arc Consistency, Yonggang Zhang, Qian Yin, Xingjun Zhu, Zhanshan Li, Sibo Zhang, and Quan Liu
Volume 2013, Article ID 904768, 10 pages

Machine Directional Register System Modeling for Shaft-Less Drive Gravure Printing Machines, Shanhai Liu, Xuesong Mei, Jian Li, and Li'e Ma
Volume 2013, Article ID 251636, 10 pages

An Efficient Schema for Cloud Systems Based on SSD Cache Technology, Jinjiang Liu, Yihua Lan, Jingjing Liang, Quanzhou Cheng, Chih-Cheng Hung, Chao Yin, and Jiadong Sun
Volume 2013, Article ID 109781, 9 pages

Opposition-Based Animal Migration Optimization, Yi Cao, Xiangtao Li, and Jianan Wang
Volume 2013, Article ID 308250, 7 pages

Time-Domain Joint Parameter Estimation of Chirp Signal Based on SVR, Xueqian Liu and Hongyi Yu
Volume 2013, Article ID 952743, 9 pages

Multiscale Feature Model for Terrain Data Based on Adaptive Spatial Neighborhood, Huijie Zhang, Yun Ma, Zhiqiang Ma, Xinting He, Yaxin Liu, and Zijun Feng
Volume 2013, Article ID 278754, 12 pages

Message Passing Algorithm for Solving QBF Using More Reasoning, Su Weihua, Yin Minghao, Wang Jianan, and Zhou Junping
Volume 2013, Article ID 165927, 6 pages

A Topology Structure Based Outer Membrane Proteins Segment Alignment Method, Han Wang, Bo Liu, Pingping Sun, and Zhiqiang Ma
Volume 2013, Article ID 541359, 10 pages

Model Predictive Control Based on Kalman Filter for Constrained Hammerstein-Wiener Systems, Man Hong and Shao Cheng
Volume 2013, Article ID 104702, 6 pages

Cross-Domain Personalized Learning Resources Recommendation Method, Long Wang, Zhiyong Zeng, Ruizhi Li, and Hua Pang
Volume 2013, Article ID 958785, 6 pages

The Prediction of Calpain Cleavage Sites with the mRMR and IFS Approaches, Wenyi Zhang, Xin Xu, Longjia Jia, Zhiqiang Ma, Na Luo, and Jianan Wang
Volume 2013, Article ID 861269, 6 pages

A New Skeleton Feature Extraction Method for Terrain Model Using Profile Recognition and Morphological Simplification, Huijie Zhang, Zhiqiang Ma, Yaxin Liu, Xinting He, and Yun Ma
Volume 2013, Article ID 523729, 16 pages

Behavior-Based Fuzzy Control for Mobile Robot Navigation, Hongwei Mo, Qirong Tang, and Longlong Meng
Volume 2013, Article ID 561451, 10 pages

The Algorithm of Link Prediction on Social Network, Liyan Dong, Yongli Li, Han Yin, Huang Le, and Mao Rui
Volume 2013, Article ID 125123, 7 pages

A Novel Framework Based on ACO and PSO for RNA Secondary Structure Prediction, Gang Wang, Wen-yi Zhang, Qiao Ning, and Hui-ling Chen
Volume 2013, Article ID 796304, 8 pages

Identification of Protein Pupylation Sites Using Bi-Profile Bayes Feature Extraction and Ensemble Learning, Xiaowei Zhao, Jian Zhang, Qiao Ning, Pingping Sun, Zhiqiang Ma, and Minghao Yin
Volume 2013, Article ID 283129, 7 pages

Sliding Mode Control for NSVs with Input Constraint Using Neural Network and Disturbance Observer, Yan-long Zhou and Mou Chen
Volume 2013, Article ID 904830, 12 pages

An Abstract Description Method of Map-Reduce-Merge Using Haskell, Lei Liu, Dongqing Liu, Shuai Lü, and Peng Zhang
Volume 2013, Article ID 147593, 12 pages

Optimization of *p*-Cycle Placement for Differentiated Levels of Protection, Hong Hui Li and Xue Liang Fu
Volume 2013, Article ID 920564, 5 pages

An Accurate and Efficient Method to Predict Y-NO Bond Homolysis Bond Dissociation Energies, Hong Zhi Li, Lin Li, Zi Yan Zhong, Yi Han, LiHong Hu, and Ying Hua Lu
Volume 2013, Article ID 860357, 10 pages

Infrared Target Detection and Location for Visual Surveillance Using Fusion Scheme of Visible and Infrared Images, Zi-Jun Feng, Xiao-Ling Zhang, Li-Yong Yuan, and Jia-Nan Wang
Volume 2013, Article ID 720979, 7 pages

Phase Transitions of Contingent Planning Problem, Junping Zhou, Weihua Su, Zhiqiang Ma, and Minghao Yin
Volume 2013, Article ID 687578, 9 pages

Reconfigurable Architecture for Elliptic Curve Cryptography Using FPGA, A. Kaleel Rahuman and G. Athisha
Volume 2013, Article ID 675161, 8 pages

On the Completeness of Pruning Techniques for Planning with Conditional Effects, Dunbo Cai, Sheng Xu, Tongzhou Zhao, and Yanduo Zhang
Volume 2013, Article ID 398245, 6 pages

Editorial

Intelligent Techniques for Simulation and Modelling

**William Guo,¹ Chih-Cheng Hung,² Ross Shepherd,¹
Zhiqiang Ma,³ and Siddhivinayak Kulkarni⁴**

¹ School of Engineering and Technology, Central Queensland University, North Rockhampton, QLD 4702, Australia

² Center for Biometrics Research, Southern Polytechnic State University, Atlanta, GA 30060-2896, USA

³ College of Computer Science and Information Technology, Northeast Normal University, Changchun, Jilin 130017, China

⁴ School of Science, Information Technology and Engineering, Federation University, Ballarat, VIC 3353, Australia

Correspondence should be addressed to William Guo; w.guo@cqu.edu.au

Received 23 January 2014; Accepted 23 January 2014; Published 26 February 2014

Copyright © 2014 William Guo 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.

Emergence and development of computational intelligence in the last 20 years have broadened the scope of simulation and modelling in many scientific and engineering disciplines. In conjunction with the 2013 International Conference on Information Technology, Computation and Applications (ICITCA2013) held in Hohhot, China on August 6–8, 2013, an open special issue on “Intelligent Techniques for Simulation and Modelling (ITSM)” in Mathematical Problems in Engineering (MPE) was organized for publishing high-quality papers selected from both the conference and open submissions. In total, fifty-three papers were submitted to this special issue through MPE’s online submission system by the closure. Contributors were from more than 70 institutions in Australia, Brazil, Canada, China, India, Iran, Mexico, Serbia, Spain, and USA. After rigorous peer review of each of these submissions, thirty-two revised papers were accepted for publishing in this special issue. Depending on the completion time of revision and final editing, these papers have been published progressively since September 2013.

An engineering or a scientific problem is often based on some physical natures. To solve such a problem, we need to formulate the problem as a mathematical model in terms of variables, functions, and equations. The process of creating such a model, solving it, and interpreting the result in physical, chemical, geometric, engineering, or other terms is *modelling*. Such a model is likely comprised of a number of unknown factors, such as temporal and spatial constraints. By varying one or more of these unknown factors within the limits, different outcomes can be generated from the mathematical model and/or solution. These outcomes

correlate to different physical, chemical, geometric, engineering scenarios. This model-based iterative mathematical computation is *simulation*. Modelling and simulation are two different but closely related approaches in many disciplines, which are illustrated in Figure 1 [1].

Many techniques play important roles in the stages of finding solutions for and interpreting the solutions for different purposes from the simulation and modelling process. Commonly used techniques in simulation and modelling include statistics, intelligent methods, numeric computations, experiments, and combinations [2–6]. Popular and emerging applications of simulation and modelling include systems control, forecasting, scheduling or planning, network security, and various optimizations [7–10]. All these techniques and applications are found in the 32 accepted papers in this special issue.

Categorically, there are six papers in adaptive or optimal control of various systems. Pedroso et al. presented an adaptive control that integrates two linear control strategies for the stepdown converter. Hong and Cheng designed a predictive control algorithm based on the Kalman filter for constrained Hammerstein—Wiener systems. Liu et al. studied the mathematical model of the machine directional register system for the multicolor gravure printing machines in order to improve the control accuracy of the register system. Mo et al. presented a new behavior-based fuzzy control method for mobile robot navigation which can deal with uncertainties in unknown environments and has the ability to accommodate different behaviors. Prema et al. described the design, modeling, simulation, control, and implementation of an

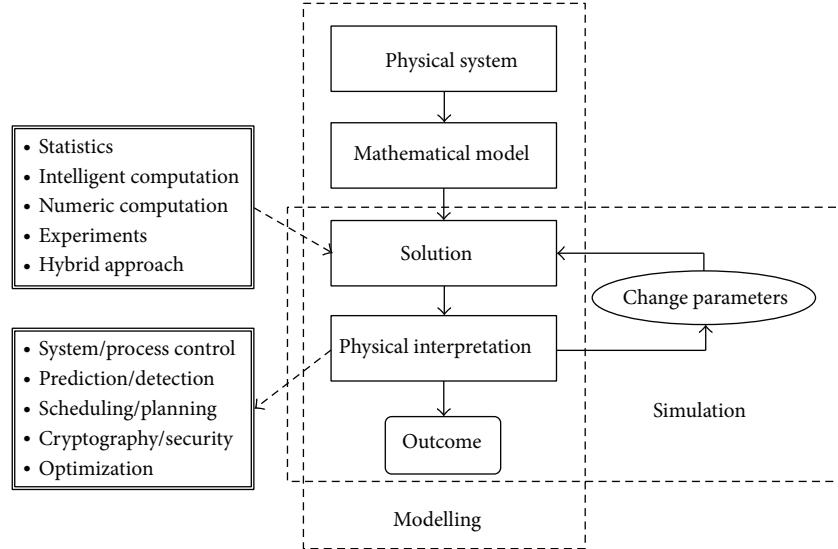


FIGURE 1: The procedure of modelling and simulation (modified from [1]).

intelligent system for controlling the teleoperated agricultural vehicle. Zhou and Chen proposed a sliding mode control scheme for near space vehicles with strong nonlinearity, high coupling, parameter uncertainty and unknown time-varying disturbance based on radial basis function neural networks (RBFNNs) and the nonlinear disturbance observer.

The five papers in scheduling, planning, and resources allocation include the paper on a new pruning strategy for search based planning and its realisation through numeric approximation by Cai et al., the paper on a new algorithm for phase transitions of the planning problem by Zhou et al., the paper on crop yield forecasting using neural network models by Guo and Xue, the paper on a new similarity-based link prediction algorithm for social networks by Dong et al., and the paper on a new method for learning resources allocation or recommendation by Wang et al. There are nine papers that proposed new algorithms for feature detection, extraction, or reconstruction in digital data and image processing contributed by Feng et al., Zhao et al., Wang et al., Zhang et al., Wang et al., Li et al., Zhang et al., Liu et al., and Su et al., respectively.

The four papers in encryption and security in digital data storage and communication networks are contributed by Zhu et al. on an effective structural attack to deanonymize social graph data on social networks, Rahuman and Athisha on a new reconfigurable architecture for elliptic curve cryptography using FPGA, Li and Fu on a new scalable model for the design of p-cycles with the differentiated levels of node protection for communication networks, and Yin et al. on a new efficient cloud storage system to protect and recover data in cloud environment. The remaining eight papers in optimization algorithms for various applications were contributed by Liu et al., Liu et al., Wang et al., Liu et al., Moreno-Salinas et al., Zhang et al., and Liu and Yu, respectively.

All these papers have made new contributions to the broad area of modelling and simulation.

Acknowledgments

We are grateful to all authors and reviewers for their contributions to this special issue.

William Guo
Chih-Cheng Hung
Ross Shepherd
Zhiqiang Ma
Siddhivinayak Kulkarni

References

- [1] W. W. Guo, *Advanced Mathematics for Engineering and Applied Sciences*, Pearson, Melbourne, Australia, 2014.
- [2] D. Bradford and C. C. Hung, “Pollen-based bee algorithm for data clustering-a computational model,” *Progress in Intelligent Computing and Applications*, vol. 1, no. 1, pp. 16–36, 2012.
- [3] T. Kohonen, *Self-Organizing Maps*, Springer, New York, NY, USA, 2001.
- [4] R. Shepherd, T. Meuwissen, and J. Woolliams, “Genomic selection and complex trait prediction using a fast EM algorithm applied to genome-wide markers,” *BMC Bioinformatics*, vol. 11, pp. 529–541, 2010.
- [5] M. M. Li, W. Guo, B. Verma, K. Tickle, and J. O’Connor, “Intelligent methods for solving inverse problems of backscattering spectra with noise: a comparison between neural networks and simulated annealing,” *Neural Computing and Applications*, vol. 18, no. 5, pp. 423–430, 2009.
- [6] C. C. Hung, S. Kulkarni, and B. C. Kuo, “A new weighted fuzzy C-means clustering algorithm for remotely sensed image classification,” *IEEE Journal on Selected Topics in Signal Processing*, vol. 5, no. 3, pp. 543–553, 2011.
- [7] X.-S. Yang, *Engineering Optimization*, Wiley, New York, NY, USA, 2010.
- [8] G. Dong, W. W. Guo, and K. Tickle, “Solving the traveling salesman problem using cooperative genetic ant systems,” *Expert Systems with Applications*, vol. 39, no. 5, pp. 5006–5011, 2012.

- [9] E. K. P. Chong and S. H. Zak, *An Introduction to Optimization (3rd Edition)*, Wiley, New York, NY, USA, 2008.
- [10] F. Giordano, W. P. Fox, and S. Horton, *A First Course in Mathematical Modeling (5th Edition)*, Cengage Learning, Stamford, Conn, USA, 2014.

Research Article

Crop Yield Forecasting Using Artificial Neural Networks: A Comparison between Spatial and Temporal Models

William W. Guo¹ and Heru Xue²

¹ School of Engineering and Technology, Central Queensland University, Rockhampton, QLD 4702, Australia

² College of Computer & Information Engineering, Inner Mongolia Agricultural University, Hohhot, Inner Mongolia 010018, China

Correspondence should be addressed to Heru Xue; xuehr@imau.edu.cn

Received 31 October 2013; Accepted 27 November 2013; Published 23 January 2014

Academic Editor: Chih-Cheng Hung

Copyright © 2014 W. W. Guo and H. Xue. 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.

Our recent study using historic data of wheat yield and associated plantation area, rainfall, and temperature has shown that incorporating statistics and artificial neural networks can produce highly satisfactory forecasting of wheat yield. However, no comparison has been made between the outcomes from the spatial neural network model and commonly used temporal neural network models in crop forecasting. This paper presents the latest research outcomes from using both the spatial and temporal neural network models in crop forecasting. Our simulation shows that the spatial NN model is able to predict the wheat yield with respect to a given plantation area with a high accuracy compared with the temporal NARNN and NARXNN models. However, the high accuracy of the spatial NN model in crop yield forecasting is limited to the forecasting of crop yield only within normal ranges. Users must be cautious when using either NARNN or NARXNN for crop yield forecasting due to their inconsistency between the results of training and forecasting.

1. Introduction

Crop yield forecasting plays an important role in farming planning and management, domestic food supply, international food trade, ecosystem sustainability, and so on [1–3]. For instance, China has the largest population in the world but with limited agricultural land so accurate crop forecasting helps the government provide sufficient food supply to the people. Australia has a small population with vast agricultural land so its concern on crop production is how to optimize revenue from international crop export to countries like China.

There are many factors that have an influence on crop yield, such as plantation area, efficiency of irrigation systems, variations in rainfall and temperature, quality of crop seeds, topographic attributes, soil quality and fertilisation, and disease occurrences [4–8]. Crop growing follows seasonal cycles but many of the factors above are largely irrelevant to the temporal factor. For example, plantation area, rainfall, fertilising, and disease occurrence vary yearly; efficiency of irrigation systems, quality of crop seeds, and soil quality may be improved or degraded from year to year; and topographic

attributes may largely remain the same for a long period of time.

Effort has been made in using either statistics to identify relationships or neural networks to establish mappings between crop yield and some of these factors [4–10]. Our recent study using historic data of wheat yield and associated plantation area, rainfall, and temperature in Queensland, Australia, has shown that incorporating statistics and artificial neural networks can produce a high level of satisfactory forecasting of wheat yield [10]. The neural network employed in this study was a spatial model that treats the wheat plantation areas and yields as mutual mappings, rather than yearly time series. Doubts have been raised about the lack of comparison between the outcomes from this spatial neural network model and commonly used temporal neural network models in crop forecasting. To address this issue, using the wheat yield in Queensland as a reference, this paper presents our research outcomes from using both the spatial and temporal neural network models in crop forecasting. Comparison and discussion are made in terms of their usefulness in crop yield forecasting.

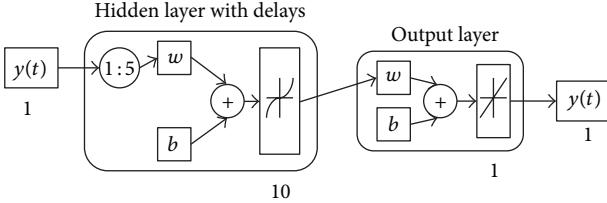


FIGURE 1: Structure of NARNN with 10 hidden neurons and 5 delays.

This is a comparative study between two types of NN-based forecasting models so our strategy is to focus on examining the performances of existing models that have been applied to forecasting, rather than introducing new forecasting models. Readers who are interested in fundamentals of NN forecasting can refer to [9–14] for details.

2. Neural Network-Based Forecasting Models

2.1. Nonlinear Autoregressive Neural Network (NARNN) Model. Nonlinear autoregressive (NAR) is a widely used statistical forecasting model for time series [15, 16]. The forecasting model takes the form as follows:

$$y(t) \approx f(y(t-1), y(t-2), \dots, y(t-d)), \quad (1)$$

where $y(t)$ is the forecasted output and f is an unknown function of d previous known outputs. Traditionally, function f is determined by statistical optimization processes, such as the minimum mean square method.

The feedforward neural network has been used to establish NAR models, in which the traditional function f is replaced by a number of neurons that work together to implicitly approximate the same functionality [11, 17, 18] as

$$\begin{aligned} y(t) &\approx f(y(t-1), y(t-2), \dots, y(t-d)) \\ &= \sum_i b_i \psi \left(\sum_{j=1}^d a_{ji} y(t-j) \right), \end{aligned} \quad (2)$$

where ψ is the transfer functions; a_{ji} denotes the input-to-hidden layer weights at the hidden neuron j ; and b_i is the hidden-to-output layer weight.

This is a time-delay and recurrent neural network model. The input is the known time series which is fed to the hidden layer as input according to the number of time delay. This model is visually illustrated in Figure 1.

2.2. Nonlinear Autoregressive with External Input Neural Network (NARXNN) Model. Nonlinear autoregressive with external input (NARX) is a modified NAR model by including another relevant time series as extra input to the forecasting model [19–21], which can be expressed as

$$\begin{aligned} y(t) &\approx f(x(t-1), x(t-2), \dots, x(t-d), y(t-1), \\ &\quad y(t-2), \dots, y(t-d)), \end{aligned} \quad (3)$$

where $x[t]$ is the external input to the forecasting model with the same number of time delay as $y[t]$.

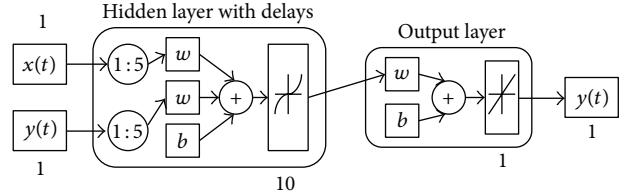


FIGURE 2: Structure of NARXNN with 10 hidden neurons and 5 delays.

Similarly the feedforward neural network is able to establish NARX models, which can be expressed as

$$y(t) \approx \sum_i c_i \psi \left(\sum_{j=1}^d (a_{ji} x(t-j) + b_{ji} y(t-j)) \right), \quad (4)$$

where ψ is the transfer functions; a_{ji} and b_{ji} denote the input-to-hidden layer weights at the hidden neuron j ; and c_i is the hidden-to-output layer weight.

This time-delay recurrent neural network model uses two known time series as independent inputs to the hidden layer according to the same number of time delay. This model is visually illustrated in Figure 2.

2.3. Spatial Feedforward Neural Network Forecasting Model. Multilayer perceptron (MLP) model belongs to feedforward neural networks. In terms of functionality, MLP has no difference from the neural networks used in both NARNN and NARXNN models if the input is time series. Additionally MLPs have been proven to be able to approximate any continuous function by adjusting the number of nodes in the hidden layer [12], with numerous cases of successful applications [13, 14, 22–24]. Figure 3 illustrates the general structure of a three-layer MLP with one hidden layer of L nodes, a p -dimensional input vector \mathbf{X} , and a q -dimensional output vector \mathbf{Y} . The relationship between the input and output components for this MLP can be generally expressed as

$$y_k = \varphi \left(\sum_{j=1}^L b_{kj} \psi \left(\sum a_{ji} x_i \right) \right), \quad (5)$$

where φ and ψ are the transfer functions; a_{ji} denotes the input-to-hidden layer weights at the hidden neuron j ; and b_{kj} is the hidden-to-output layer weights at the output unit k .

There are at least two relevant time series used in the NARXNN model, the internal series $y[t]$ and external series $x[t]$. Time series analysis emphasises on the appearance of consecutive events. However, for example, in crop yield forecasting, the current plantation area should have a much higher impact on the forthcoming crop yield than the historic yields of any past years.

Treating crop yield and plantation area as a correlated pair, MLPs have been used to approach the nonlinear relation that may exist between the two sequences in a correlated “spatial” manner [9, 10], rather than a correlated temporal mode. This has resulted in some encouraging outcomes.

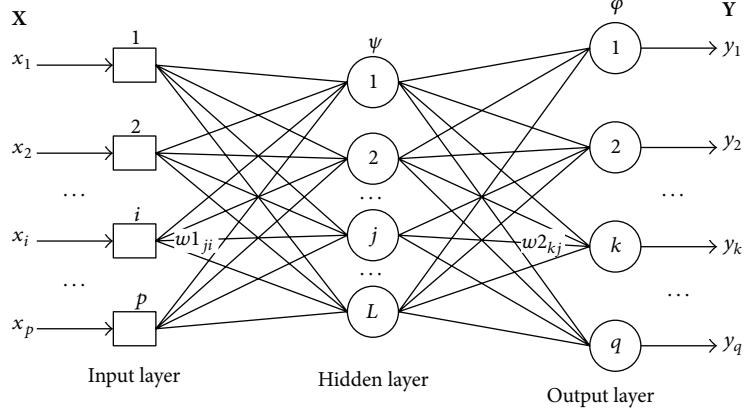


FIGURE 3: Three layer multilayer perceptron (MLP) neural network.

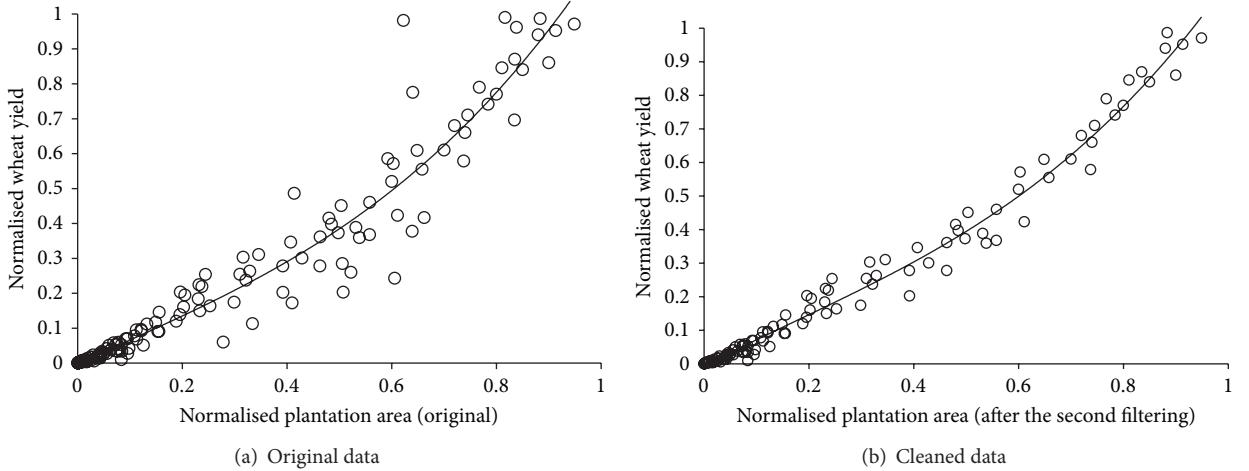


FIGURE 4: Correlations between normalised wheat yield and plantation area.

3. Crop Data for Neural Network Simulation

3.1. Historic Crop Data. The Queensland historic wheat plantation area in hectare and wheat yield in ton from 1861 to 2007 are extracted from the report of Australian Bureau of Statistics [25], which gives a total of 135 entries over the past 147 years. Both plantation area in hectare and wheat yield in ton are listed in their approximated absolute values each year in the original data. We normalise both factors with their ceilings in the order of millions. The ceiling for plantation area is 13 million hectares and that for wheat yield is 20 million tons. Plot of these two normalised factors is shown in Figure 4(a). After two rounds of outlier detection and exclusion, a third-order polynomial correlation has been defined as

$$w = 0.8197a^3 - 0.5102a^2 + 0.8511a - 0.0073, \quad (6)$$

where w represents the normalised annual wheat yield and a is the normalised plantation area. This correlation fits the filtered data well (Figure 4(b)) with a coefficient of 0.9904. This nonlinear correlation indicates that, through properly training, a neural network system can be used to approach

such nonlinear relation between the crop production and plantation area.

3.2. Data for Training and Testing Neural Network Models. Neural network training requires a sufficient amount of data for achieving a high reliability. For MLPs, since temporal factor does not play any role in correlation (6), this correlation can be used to generate more data without changing the general trend. By keeping the same pattern, a moving window operator with different sizes is repeatedly applied to these cleaned data so as to generate more entries to fill the gaps where the original entries are scarce. The final data to train and test the selected neural networks are compiled by mixing the cleaned and regenerated entries together.

For both NARNN and NARXNN models, such data expansion cannot be applied because the training data must be a sequence ordered by time. Among the 135 datasets, the first three are with an interval of 5 years and thus excluded from both NARNN and NARXNN training and testing. Time series data for NARNN model is the normalised annual wheat yield by the corresponding plantation area because NARNN

TABLE 1: Statistical results of neural network training and testing.

	N	Training		Testing	
		MSE	R	N	MSE
MLP (100)	329	0.0001	0.9996	40	0.0001
MLP (200)	329	0.0001	0.9998	40	0.0001
NARNN (5 d)	105	0.1035	0.9682	20	0.0231
NARNN (10 d)	105	0.0069	0.9753	20	0.0188
NARXNN (5 d)	105	0.0174	0.908	20	0.0528
NARXNN (10 d)	105	0.0271	0.8574	20	0.0628

takes only one time series as the input. To some extent, this normalised series actually absorbs the effect of plantation area on crop yield into the forecasting. For NARXNNs, the normalised annual wheat yield by wheat ceiling is the internal input $y[t]$ and the normalised plantation area by plantation ceiling is the external input $x[t]$. If the normalised wheat yield by plantation area is used as the internal input $y[t]$, the effect of plantation area on wheat yield will be doubly accounted in the forecasting.

4. Results of Neural Network Simulation

Two MLP models are used for training and simulation. By running the process ten times with both the 100-node hidden-layer MLP and 200-node hidden-layer MLP, the simulations produce a highly satisfactory average outcome for both training and testing (Table 1). The difference is that the latter achieves a slightly lower MSE and a higher correlation than the former but both show a high consistency between the results of training and simulation or testing.

Two NARNN models are also used in training and simulation. Both models have ten hidden neurons but with 5 delays and 10 delays, respectively. Running ten repetitions for each model has resulted in a fairly satisfactory outcome on average shown in Table 1. Since the model changes data partition for training and testing dynamically between separate runs, a highly satisfactory outcome from training does not always produce a highly satisfactory outcome from testing. In general, the result of simulation is always inferior to that of training, with the 10-delay model performing slightly better than the 5-delay model. Similar trends are found from the results of training and testing the two NARXNN models, whose performance is even worse than that of NARNNs (Table 1).

5. Discussion

In terms of consistency between the performances of training and testing using the same model, MLPs are able to achieve the highest consistency and hence produce the best simulation results among the three forecasting models. This is mainly because the data used to train MLPs have been subject to outlier cleaning, which means the abnormal wheat yields outside the statistical trend have no impact on the training and testing. In addition, without the temporal constraint, the expanded dataset ensures that the MLPs are adequately

trained and tested with multiple crossing validations. Since the original data have been cleaned, in theory the MLPs should only be effective for crop forecasting of any “normal” year.

NARNNs exhibit a highly satisfactory performance in training but the simulation is highly dependent on the selection of testing dataset; hence, the range of forecasting error is large. This indicates that a well trained NARNN model is not able to produce consistently accurate forecasting. This inconsistency between the training and testing is clearly illustrated in Figure 5. Our experiments also show that changes in number of hidden neuron and length of delay (>3) for NARNN do not make significant improvement to the performance of forecasting. Although the NARNNs are not consistent in forecasting, they use the whole data without excluding “abnormal” datasets in both training and testing. This is a complement to MLPs to some extent.

NARXNNs exhibit similar inconsistent patterns between training and testing but even worse than NARNNs (Figure 6). This may be caused by the double impacts on forecasting exerted by the “anomalies” in both the wheat yield series and plantation area series, which were not excluded through data cleaning.

Data cleaning for both NARNN and NARXNN is very challenging since both models use time series as the input. Excluding some temporal events will leave irregular gaps in the time series, which in turn influences the training and testing. The other possible reason that contributed to the inconsistency between the training and testing of both NARNN and NARXNN may be the inadequacy of the historic data in this case. Since we cannot artificially create extra yearly crop yields, like using interpolation to generate extra spatial datasets [9, 26], using time series based NN models to forecast crop yield may be immature at this stage.

6. Conclusion

The spatial NN model is able to predict the wheat yield with respect to a given plantation area with a high accuracy, compared with the temporal NN models such as NARNN and NARXNN. However, the high accuracy of the spatial NN model in crop yield forecasting is only applicable to the forecasting of crop yield within normal ranges because the model is trained using the cleaned and expanded data following a third-order polynomial trend between the crop yield and

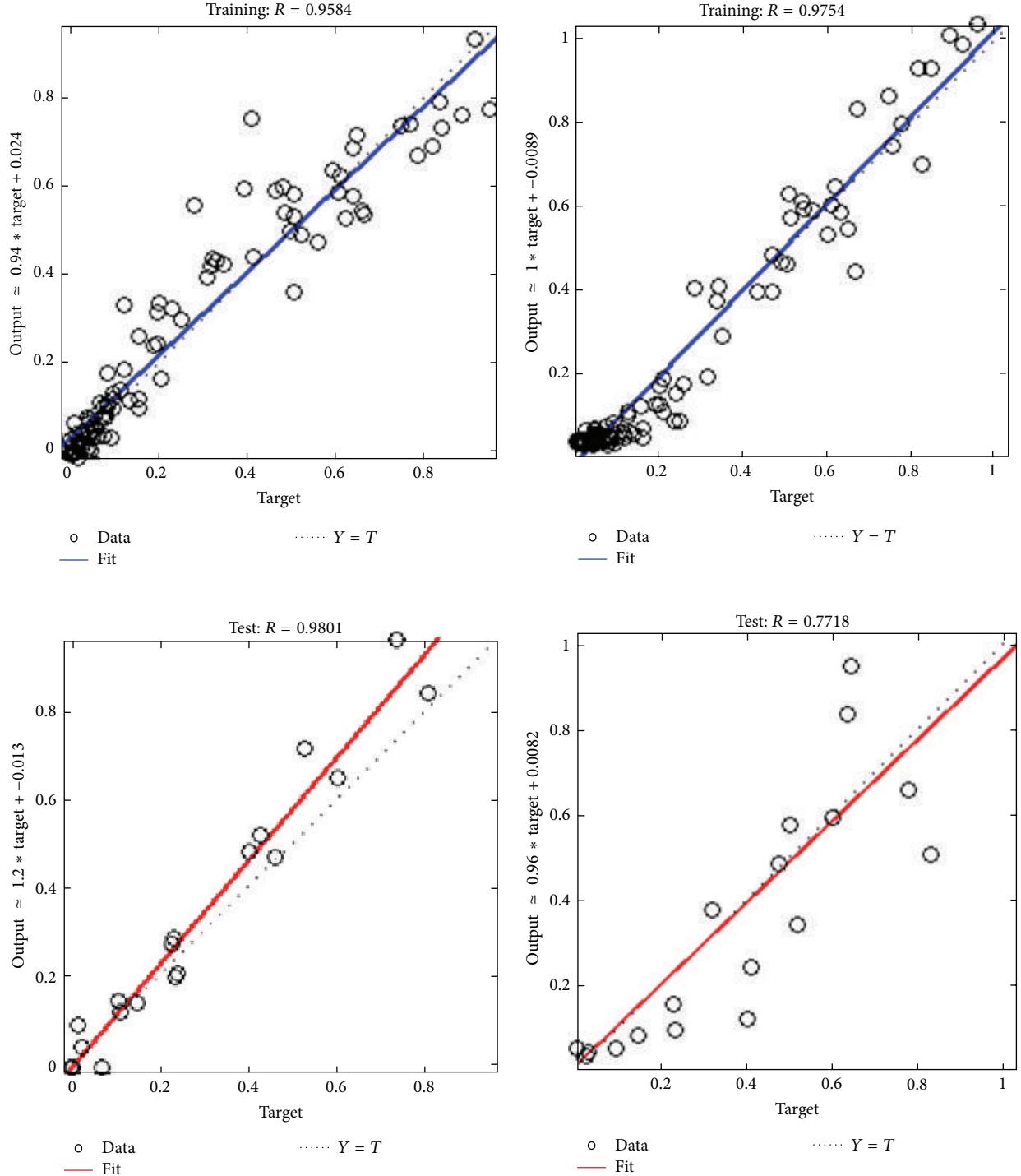


FIGURE 5: Linear regression for two separate training and testing runs of NARNN.

plantation area. NARNNs may be used as a complementary means to the MLPs due to its usage of the whole data. Users must be cautious when using either NARNN or NARXNN for crop yield forecasting due to the inconsistency between training and forecasting.

In the future, other factors, such as efficiency of irrigation systems, variations in rainfall and temperature, quality of crop seeds, topographic attributes, soil quality and fertilisation, and disease occurrences, should be incorporated in forecasting model building and simulation.

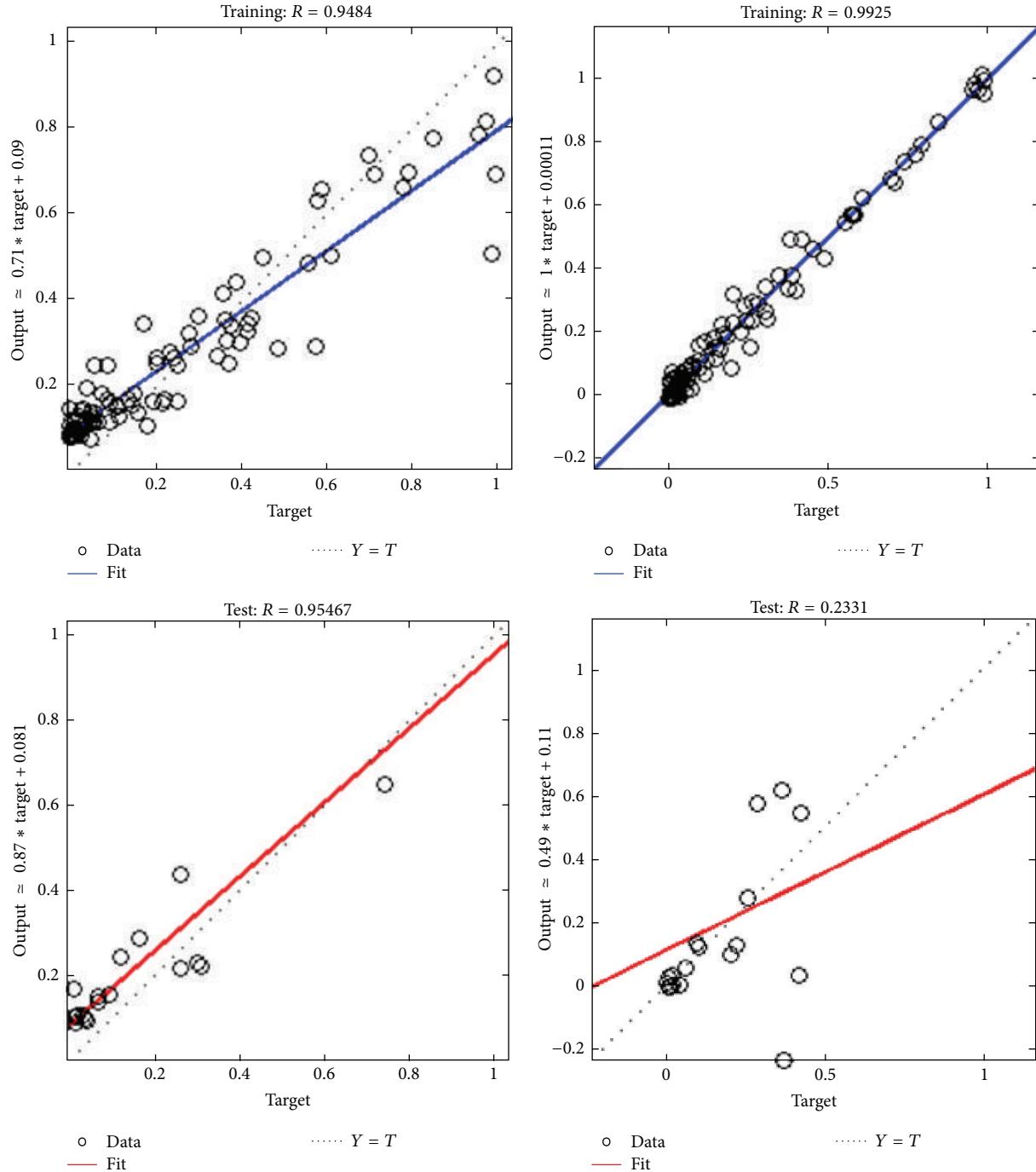


FIGURE 6: Linear regression for two separate training and testing runs of NARXNN.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] Y. W. Jame and H. W. Cutforth, "Crop growth models for decision support systems," *Canadian Journal of Plant Science*, vol. 76, no. 1, pp. 9–19, 1996.
- [2] J. I. Yun, "Predicting regional rice production in South Korea using spatial data and crop-growth modeling," *Agricultural Systems*, vol. 77, no. 1, pp. 23–38, 2003.
- [3] F. Tao, M. Yokozawa, Z. Zhang, Y. Xu, and Y. Hayashi, "Remote sensing of crop production in China by production efficiency models: models comparisons, estimates and uncertainties," *Ecological Modelling*, vol. 183, no. 4, pp. 385–396, 2005.
- [4] G. Hoogenboom, "Contribution of agrometeorology to the simulation of crop production and its applications," *Agricultural and Forest Meteorology*, vol. 103, no. 1-2, pp. 137–157, 2000.

- [5] Y. He, Y. Zhang, S. Zhang, and H. Fang, "Application of artificial neural network on relationship analysis between wheat yield and soil nutrients," in *Proceedings of the IEEE 27th Annual Conference on Engineering in Medicine and Biology*, pp. 4530–4533, Shanghai, China, September 2005.
- [6] T. R. Green, J. D. Salas, A. Martinez, and R. H. Erskine, "Relating crop yield to topographic attributes using spatial analysis neural networks and regression," *Geoderma*, vol. 139, no. 1-2, pp. 23–37, 2007.
- [7] E. de Jong and D. A. Rennie, "Effect of soil profile type and fertilizer on moisture use by wheat grown on fallow or stubble land," *Canadian Journal of Soil Science*, vol. 49, no. 2, pp. 189–197, 1969.
- [8] C. A. Campbell, R. P. Zentner, and P. J. Johnson, "Effect of crop rotation and fertilization on the quantitative relationship between spring wheat yield and moisture use in southwestern Saskatchewan," *Canadian Journal of Soil Science*, vol. 68, no. 1, pp. 1–16, 1988.
- [9] W. W. Guo, L. D. Li, and G. Whymark, "Simulating wheat yield in New South Wales of Australia using interpolation and neural networks," in *Neural Information Processing, Models and Applications*, vol. 6444 of *Lecture Notes in Computer Science*, pp. 708–715, Springer, Berlin, Germany, 2010.
- [10] W. W. Guo and H. Xue, "An incorporative statistic and neural approach for crop yield modelling and forecasting," *Neural Computing and Applications*, vol. 21, no. 1, pp. 109–117, 2012.
- [11] F. Liang, "Bayesian neural networks for nonlinear time series forecasting," *Statistics and Computing*, vol. 15, no. 1, pp. 13–29, 2005.
- [12] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [13] W. W. Guo, M. M. Li, G. Whymark, and Z.-X. Li, "Mutual complement between statistical and neural network approaches for rock magnetism data analysis," *Expert Systems with Applications*, vol. 36, no. 6, pp. 9678–9682, 2009.
- [14] M. M. Li, W. Guo, B. Verma, K. Tickle, and J. O'Connor, "Intelligent methods for solving inverse problems of backscattering spectra with noise: a comparison between neural networks and simulated annealing," *Neural Computing and Applications*, vol. 18, no. 5, pp. 423–430, 2009.
- [15] S. Y. Hwang and I. V. Basawa, "Large sample inference based on multiple observations from nonlinear autoregressive processes," *Stochastic Processes and their Applications*, vol. 49, no. 1, pp. 127–140, 1994.
- [16] G. Kapetanios, "Nonlinear autoregressive models and long memory," *Economics Letters*, vol. 91, no. 3, pp. 360–368, 2006.
- [17] T. Taskaya-Temizel and M. Casey, "A comparative study of autoregressive neural network hybrids," *Neural Networks*, vol. 18, no. 5-6, pp. 781–789, 2005.
- [18] W. Pawlus, H. R. Karimi, and K. G. Robbersmyr, "Data-based modeling of vehicle collisions by nonlinear autoregressive model and feedforward neural network," *Information Sciences*, vol. 35, pp. 65–79, 2012.
- [19] S. Chatterjee, S. Nigam, J. B. Singh, and L. N. Upadhyaya, "Software fault prediction using Nonlinear Autoregressive with eXogenous Inputs (NARX) network," *Applied Intelligence*, vol. 37, no. 1, pp. 121–129, 2012.
- [20] S. H. Arbain and A. Wibowo, "Neural networks based nonlinear time series regression for water level forecasting of Dungun River," *Journal of Computer Science*, vol. 8, no. 9, pp. 1506–1513, 2012.
- [21] F. Huo and A.-N. Poo, "Nonlinear autoregressive network with exogenous inputs based contour error reduction in CNC machines," *International Journal of Machine Tools and Manufacture*, vol. 67, pp. 45–52, 2013.
- [22] W. W. Guo, "Incorporating statistical and neural network approaches for student course satisfaction analysis and prediction," *Expert Systems with Applications*, vol. 37, no. 4, pp. 3358–3365, 2010.
- [23] E. Bojórquez, J. Bojórquez, S. E. Ruiz, and A. Reyes-Salazar, "Prediction of inelastic response spectra using artificial neural networks," *Mathematical Problems in Engineering*, vol. 2012, Article ID 937480, 5 pages, 2012.
- [24] Y. E. Shao, "Prediction of currency volume issued in Taiwan using a hybrid artificial neural network and multiple regression approach," *Mathematical Problems in Engineering*, vol. 2013, Article ID 676742, 9 pages, 2013.
- [25] Australian Bureau of Statistics, *Agricultural Commodities—Historical Data*, Commonwealth of Australia, Canberra, Australia, 2008.
- [26] W. W. Guo, "A novel application of neural networks for instant iron-ore grade estimation," *Expert Systems with Applications*, vol. 37, no. 12, pp. 8729–8735, 2010.

Research Article

Design and Implementation of Fuzzy Logic Controller for Online Computer Controlled Steering System for Navigation of a Teleoperated Agricultural Vehicle

Prema Kannan,¹ Senthil Kumar Natarajan,² and Subhransu Sekhar Dash³

¹ Department of Electronics and Instrumentation Engineering, SRM University, Chennai, Tamil Nadu 603 203, India

² Department of Electrical and Electronics Engineering, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu 626 005, India

³ Department of Electrical and Electronics Engineering, SRM University, Chennai, Tamil Nadu 603 203, India

Correspondence should be addressed to Prema Kannan; prema.k@ktr.srmuniv.ac.in

Received 12 July 2013; Revised 28 November 2013; Accepted 3 December 2013

Academic Editor: Siddhivinayak Kulkarni

Copyright © 2013 Prema Kannan 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 describes design, modeling, simulation, control, and implementation of teleoperated agricultural vehicle using intelligent technique. This vehicle can be used for ploughing, sowing, and soil moisture sensing. Online computer controlled steering system for a vehicle utilizing two independent drive wheels can be used to avoid obstacles and to improve the ability to resist external side forces. To control the steer angles of the nondriven wheels, the mathematical relationships between the drive wheel speeds and the steer angles of the nondriven wheels are used. A fuzzy logic controller is designed to change the drive wheel speeds and to achieve the desired steer angles. Online control of the agricultural vehicle is achieved from a remote place by means of Web Publishing Tool in LabVIEW. IR sensors in the vehicle are used to detect and to avoid the obstacles around. The developed steering angle control algorithm and fuzzy logic controller have been implemented in an agricultural vehicle which depicts that the vehicle performs its operation efficiently and reduces the manpower and becomes advantageous.

1. Introduction

The diminishing number of agricultural wages in India is a momentous issue for Indian agriculture. Robot tractors have been introduced to replace human labors for field work and for improving work efficiency. However, robot tractor needs safety devices to ensure safe operation when it is not operated by a human. Therefore, the safety issue is an important research topic for agricultural robots when utilized on a farm [1].

Numerous studies on the automation of agricultural machines, particularly on a solution of this problem, have been carried out. Autonomous navigation and vehicle guidance, a global positioning system (GPS), a gyroscope, machine vision, and other sensors have been used in most of those studies [2–4].

Sgorbissa and Zaccaria [5] have proposed a navigation subsystem of a mobile robot which operates in human environments to carry out different tasks, such as transporting

waste in hospitals or escorting people in exhibitions. This approach integrates prior knowledge of the environment with local perceptions to carry out the given tasks efficiently and safely. They had discussed the properties of their approach and experimental results recorded during real-world experiments.

Murakami et al., [6] have presented a teleoperation system for a hydrostatic transmission drive crawler type robotic vehicle. Their system was developed to satisfy the needs of various farm operations and teleoperation in unknown agricultural fields.

There are two vital issues to be resolved for developing a safety device for a teleoperated agricultural vehicle; one is to detect an obstacle surrounding the vehicle and the other one is to determine the location of the obstacle. The performance of various sensors for detecting obstacles has been investigated in recent studies to improve the safety of an autonomous vehicle or assist human driving. Noguchi et al. [7] have introduced a master-slave robot system in which a

wireless local area network was utilized to broadcast the GPS position of a slave robot to a master robot, so that the master and slave robots could work together without collision of two robot tractors but could not detect other objects except the robot. In order to identify general obstacles, radar and stereo vision were used in some studies for automobiles [8, 9]. A laser scanner was used to identify and locate other vehicles working in the same field with the robot [10]. In addition, ultrasonic sensors [11] and a bumper switch were mounted in front of robot tractor as a safety system. In those previous studies, a laser scanner, radar, and an ultrasonic sensor were used as a range finder because these sensors can determine distance to the obstacle with a high accuracy.

Knudson and Turner [12] have evaluated reactive and learning navigation algorithms for exploration robots that avoid obstacles and reach specific destinations in limited time and with limited observations. This method uses neuroevolutionary based navigation. Neuroevolutionary approach is policy search method where control is achieved through a search across policies. This search through a population of policies allows the discovery of new and robust control strategies.

Direct current (DC) motors are broadly used in conveyors, textile mills, paper mills, position control, and robotic manipulators because they are reliable for an extensive range of operating conditions and their control is fairly simple. Usually DC motors are modeled as linear systems and controlled by linear control approaches. However, most linear controllers give unsatisfactory performance due to changes in the control parameters, loading conditions, and nonlinearities introduced by the armature reaction. When these nonlinearities of the motor are known functions, then adaptive tracking control method can be used [13, 14]. If these nonlinearities of the motor are unknown, neural or fuzzy control is suitable for ensuing satisfactory performance of the closed loop system [15–18]. Senthil Kumar et al., [19] have designed an artificial neuron controller for chopper fed embedded DC drives. The designed neuron controller reduces the steady state error, overshoot, and settling time.

The objective of this work is to develop a teleoperated agricultural vehicle which can be used for ploughing, sowing, and soil moisture sensing. Online computer controlled steering system for the vehicle is described and a fuzzy logic controller is designed to achieve steering control based on obstacle and boundary information of the agricultural land. The mathematical relationships between the drive wheel speeds and the steer angles of the nondriven wheels are used to calculate and control the steer angles. The developed fuzzy logic controller controls the speeds of DC motors connected with rear wheels of the agricultural vehicle to achieve the desired steer angle. Using the proposed approach, a teleoperated agricultural vehicle is controlled from a remote place. Infrared (IR) sensors are employed to detect the obstacle. The status of IR sensor helps path planning and changing the direction of the vehicle. Data transmission between the agricultural vehicle and the remote client is achieved by means of Web Publishing Tool. To accomplish this goal, a soil moisture sensor, a stepper motor, and

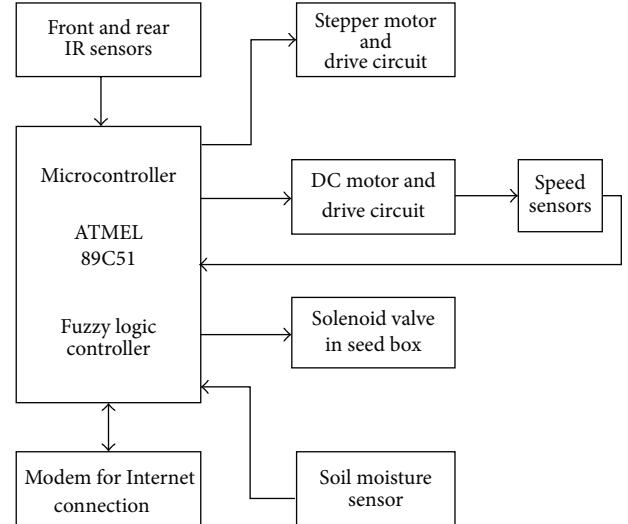


FIGURE 1: Hardware structure of the proposed system.

a solenoid valve are used for measuring soil moisture, to control ploughing tool and seed box open and close operation, respectively.

2. Architecture of the Proposed System

Figure 1 shows the hardware structure of the proposed system. The system consists of DC motors, stepper motor, soil moisture sensor, IR sensors, and a solenoid valve. Soil moisture sensor is used to measure the moisture content of the land. The obstacles around the vehicle are detected using IR sensors. The seed box open and close operation is controlled by solenoid valve during sowing operation. Stepper motor is used to control the up and down movement of ploughing tool. The depth of ploughing can also be controlled by this stepper motor by varying its step angle.

The steer angles of the nondriven wheels are controlled by two DC motors connected with the rear driven wheels of the vehicle. Based on the desired steering angle values, the desired speed values of the DC motors are calculated by using the developed algorithm. The speeds of the DC motors are controlled to the desired value by the designed fuzzy logic controller.

The length and width of the agricultural land are measured. The time taken by the vehicle to travel a particular length is determined. Based on this determined time value and the length of the land, the time taken by the vehicle to reach one end of the land to the other end is calculated. The timer value is set to this calculated time value to navigate the vehicle in forward direction. For this calculated time period, the vehicle moves in forward direction, subsequently turns left, and moves in forward direction for the same period and then it turns right and moves forward. Thus by using the time taken by the vehicle to travel the length once, wheel track, and the width of the land, its navigation and steering control actions are repeated.

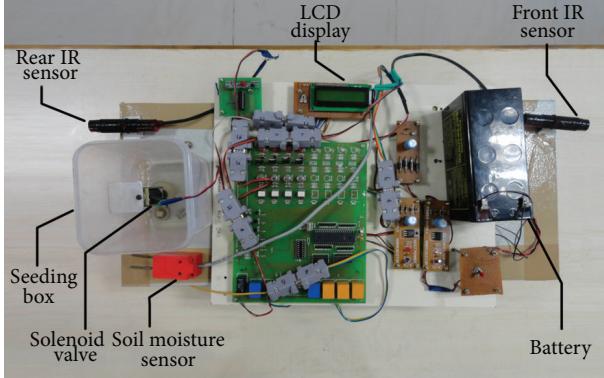


FIGURE 2: Top view of hardware prototype of the developed teleoperated agricultural vehicle.

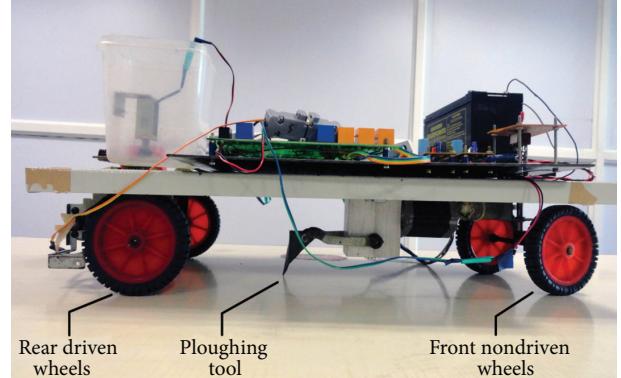


FIGURE 3: Elevation of hardware prototype of the developed teleoperated agricultural vehicle.

3. Experimental Setup

The hardware prototype set-up of the teleoperated agricultural vehicle is shown in Figures 2 and 3. The developed fuzzy logic controller and the control algorithm for ploughing action and solenoid valve were implemented using ATMEL 89C51 microcontroller. A two-way normally closed electromechanical solenoid valve is used to control the flow of seeds. The solenoid valve is energized by a normally closed relay. When the relay is energized, it comes to normally open position. In this relay position, the solenoid valve is energized and it opens to drop the seeds. By increasing or decreasing the frequency of energization of relay circuit, the distance between the seeds can be increased or decreased.

A 12 V stepper motor is used to adjust the position of ploughing tool. By controlling the stepping angle of the stepper motor, the ploughing tool position is being controlled. The navigation speed and steering angle control are achieved by two 12 V DC motors connected with two independent drive wheels. The stepper motor and DC motors are supplied by a 12 V, 7 Ah sealed lead acid battery. A soil moisture sensor is used to measure the moisture content of the soil. The measured value is displayed using LCD display.

4. Fuzzy Logic Controller

General proportional integral (PI) and proportional integral derivative (PID) controllers are extensively used for motor control applications. But they do not give satisfactory results when control parameters, loading conditions, and the motor itself are changed. But fuzzy logic controller can be designed without the exact model of the system. This approach of fuzzy logic controller (FLC) design guarantees the stable operation even if there is a change in the parameters and the motor [20].

Fuzzy logic control is derived from fuzzy set theory introduced by Zadeh in 1965. In fuzzy set theory, the transition between membership and nonmembership can be gradual. Therefore, boundaries of fuzzy sets can be vague and ambiguous, making it useful for approximate systems. Fuzzy logic controller (FLC) is an attractive choice when precise mathematical formulations are not possible [21].

The designed fuzzy logic controller has two inputs and one output. The inputs are error value and change in error

TABLE 1: Rule table.

Error	Change in error								
	NVB	NB	NM	NS	Z	PS	PM	PB	PVB
NVB	NVB	NVB	NVB	NVB	NVB	NB	NM	NS	Z
NB	NVB	NVB	NVB	NVB	NB	NM	NS	Z	PS
NM	NVB	NVB	NVB	NB	NM	NS	Z	PS	PM
NS	NVB	NVB	NB	NM	NS	Z	PS	PM	PB
Z	NVB	NB	NM	NS	Z	PS	PM	PB	PVB
PS	NB	NM	NS	Z	PS	PM	PB	PVB	PVB
PM	NM	NS	Z	PS	PM	PB	PVB	PVB	PVB
PB	NS	Z	PS	PM	PB	PVB	PVB	PVB	PVB
PVB	Z	PS	PM	PB	PVB	PVB	PVB	PVB	PVB

and the output is duty cycle. The seven linguistic variables used for “error” and “change in error” are NVB (negative very big), NB (negative big), NM (negative medium), NS (negative small), Z (zero), PS (positive small), PM (positive medium), PB (positive big), and PVB (positive very big). The duty cycle output uses seven linguistic variables like NVB (negative very big), negative big (NB), negative medium (NM), negative small (NS), zero (Z), positive small (PS), positive medium (PM), PB (positive big), and PVB (positive very big). The rule table for the designed fuzzy logic controller is given in Table 1. From the rule table, the rules are manipulated as follows.

If error is NS and change in error is PB, then output is PM.

The symbolic expression of k th rule of the designed fuzzy logic controller is given as follows.

$$\text{If } e \text{ is } LE^{(k)} \text{ and } \Delta e \text{ is } L\Delta E^{(k)}, \text{ then } \Delta u \text{ is } L\Delta U^{(k)}, \quad (1)$$

where $LE^{(k)}$, $L\Delta E^{(k)}$, and $L\Delta U^{(k)}$ are linguistic values from term sets of error, change in error, and change in output duty cycle, respectively.

The meaning of the above defined k th rule in terms of mamdani type implication is given as a fuzzy relation $R^{(k)}$ as follows

$$\mu_R^{(k)}(e, \Delta e, \Delta u) = \mu_{LE}^{(k)}(e) \Lambda \mu_{L\Delta E}^{(k)}(\Delta e) \Lambda \mu_{L\Delta U}^{(k)}(\Delta u). \quad (2)$$

The overall conclusion by combining the outputs of all the fuzzy rules can be written as

$$\begin{aligned}\mu_R(e, \Delta e, \Delta u) &= \mu_R^{(1)}(e, \Delta e, \Delta u) V \mu_R^{(2)}(e, \Delta e, \Delta u) \\ &\cdots V \mu_R^{(k)}(e, \Delta e, \Delta u).\end{aligned}\quad (3)$$

The value of $\mu_R^{(k)}$ for each value of k is defined in (2). The crisp value of change in duty cycle is computed using centre of area method as follows:

$$\text{Duty cycle output } u^* = \frac{\int u \cdot \mu_U(u) \cdot du}{\int \mu_U(u) \cdot du}, \quad (4)$$

where u^* is defuzzified duty cycle, u is duty cycle output, and $\mu_U(u)$ is Union of the clipped control outputs.

5. Mathematical Model for Steering System

Figure 4 represents kinematic model of the proposed agricultural vehicle with a primary steering system using two independently driven wheels. ω_L and ω_R are the wheel speeds of the rear left and rear right drive wheels, respectively. Φ_L and Φ_R are the steer angles of the front left and front right wheels, respectively. b is the wheel base. t_B is the track of the rear wheels and t_F is the track of the front wheels. R is the radius of the turning circle of the vehicle (radius of curvature) [22].

The rate of rotation of the drive axle about the centre of curvature is given by

$$\dot{\theta} = \frac{V_L}{R + (t_B/2)} = \frac{V_R}{R - (t_B/2)}, \quad (5)$$

where R is radius of the turning circle of the vehicle (radius of curvature),

V_L is translational velocity of the left drive wheel, V_R is translational velocity of the right drive wheel, t_B is the track of the rear wheels, and t_F is the track of the front wheels.

If longitudinal slip is considered, the translational velocities become

$$\begin{aligned}V_L &= r\omega_L(1 - i_L), \\ V_R &= r\omega_R(1 - i_R),\end{aligned}\quad (6)$$

where r is radius of a drive wheel, and i is longitudinal slip

When we substitute (6) in (5)

The centre of curvature

$$\dot{\theta} = \frac{r\omega_L(1 - i_L)}{R + t_B/2} = \frac{r\omega_R(1 - i_R)}{R - t_B/2}. \quad (7)$$

When the longitudinal slips are equal for left and right drive wheels, as well as equal radius for the drive wheels, then

$$\dot{\theta} = \frac{\omega_L}{R + t_B/2} = \frac{\omega_R}{R - t_B/2}. \quad (8)$$

Equal longitudinal slips will occur where traction conditions are the same and under these conditions, slip angles will not be significant.

For the front left steerable wheel,

$$\text{Steering angle } \Phi_L = \tan^{-1} \frac{2b(\omega_L - \omega_R)}{\omega_L(t_B + t_F) + \omega_R(t_B - t_F)}. \quad (9)$$

Similarly, for the front right steerable wheel,

$$\text{Steering angle } \Phi_R = \tan^{-1} \frac{2b(\omega_L - \omega_R)}{\omega_L(t_B - t_F) + \omega_R(t_B + t_F)}. \quad (10)$$

If $t_B = t_F$, (9) and (10) will be simplified as given in (11), and (12), respectively,

$$\Phi_L = \tan^{-1} \left[\frac{b}{t} \left(1 - \frac{\omega_R}{\omega_L} \right) \right], \quad (11)$$

$$\Phi_R = \tan^{-1} \left[\frac{b}{t} \left(\frac{\omega_L}{\omega_R} - 1 \right) \right]. \quad (12)$$

6. Calculation of Drive Wheel Speed for Steering Control

If both drive wheels turn with same speed, the robot moves in a straight line. If one wheel rotates faster than the other, the robot follows a curved path inward towards the slower wheel. If the wheels turn at equal speed, but in opposite directions, the robot spins around. Thus, steering the robot is just a matter of varying the speeds of the drive wheels. These drive wheel speeds are calculated from the steer angles Φ_L and Φ_R .

6.1. Calculation of Drive Wheels Speed to Turn the Vehicle Left. To turn the vehicle left, the left wheel speed should be less than the right wheel speed. Hence the left wheel speed is assigned with some low speed value and the expression for desired right wheel speed ω_{DR} is derived from the steering angle Φ_L of left steerable wheel.

Let $\omega_L = 1 \text{ rpm}$; then, (9) will be

$$\tan \Phi_L = \frac{2b(1 - \omega_R)}{(t_B + t_F) + \omega_R(t_B - t_F)}. \quad (13)$$

From (13), the desired right wheel speed can be written as

$$\omega_{DR} = \frac{2b - \tan \Phi_L(t_B + t_F)}{2b + \tan \Phi_L(t_B - t_F)}. \quad (14)$$

6.2. Calculation of Drive Wheels Speed to Turn the Vehicle Right. To turn the vehicle right, the right wheel speed should be less than the left wheel speed. Hence the right wheel speed is assigned with some low speed value and the expression for desired left wheel speed ω_{DL} is derived from the steering angle Φ_R of right steerable wheel.

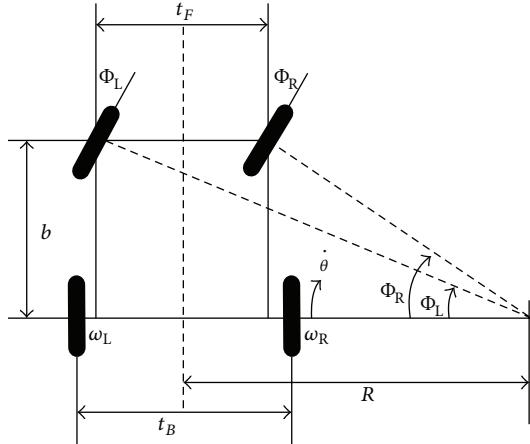


FIGURE 4: Kinematic model of the proposed agricultural vehicle.

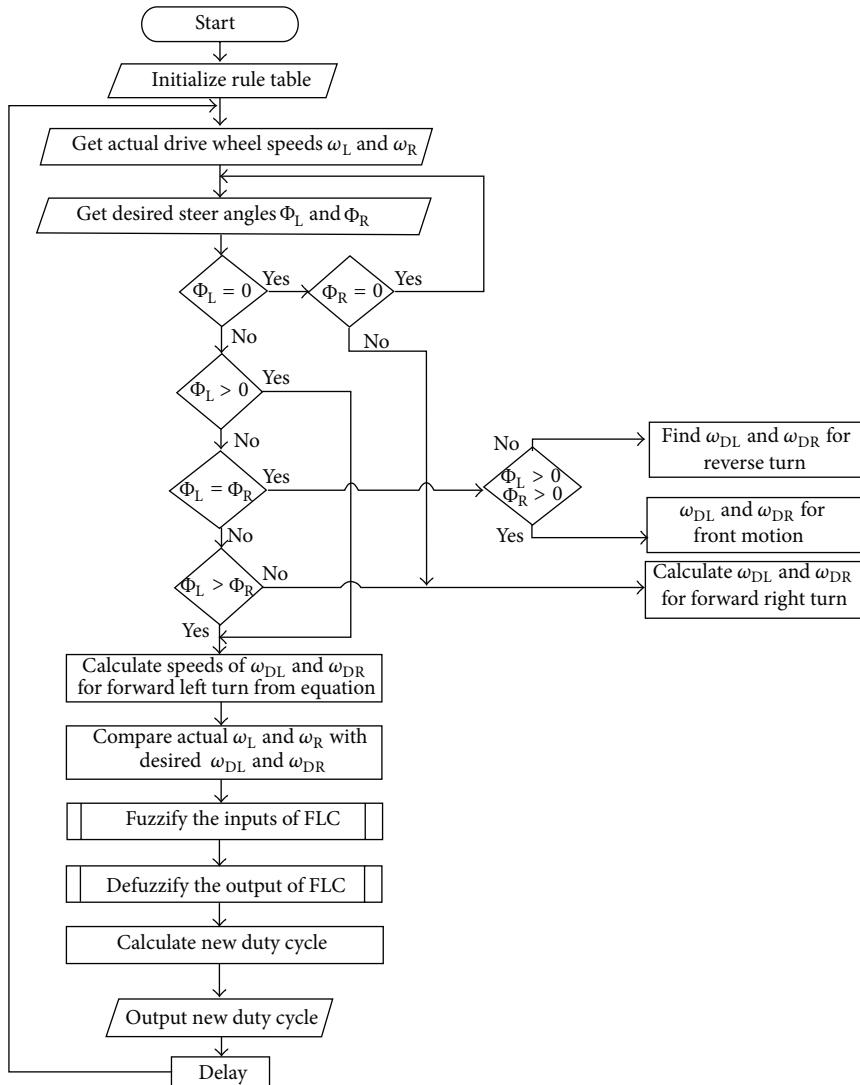


FIGURE 5: Flow chart of the control algorithm.

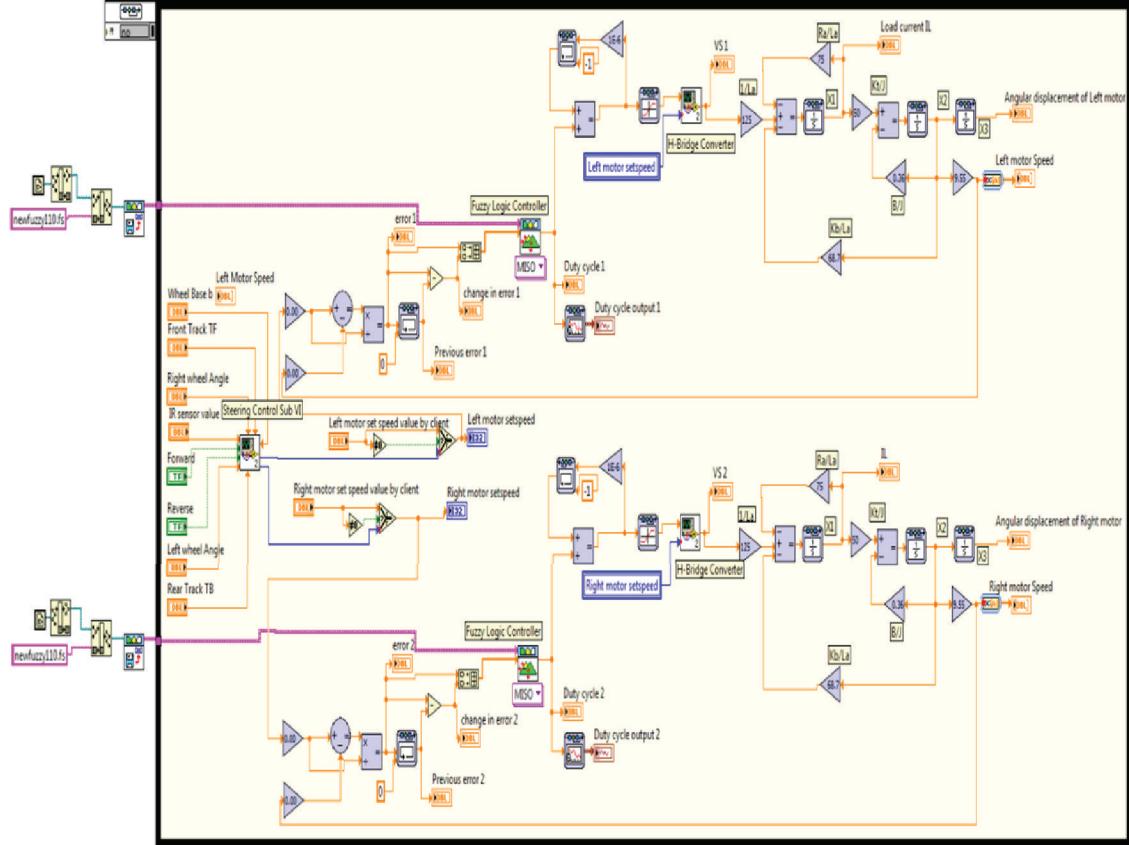


FIGURE 6: Block diagram of the simulated fuzzy logic control system for a teleoperated agricultural vehicle.

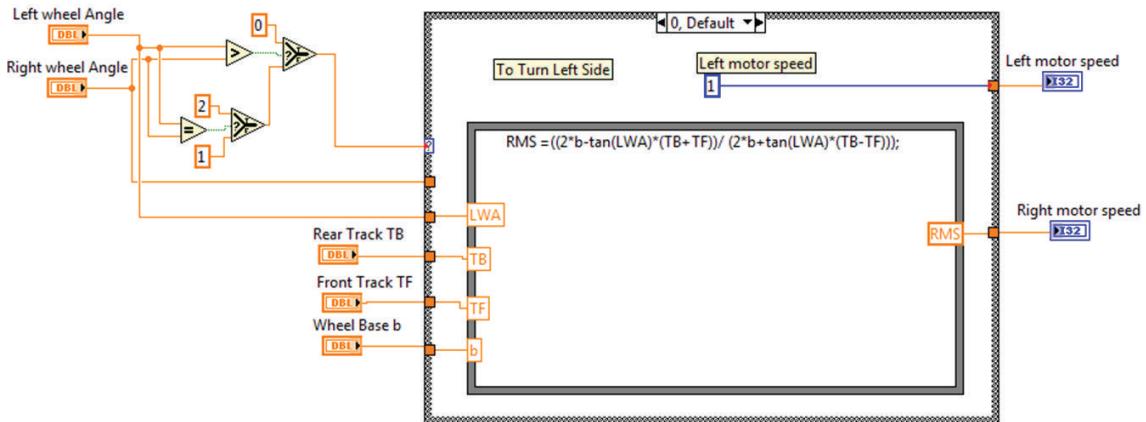


FIGURE 7: Block diagram of the simulated steering angle control system to turn the vehicle left.

Let $\omega_R = 1$ rpm and then (10) will be

$$\tan \Phi_R = \frac{2b(\omega_L - 1)}{\omega_L(t_B - t_F) + (t_B + t_F)}. \quad (15)$$

From (15), the desired right wheel speed can be written as

$$\omega_{DL} = \frac{2b + \tan \Phi_R(t_B + t_F)}{2b - \tan \Phi_R(t_B - t_F)}. \quad (16)$$

7. Computer Simulation and Teleoperation

The simulation of online computer controlled steering system for a teleoperated agricultural vehicle is done based on equation modeling using LabVIEW. Flow chart of the proposed control algorithm is given in Figure 5. The simulated models are shown in Figures 6, 7, and 8. The simulation of computer controlled steering system was done with a fuzzy logic controller.

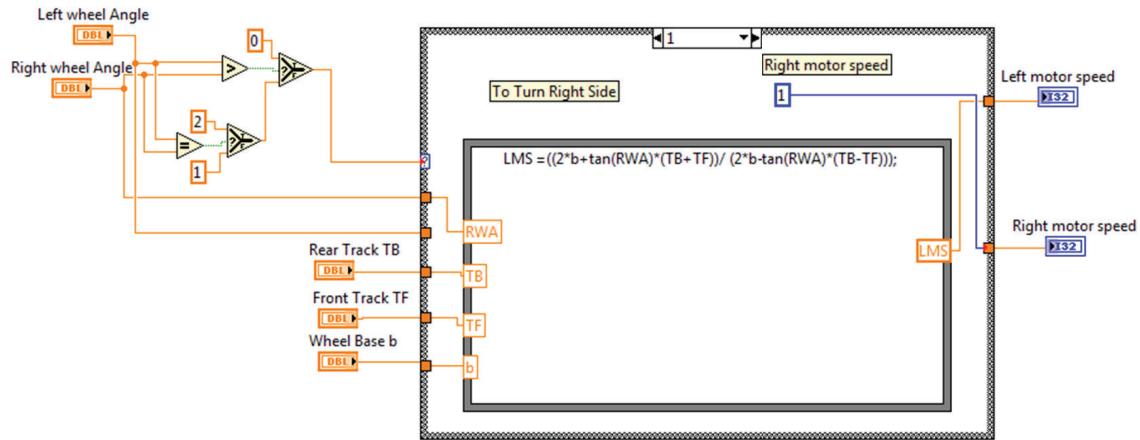


FIGURE 8: Block diagram of the simulated steering angle control system to turn the vehicle right.

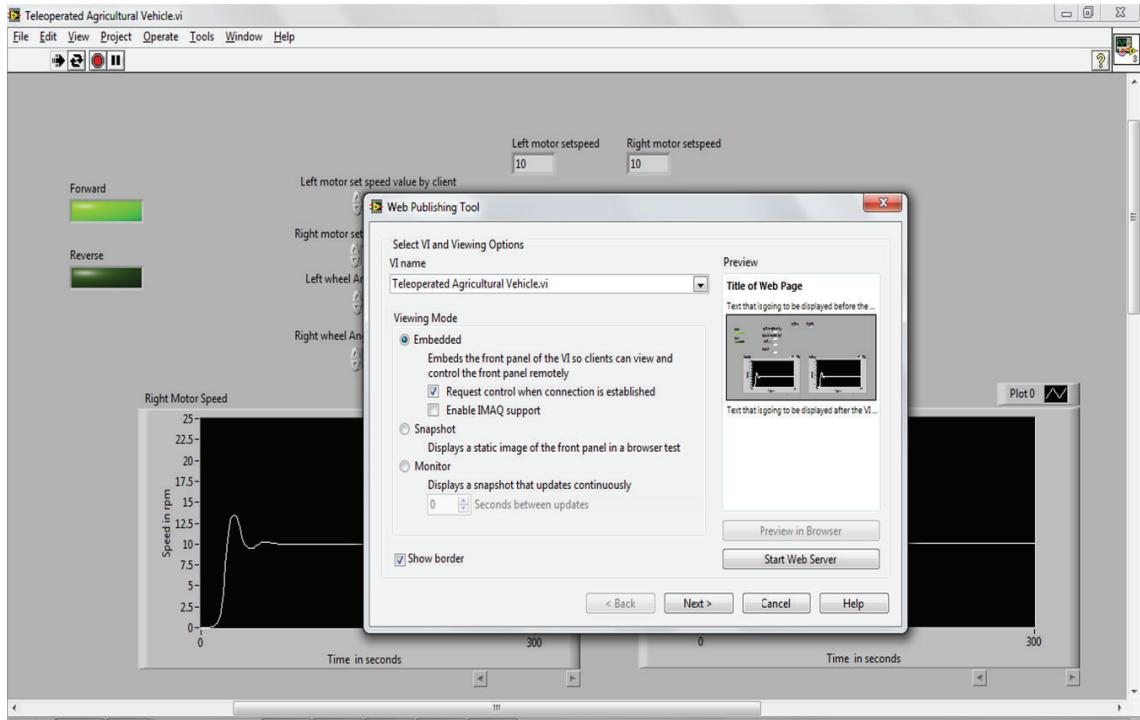


FIGURE 9: Configuration of server computer for remote clients using Web Publishing Tool in LabVIEW.

Online control is achieved from remote place using Web Publishing Tool in LabVIEW. This tool is used to create an HTML document and embed static or animated images of the server computer's front panel in an HTML document as shown in Figures 9 and 10. When the HTML document is created, URL is generated as shown in Figure 11. Using this generated URL, a remote user can view and control the front panel remotely by a web browser.

8. Experimental Results

The developed fuzzy logic controller and the control algorithm for ploughing action and solenoid valve were implemented using ATMEL 89C51 microcontroller. The drive wheels speeds are controlled by two 12 V DC motors. By using the proposed control algorithm, the speeds of these DC motors are controlled and measured values are plotted in the graph. The experimental response of left and right DC motors

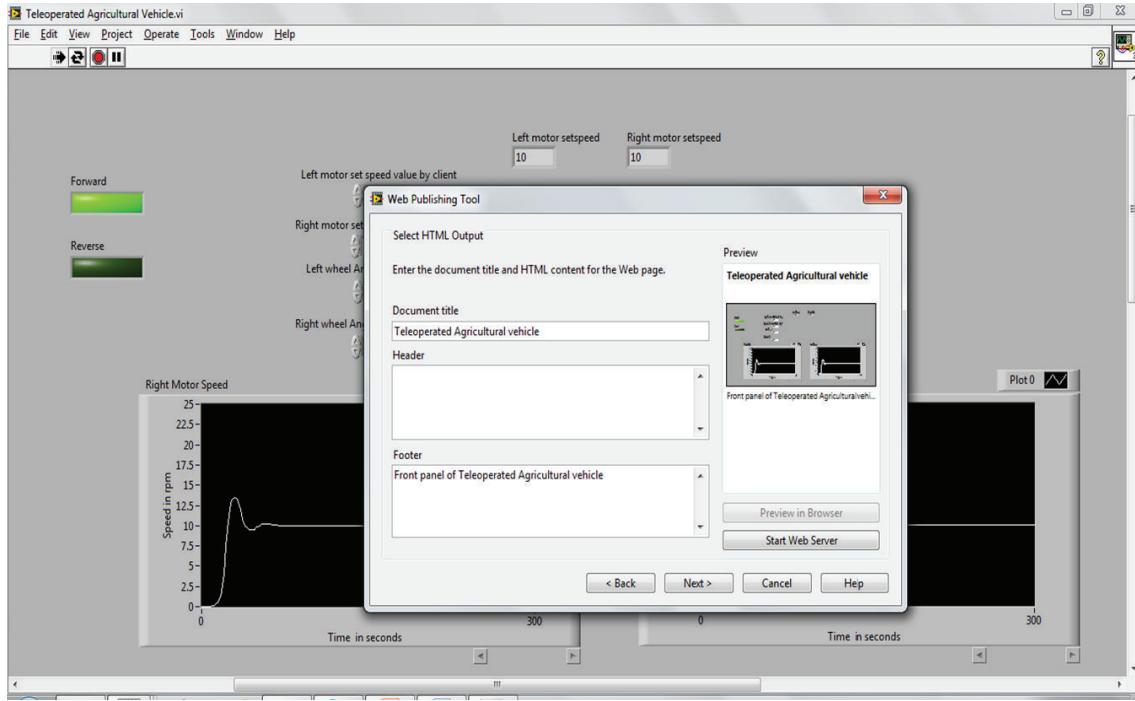


FIGURE 10: Configuration of server computer for remote clients using Web Publishing Tool in LabVIEW with header and footer details.

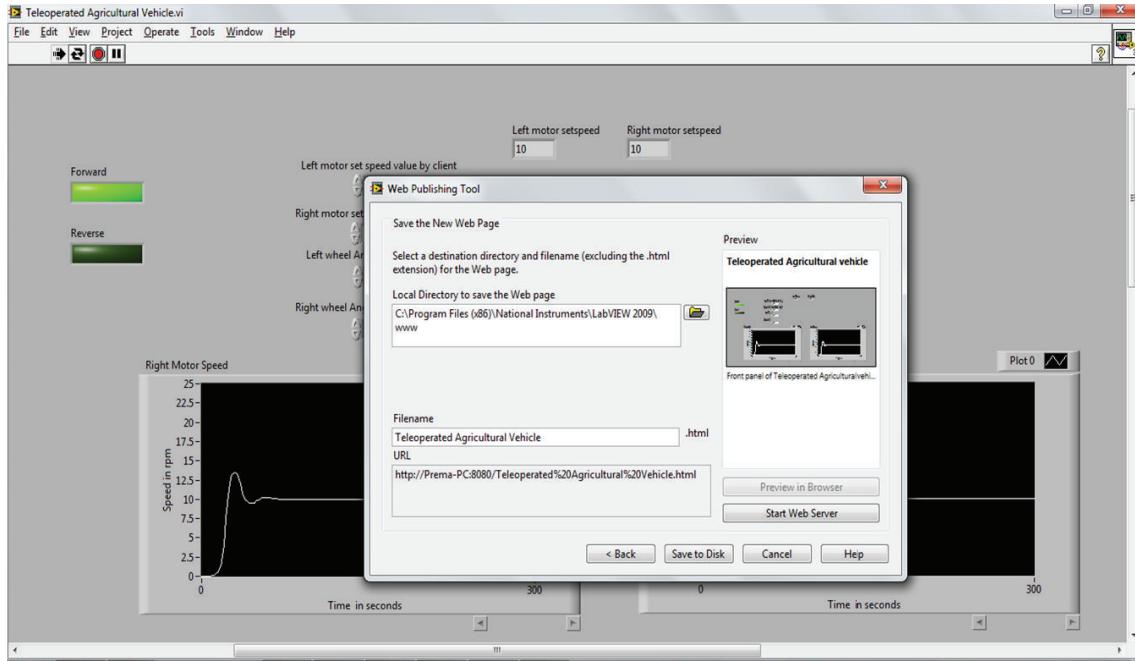


FIGURE 11: URL generation by the HTML document.

when the vehicle moves in forward, reverse, left, and right directions are shown in Figures 12, 13, 14, and 15.

9. Conclusion

In this paper, a teleoperated agricultural vehicle capable of performing ploughing, sowing, and soil moisture sensing

was successfully developed. Two independent drive wheels installed on the vehicle are used for motion and steering control. The developed fuzzy logic controller controls the steering angle of the teleoperated vehicle according to the desired steering angle. Using the proposed agricultural vehicle depth of ploughing and the distance between the seeds can be varied. By measuring the moisture content of

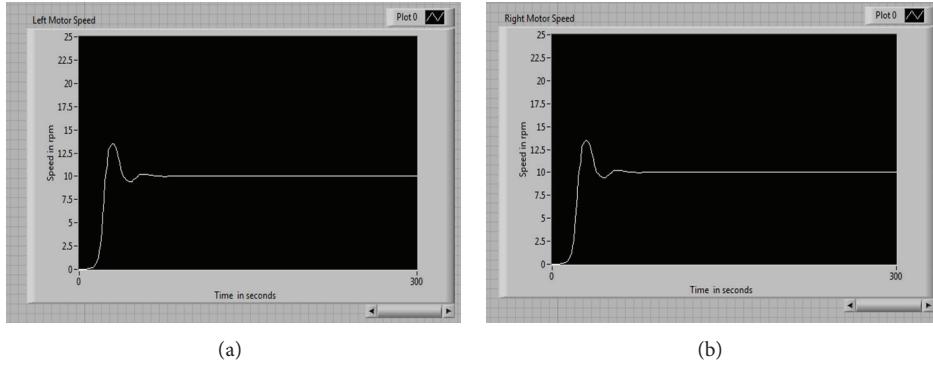


FIGURE 12: Graph of actual speed values of left motor and right motor when the agricultural vehicle moves in forward direction with the set speed value of 10 rpm.

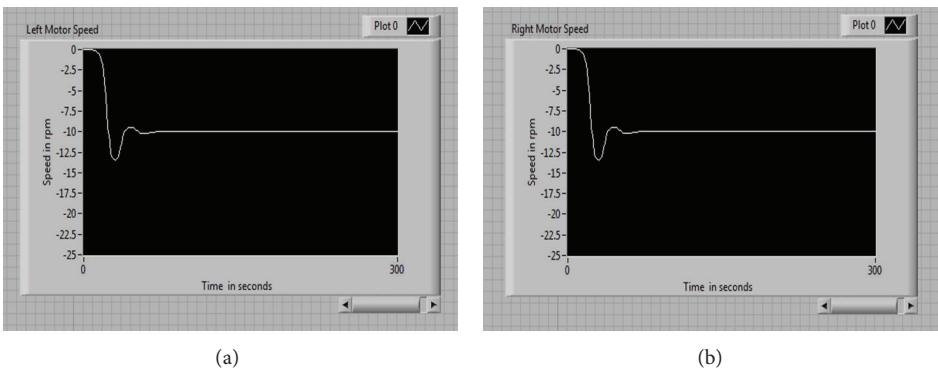


FIGURE 13: Graph of actual speed values of left motor and right motor when the agricultural vehicle moves in reverse direction with the set speed value of 10 rpm.

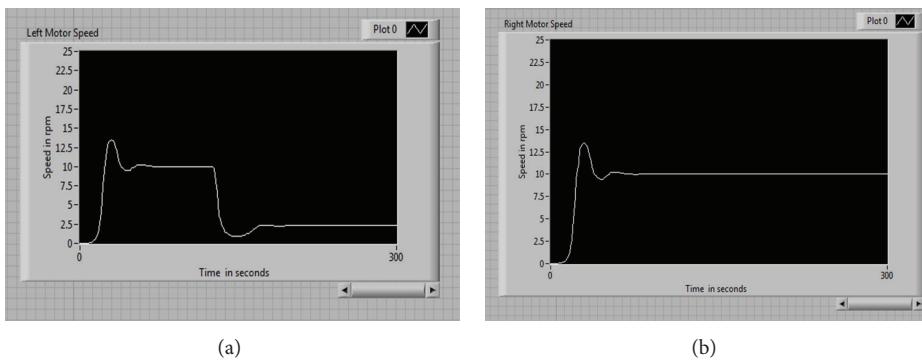


FIGURE 14: Graph of actual speed values of left motor and right motor when the agricultural vehicle turns in left direction.

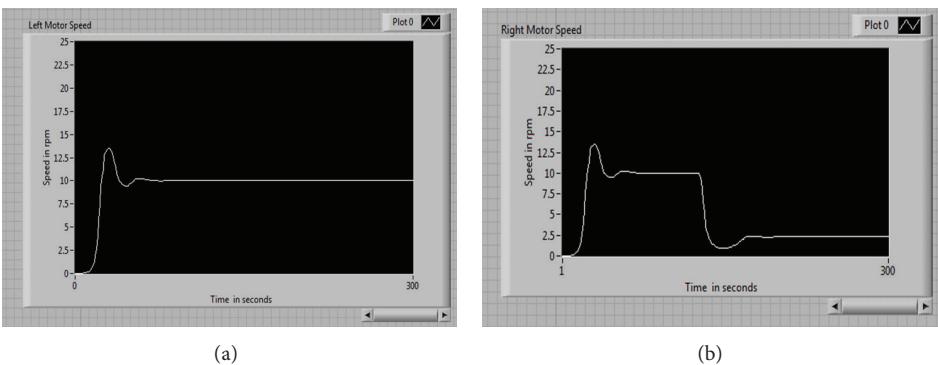


FIGURE 15: Graph of actual speed values of left motor and right motor when the agricultural vehicle turns in right direction.

the soil, the crops are watered adequately. Thus this teleoperated agricultural vehicle reduces the manpower and becomes advantageous.

References

- [1] N. Kondo, M. Monta, and N. Noguchi, *Agricultural Robots (1)*, Corona, 2006.
- [2] J. Marchant, T. Hague, N. Tillett, and J. Sanchiz, "Research on an autonomous vehicle for precise plant treatments," in *Proceedings of the 5th International Workshop on Robotics and Automated Machinery for Bio-Productions*, pp. 237–242, Gandia, Spain, 1997.
- [3] K. Inoue, K. Otsuka, M. Sugimoto, and N. Murakami, "Estimation of place of tractor and adaptive control method of autonomous tractor using INS and GPS," in *Proceedings of the 5th International Workshop on Robotics and Automated Machinery for Bio-Productions*, pp. 27–32, Gandia, Spain, 1997.
- [4] T. Pilarski, M. Happold, H. Pangels, M. Ollis, K. Fitzpatrick, and A. Stentz, "The Demeter system for automated harvesting," *Autonomous Robots*, vol. 13, no. 1, pp. 9–20, 2002.
- [5] A. Sgorbissa and R. Zaccaria, "Planning and obstacle avoidance in mobile robotics," *Robotics and Autonomous Systems*, vol. 60, no. 4, pp. 628–638, 2012.
- [6] N. Murakami, A. Ito, J. D. Will et al., "Development of a teleoperation system for agricultural vehicles," *Computers and Electronics in Agriculture*, vol. 63, no. 1, pp. 81–88, 2008.
- [7] N. Noguchi, J. Will, J. Reid, and Q. Zhang, "Development of a master-slave robot system for farm operations," *Computers and Electronics in Agriculture*, vol. 44, no. 1, pp. 1–19, 2004.
- [8] Y. Fang, I. Masaki, and B. Horn, "Depth-based target segmentation for intelligent vehicles: fusion of radar and binocular stereo," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 3, pp. 196–202, 2002.
- [9] S. Wu, S. Decker, P. Chang, T. Camus, and J. Eledath, "Collision sensing by stereo vision and radar sensor fusion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 4, pp. 606–614, 2009.
- [10] M. Kise, Q. Zhang, and N. Noguchi, "An obstacle identification algorithm for a laser range finder-based obstacle detector," *Transactions of the American Society of Agricultural Engineers*, vol. 48, no. 3, pp. 1269–1278, 2005.
- [11] L. Guo, Q. Zhang, and S. Han, "Agricultural machinery safety alert system using ultrasonic sensors," *Journal of Agricultural Safety and Health*, vol. 8, no. 4, pp. 385–396, 2002.
- [12] M. Knudson and K. Turner, "Adaptive navigation for autonomous robots," *Robotics and Autonomous Systems*, vol. 59, no. 6, pp. 410–420, 2011.
- [13] K. H. Kim, I. C. Baik, S. K. Chung, and M. J. Youn, "Robust speed control of brushless DC motor using adaptive input-output linearization technique," *IEEE Proceedings on Electric Power Applications*, vol. 144, pp. 469–475, 1997.
- [14] M. S. Ibbini and W. S. Zakaria, "Nonlinear control of DC machines," *Electric Machines and Power Systems*, vol. 24, no. 1, pp. 21–35, 1996.
- [15] M. A. Rahman and M. A. Hoque, "On-line adaptive artificial neural network based vector control of permanent magnet synchronous motors," *IEEE Transactions on Energy Conversion*, vol. 13, no. 4, pp. 311–318, 1998.
- [16] A. Rubaai and R. Kotaru, "Online identification and control of a dc motor using learning adaptation of neural networks," *IEEE Transactions on Industry Applications*, vol. 36, no. 3, pp. 935–942, 2000.
- [17] J.-H. Horng, "Neural adaptive tracking control of a DC motor," *Information Sciences*, vol. 118, no. 1, pp. 1–13, 1999.
- [18] A. Rubaai, D. Ricketts, and M. D. Kankam, "Development and implementation of an adaptive fuzzy-neural-network controller for brushless drives," *IEEE Transactions on Industry Applications*, vol. 38, no. 2, pp. 441–447, 2002.
- [19] N. Senthil Kumar, V. Sadasivam, H. M. Asan Sukriya, and S. Balakrishnan, "Design of low cost universal artificial neuron controller for chopper fed embedded DC drives," *Applied Soft Computing Journal*, vol. 8, no. 4, pp. 1637–1642, 2008.
- [20] N. S. Kumar, V. Sadasivam, and K. Prema, "Design and simulation of fuzzy controller for closed loop control of chopper fed embedded DC drives," in *Proceedings of the International Conference on Power System Technology (POWERCON '04)*, pp. 613–617, Singapore, November 2004.
- [21] D. Driankov, H. Hellendoorn, and M. Reinfrank, *An Introduction to Fuzzy Control*, Narosa Publishing House, New Delhi, India, 2nd edition, 2001.
- [22] B. C. Besselink, "Computer controlled steering system for vehicles having two independently driven wheels," *Computers and Electronics in Agriculture*, vol. 39, no. 3, pp. 209–226, 2003.

Research Article

A Sarsa(λ) Algorithm Based on Double-Layer Fuzzy Reasoning

Quan Liu,^{1,2} Xiang Mu,¹ Wei Huang,¹ Qiming Fu,¹ and Yonggang Zhang²

¹ School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215006, China

² Key Laboratory of Symbol Computation and Knowledge Engineering, Jilin University, Ministry of Education, Changchun 130012, China

Correspondence should be addressed to Quan Liu; quanliu@suda.edu.cn

Received 1 August 2013; Accepted 13 September 2013

Academic Editor: Zhiqiang Ma

Copyright © 2013 Quan Liu 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.

Solving reinforcement learning problems in continuous space with function approximation is currently a research hotspot of machine learning. When dealing with the continuous space problems, the classic Q-iteration algorithms based on lookup table or function approximation converge slowly and are difficult to derive a continuous policy. To overcome the above weaknesses, we propose an algorithm named DFR-Sarsa(λ) based on double-layer fuzzy reasoning and prove its convergence. In this algorithm, the first reasoning layer uses fuzzy sets of state to compute continuous actions; the second reasoning layer uses fuzzy sets of action to compute the components of Q-value. Then, these two fuzzy layers are combined to compute the Q-value function of continuous action space. Besides, this algorithm utilizes the membership degrees of activation rules in the two fuzzy reasoning layers to update the eligibility traces. Applying DFR-Sarsa(λ) to the Mountain Car and Cart-pole Balancing problems, experimental results show that the algorithm not only can be used to get a continuous action policy, but also has a better convergence performance.

1. Introduction

Reinforcement learning is a kind of machine learning methods that gets the maximum cumulative rewards by interacting with the environment [1, 2]. If a reinforcement learning problem can be modeled as a Markov decision process (MDP), methods such as dynamic programming (DP), Monte Carlo (MC), and temporal difference (TD) can be used to get an optimal policy.

Classic reinforcement learning methods are generally used for dealing with discrete state and action space problems, where each of the state values or state action values is stored in a lookup table. This kind of methods can effectively solve simple tasks, but not for large, continuous space problems. At present, the most common approach to solve this problem is using function approximation methods to approximate the state value or action value function. The approximate function can generalize the learned experience from a state space subset to the entire state space. Besides, an agent can choose the best action sequence through the function approximation [3, 4]. A variety of function approximation methods are used to reinforcement learning problems at

present. Sutton et al. proposed a gradient TD (GTD) learning algorithm [5], which combined TD algorithms with linear function approximation, and also introduced a new objective function related to Bellman errors. Sherstov and Stone proposed a linear function approximation algorithm based on online adaptive tile coding, in which the experimental results verified its effectiveness [6]. Heinen and Engel used incremental probabilistic neural network to approximate value function in reinforcement learning, which can be used to solve continuous state space problems well [7].

Reinforcement algorithms with the function approximation methods mentioned above usually have slow convergence and generally can only be used for getting discrete action policies [5–9]. By introducing prior knowledge, reinforcement learning algorithms based on fuzzy inference systems (FIS) not only can effectively accelerate the convergence rate, but also may get continuous action policies [10–12]. Horiuchi et al. put forward fuzzy interpolation-based Q-learning, which can solve the continuous space problems [13]. Gorenne and Jouffe combined FIS and Q-learning, using prior knowledge to make the global approximator, which can effectively speed up the convergence rate. However, the

algorithm cannot be used to get a continuous action policy [14]. Fuzzy Sarsa proposed by Tokarchuk et al. can effectively reduce the scale of state space and accelerate the convergence rate, but it easily causes “curse of dimensionality” when applied to multidimensional state-space problems [15]. Type-2 fuzzy Q-learning proposed by Hsu and Juang has strong robustness to noise, but its time complexity is relatively high, and meanwhile, it cannot guarantee convergence [12].

Though the classic Q-iteration algorithms based on only one fuzzy inference system can be used for solving continuous action space problems, there still exist reasons for the slow convergence: for each iteration step in the learning process, there might exist a state-action pair that corresponds to different Q-values due to the structure of FIS. If the next iteration step needs to use the Q-value of the mentioned state-action pair to update the value function, the algorithm will simply select a Q-value randomly, since there are no criteria on how to choose the best one from different Q-values, which will influence the learning speed. Because this situation may happen many times in the learning process, it will greatly slow down the convergence rate.

In allusion to the problem that classic Q-iteration algorithms based on the lookup table and fuzzy inference system converge slowly and cannot obtain continuous action policies as well, DFR-Sarsa(λ), which means Sarsa(λ) based on double-layer fuzzy reasoning, is proposed in this paper, and the convergence is proven theoretically. The algorithm has two-layer fuzzy reasoning. Firstly, it puts states as input of the first fuzzy reasoning layer and gets continuous actions as output. Secondly, the second fuzzy reasoning layer uses the obtained actions from the first layer as input and gets Q-value component of each activation rule of the first layer. Finally, through the combination of two-layer fuzzy reasoning, Q-values of the input states are obtained. What is more, a new eligibility trace based on gradient descent is defined, which is dependent on membership degrees of activation rule in two-layer fuzzy reasoning. Applying DFR-Sarsa(λ) and other algorithms to Mountain Car and Cart-pole Balancing problems, the results show that DFR-Sarsa(λ) not only can obtain a continuous action policy, but also has a better convergence performance.

2. Backgrounds

2.1. Markov Decision Process. In reinforcement learning framework, the process interacting with the environment can be modeled as an MDP [16], and the MDP can be described as a quadruple $M = \langle X, U, \rho, f \rangle$, where

- (1) X is the state set and $x_t \in X$ is the state at time t ;
- (2) U is the action set and $u_t \in U$ is the action that the agent takes at time t ;
- (3) $\rho : X \times U \rightarrow \mathbb{R}^n$ is the reward function, that means, after the agent takes action u_t at time t , the current state transfers from x_t to x_{t+1} , and the agent receives an immediate reward $r(x_t, u_t, x_{t+1})$ at the same time. r_t represents a random reward generated from a distribution with mean $r(x_t, u_t, x_{t+1})$;

(4) $f : X \times U \times X \rightarrow [0, 1]$ is the state transition function, where $f(x, u, x')$ represents the probability of reaching x' after taking action u in state x .

The policy $h(x, u)$ is a mapping from state space X to action space U , $h : X \rightarrow U$, which represents the probability that the agent selects action u in state x . $h(x, u)$ is used to solve the state value function (V -function) or action value function (Q -function). V -function satisfies (1)

$$\forall x \in X:$$

$$V^h(x) = \sum_{u \in U} h(x, u) \left(\rho(x, u) + \gamma \sum_{x' \in X} f(x, u, x') V^h(x') \right). \quad (1)$$

And Q -function satisfies (2)

$$\forall x \in X : Q^h(x, u) = \rho(x, u) + \gamma \sum_{x' \in X} f(x, u, x') V^h(x'). \quad (2)$$

The objective of reinforcement learning is to get the optimal policy h^* . It satisfies for all $x \in X$: $V^{h^*}(x) \geq V^h(x)$. Under the optimal policy h^* , the optimal V -function and optimal Q -function satisfy (3) and (4), respectively:

$$\forall x \in X:$$

$$V^*(x) = \max_{u \in U} \left(\rho(x, u) + \gamma \sum_{x' \in X} f(x, u, x') V^*(x') \right), \quad (3)$$

$$\forall x \in X, u \in h(x):$$

$$Q^*(x, u) = \rho(x, u) + \gamma \sum_{x' \in X} f(x, u, x') \max_{u' \in U} Q^*(x', u'). \quad (4)$$

If f and ρ are known, DP is a good solution for getting optimal action policy. However if f and ρ are unknown, TD algorithms such as Q-learning or Sarsa can be the choice. Sarsa is an on-policy algorithm, and when the eligibility trace mechanism is introduced, it becomes a more efficient algorithm, which can effectively deal with temporal credit assignment. Besides, Sarsa(λ) can be combined with function approximation to solve continuous state space problems.

Definition 1 is a constraint on bounded MDP (mainly about state-space, action-space, reward, and value function). Attention should be given that all algorithms in this paper meet the definition.

Definition 1 (bounded MDP). X and U are known as finite sets; let Z represent the state-action set; that is, $Z : X \times U$, and then Z is also a finite set. Reward function ρ satisfies $0 \leq \rho(x, u) \leq C$. The bound factor of MDP is $\beta = 1/(1 - \gamma)$, where γ is a discount factor. For all $x \in X$ and for all $(x, u) \in Z$, $0 \leq V(x) \leq \beta C$ and $0 \leq Q(x, u) \leq \beta C$ hold.

2.2. Fuzzy Inference System. FIS is a system that can handle fuzzy information. Typically, it mainly consists of a set of fuzzy rules whose design and interaction are crucial to the FIS's performance.

There are many types of fuzzy inference systems at present [17] in which a simple type of FIS named TSK-FIS is described as follows:

$$\text{Rule } R_r: \text{if } x_1 \text{ is } \chi_1^r \text{ AND } \dots \text{ AND } x_n \text{ is } \chi_n^r \text{ then } y = g_r(\mathbf{x}), \quad (5)$$

where the first part is called antecedent and the second part is called consequent. R_r means r th rule in the rule base. $\mathbf{x} = (x_1, x_2, \dots, x_N)$ is an N -dimensional input variable. χ_i^r is the fuzzy set in the r th fuzzy rule which corresponds to the i th dimension of input variable. A membership function $\mu_{\chi_{r,i}}(x_i)$ is usually used to describe it. $y = g_r(\mathbf{x})$ is a polynomial function with an input variable \mathbf{x} . If the input \mathbf{x} is a vector, the output y is also a vector. When $g_r(\mathbf{x})$ is a constant, FIS is called zero-order FIS.

When the FIS has an exact input value $\mathbf{x} = (x_1, x_2, \dots, x_N)$, we can calculate the firing strength $\phi_r(\mathbf{x})$ of the r th rule (for T-norm product):

$$\phi_r(\mathbf{x}) = \mu_{\chi_{r,1}}(x_1) \cdot \mu_{\chi_{r,2}}(x_2) \cdots \mu_{\chi_{r,N}}(x_N), \quad r=1, \dots, N_R. \quad (6)$$

$\phi_r(\mathbf{x})$ is used to calculate the output of FIS: set firing strength $\phi_r(\mathbf{x})$ as weight, multiply their corresponding consequent and sum up; then we can obtain the final output $Y(\mathbf{x})$ as follows:

$$Y(\mathbf{x}) = \frac{\sum (\phi_1(\mathbf{x}) y_1 + \phi_2(\mathbf{x}) y_2 + \dots + \phi_{N_R}(\mathbf{x}) y_{N_R})}{\sum (\phi_1(\mathbf{x}) + \phi_2(\mathbf{x}) + \dots + \phi_{N_R}(\mathbf{x}))}. \quad (7)$$

TSK-FIS can be used for function approximation which approximates the objective function by updating the consequent of fuzzy rules. In general, the approximation error is measured by mean square error (MSE). When FIS gets an optimal approximation performance, the vector θ , which consists of all rules consequents, satisfies (8)

$$\theta = \arg \min_{\theta} \sum_{i=1}^N (Y_i(\mathbf{x}) - \hat{Y}_i(\mathbf{x}))^2, \quad (8)$$

where $Y_i(\mathbf{x})$ is the objective function and $\hat{Y}_i(\mathbf{x})$ is its approximate function.

3. DFR-Sarsa(λ)

3.1. The Update of Q-Value. Under the framework of MDP, two-layer fuzzy inference structures are constructed to approximate Q-function. Figure 1 shows the framework using two-layer fuzzy reasoning to approximate Q-function, where the inputs of FIS1 are states; the outputs are continuous actions obtained by FIS1 through fuzzy reasoning; the inputs of FIS2 are continuous actions obtained from FIS1; the outputs are the components of Q-value of the continuous

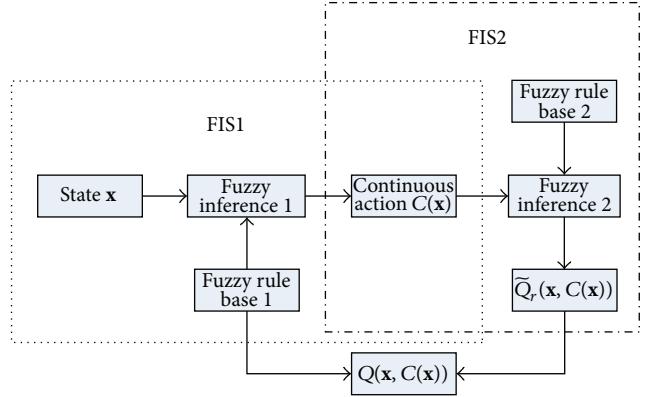


FIGURE 1: Framework of approximating the Q-value functions by using double-layer fuzzy reasoning.

actions. Then, the two-layer FISs are combined to get the approximating Q-function of continuous action $C(\mathbf{x})$.

The main structure of the two-layer FIS is described as follows.

(1) The rule of FIS1 is given as follows:

$$\begin{aligned} \text{Rule } R_r: & \text{if } x_1 \text{ is } \chi_1^r \text{ AND } \dots \text{ AND } x_n \text{ is } \chi_n^r \\ & \text{then } y = u_{r,1} \text{ with } q_{r,1} = \theta_{r,1} \\ & \text{or } y = u_{r,2} \text{ with } q_{r,2} = \theta_{r,2} \\ & \vdots \\ & \text{or } y = u_{r,M} \text{ with } q_{r,M} = \theta_{r,M}, \end{aligned} \quad (9)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_N)$ is the state and $u_{r,j}$ is the j th discrete action in the r th fuzzy rule. The action space is divided into M discrete actions. $q_{r,j}$ is a component of Q-value corresponding to the j th discrete action in the r th fuzzy rule. When the state is \mathbf{x} , the firing strength of the r th rule is

$$\varphi_r(\mathbf{x}) = \mu_{\chi_{r,1}}(x_1) \cdot \mu_{\chi_{r,2}}(x_2) \cdots \mu_{\chi_{r,N}}(x_N). \quad (10)$$

If $\varphi_r(\mathbf{x}) > 0$, we call the r th rule "the activation rule."

In the activation rule R_r , we select an action from M discrete actions by ϵ -greedy action selection policy according to the value $q_{r,j}$. The selected action is called activation action, denoted by \bar{u}_r . Therefore, by multiplying activation actions selected from FIS1 to its firing strength $\varphi_r(\mathbf{x})$ and summing them up, we get the continuous action $C(\mathbf{x})$ as follows:

$$C(\mathbf{x}) = \frac{\sum_{r=1}^{N_R} \varphi_r(\mathbf{x}) \bar{u}_r}{\sum_{r=1}^{N_R} \varphi_r(\mathbf{x})}. \quad (11)$$

We call $C(\mathbf{x})$ a continuous action because the change of $C(\mathbf{x})$ is smooth with state \mathbf{x} , which does not mean that any action in action space can be selected in state \mathbf{x} . To simplify (11), regularize the firing strength $\varphi_r(\mathbf{x})$ as follows:

$$\varphi_r(\mathbf{x}) = \frac{\varphi_r(\mathbf{x})}{\sum_{r=1}^{N_R} \varphi_r(\mathbf{x})}; \quad (12)$$

so (11) can be written as

$$C(\mathbf{x}) = \sum_{r=1}^{N_R} \phi_r(\mathbf{x}) \tilde{u}_r. \quad (13)$$

(2) The rule of FIS2 is given as follows:

$$\begin{aligned} \tilde{R}_{r,1}: & \text{ if } u \text{ is } \nu_{r,1} \text{ then } q_{r,1} = \theta_{r,1} \\ \tilde{R}_{r,2}: & \text{ if } u \text{ is } \nu_{r,2} \text{ then } q_{r,2} = \theta_{r,2} \\ & \vdots \\ \tilde{R}_{r,M}: & \text{ if } u \text{ is } \nu_{r,M} \text{ then } q_{r,M} = \theta_{r,M}. \end{aligned} \quad (14)$$

The construction of $\tilde{R}_{r,j}$ depends on FIS1. The core of the fuzzy set $\nu_{r,j}$ is the j th action of the r th rule in FIS1, and its membership function is described as $\sigma_{\nu_{r,j}}(u)$; the value $q_{r,j}$ from the consequent part of the rule equals the value $q_{r,j}$ in FIS1.

Set the continuous action $C(\mathbf{x})$ obtained from FIS1 as the input of FIS2; it can activate N_R rules of FIS2. Through fuzzy reasoning of FIS2, we can get the Q -value component of the r th rule in FIS1 as follows:

$$\tilde{Q}_r(\mathbf{x}, C(\mathbf{x})) = \frac{\sum_{j=1}^M \sigma_{\nu_{r,j}}(C(\mathbf{x})) \theta_{r,j}}{\sum_{j=1}^M \sigma_{\nu_{r,j}}(C(\mathbf{x}))}. \quad (15)$$

In the same way of getting (12), regularize the membership function $\sigma_{\nu_{r,j}}(C(\mathbf{x}))$ in (15); we get

$$\mu_{\nu_{r,j}}(C(\mathbf{x})) = \frac{\sigma_{\nu_{r,j}}(C(\mathbf{x}))}{\sum_{j=1}^M \sigma_{\nu_{r,j}}(C(\mathbf{x}))}; \quad (16)$$

then (15) can be written as

$$\tilde{Q}_r(\mathbf{x}, C(\mathbf{x})) = \sum_{j=1}^M \mu_{\nu_{r,j}}(C(\mathbf{x})) \theta_{r,j}. \quad (17)$$

From (17), we can get $\tilde{Q}_r(\mathbf{x}, C(\mathbf{x}))$, the Q -value component obtained by the activation rule R_r of FIS1. So when taking continuous action $C(\mathbf{x})$, the Q -value of all activation rules in FIS1 is given as follows:

$$\begin{aligned} Q(\mathbf{x}, C(\mathbf{x})) &= \sum_{r=1}^{N_R} \phi_r(\mathbf{x}) \tilde{Q}_r(\mathbf{x}, C(\mathbf{x})) \\ &= \sum_{r=1}^{N_R} \sum_{j=1}^M \phi_r(\mathbf{x}) \mu_{\nu_{r,j}}(C(\mathbf{x})) \theta_{r,j}. \end{aligned} \quad (18)$$

From (18) we can see that Q -value depends on fuzzy sets of the two-layer FIS and their shared consequent variables $\theta_{r,j}$. Since fuzzy sets are set according to prior knowledge in advance, they are no longer changed in the algorithm. In order to get convergent Q -value, the FISs require updating $\theta_{r,j}$ until convergence.

In order to minimize the approximation error of FIS, that is, parameter vector $\boldsymbol{\theta}$ meets (8), the algorithm uses gradient descent method to update the parameter vector $\boldsymbol{\theta}$ as follows:

$$\begin{aligned} \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \frac{1}{2} \alpha \nabla_{\boldsymbol{\theta}_t} [r_{t+1} + \gamma Q_t(\mathbf{x}_{t+1}, u_{t+1}) - Q_t(\mathbf{x}_t, u_t)]^2 \\ &= \boldsymbol{\theta}_t + \alpha [r_{t+1} + \gamma Q_t(\mathbf{x}_{t+1}, u_{t+1}) - Q_t(\mathbf{x}_t, u_t)] \nabla_{\boldsymbol{\theta}_t} Q_t(\mathbf{x}_t, u_t), \end{aligned} \quad (19)$$

where the bracket part in (19) is the TD error. Set $\delta = r_{t+1} + \gamma Q_t(\mathbf{x}_{t+1}, u_{t+1}) - Q_t(\mathbf{x}_t, u_t)$; combining the backward TD(λ) algorithm [1], we get

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \delta \mathbf{e}_t, \quad (20)$$

where α is a step-size parameter and \mathbf{e}_t is the eligibility trace vector at time t , which corresponds to parameter vector $\boldsymbol{\theta}_t$. It is updated as follows:

$$\begin{aligned} \mathbf{e}_t(r, j) &\leftarrow \begin{cases} \gamma \lambda \mathbf{e}_{t-1}(r, j) + \nabla_{\boldsymbol{\theta}_t} Q_t(\mathbf{x}_t, u_t), & \text{if } \nabla_{\boldsymbol{\theta}_t} Q_t(\mathbf{x}_t, u_t) \neq \vec{0}, \\ \gamma \lambda \mathbf{e}_{t-1}(r, j), & \text{else.} \end{cases} \end{aligned} \quad (21)$$

\mathbf{e}_t of (21) is a kind of accumulating trace [1], where γ is the discount factor and λ is the decay factor. $\nabla_{\boldsymbol{\theta}_t} Q_t(\mathbf{x}_t, u_t)$ represents the gradient vector obtained by the partial derivative of Q -function on each dimension of parameter vector at time t [1]. According to (18), we get the gradient value of each dimension in $\boldsymbol{\theta}_t$ at time t as follows:

$$\nabla_{\theta_{r,j}} Q_t(\mathbf{x}, u) = \nabla_{\theta_{r,j}} \sum_{i=1}^{N_R} \sum_{j=1}^M \phi_r(\mathbf{x}) \mu_{\nu_{r,j}}(u) \theta_{r,j} = \phi_r(\mathbf{x}) \mu_{\nu_{r,j}}(u),$$

$$\text{where } r = 1, \dots, N_R, j = 1, \dots, M; \quad (22)$$

then (21) can be further expressed as

$$\begin{aligned} \mathbf{e}_t(r, j) &\leftarrow \begin{cases} \gamma \lambda \mathbf{e}_{t-1}(r, j) + \phi_r(\mathbf{x}) \mu_{\nu_{r,j}}(u), & \text{if } \phi_r(\mathbf{x}) \neq 0 \text{ and} \\ & \mu_{\nu_{r,j}}(u) \neq 0, \\ \gamma \lambda \mathbf{e}_{t-1}(r, j), & \text{else.} \end{cases} \end{aligned} \quad (23)$$

3.2. The Learning Process of DFR-Sarsa(λ). In this section, DFR-Sarsa(λ) is proposed based on the algorithm Sarsa in literature [1] and the content of MDP in Section 2.1. DFR-Sarsa(λ) not only can solve reinforcement learning problems with continuous state and discrete action space, but also can solve problems with continuous state and continuous action space. Algorithm 1 describes the general process of DFR-Sarsa(λ).

- (1) Initialize parameter vector $\theta = \vec{0}$, eligibility trace vector $e = \vec{0}$, discount factor γ , step-size parameter α
- (2) Repeat(for every episode):
 - (3) $x \leftarrow$ initial state
 - (4) According to (10), compute $\phi_r(x)$, $r = 1, \dots, N_R$
 - (5) According to ε -greedy policy, select activation action \tilde{u}_r , $r = 1, \dots, N_R$
 - (6) According to (13), select action u when state is x
 - (7) According to (16), compute $\mu_{\nu_{r,j}}(u)$, $r = 1, \dots, N_R$, $j = 1, \dots, M$
 - (8) According to (17) and (18), compute Q_u
 - (9) Repeat(for each step of episode)
 - (10) Update eligibility trace: $e(r, j) \leftarrow \gamma \lambda e(r, j) + \phi_r(x) \mu_{\nu_{r,j}}(u)$, $r = 1, \dots, N_R$, $j = 1, \dots, M$
 - (11) Take action u , receive next state x' and reward r
 - (12) $\delta \leftarrow r - Q_u$
 - (13) According to ε -greedy policy, select activation action \tilde{u}_r , $r = 1, \dots, N_R$
 - (14) According to (13), select action u' when state is x'
 - (15) According to (16), compute $\mu_{\nu_{r,j}}(u')$, $r = 1, \dots, N_R$, $j = 1, \dots, M$
 - (16) According to (10), compute $\phi_r(x')$, $r = 1, \dots, N_R$
 - (17) According to (17) and (18), compute $Q_{u'}$
 - (18) $\delta \leftarrow \delta + \gamma Q_{u'}$
 - (19) $\theta \leftarrow \theta + \alpha \delta e$
 - (20) $u \leftarrow u'$
 - (21) Until x' is the terminal state
 - (22) Until preset episode number or other terminal condition meets

ALGORITHM 1: DFR-Sarsa(λ).

3.3. Convergence Analysis. In the literature [18, 19], the convergence of on-policy TD(λ) using linear function approximation is analyzed in detail. When this kind of algorithm meets some assumptions and lemmas, it converges with probability 1. Since DFR-Sarsa(λ) is exactly such an on-policy TD(λ) algorithm, it can be proved to be convergent when it satisfies some assumptions and lemmas in literature [18]. And this paper will not take too much details for its convergence proof.

Assumption 2. The state transition function and reward function of MDP follow stable distributions.

Lemma 3. *The Markov chain that DFR-Sarsa(λ) depends on is irreducible and aperiodic, and the reward and value function are bounded.*

Proof. Firstly, we prove its irreducibility. According to the property of Markov process, if any two states of a Markov process can be transferred from each other, it is irreducible [20]. DFR-Sarsa(λ) is used for solving reinforcement learning problems that satisfy MDP framework, and the MDP meets Definition 1. Thus for any state x in the MDP, there must exist an f that meets $f(x, u, x') \geq 0$, which indicates that state x can be visited infinitely. Therefore, each state can be transferred to any other state. So the Markov chain of DFR-Sarsa(λ) is irreducible.

Secondly, we prove that it is aperiodic. For the irreducible Markov chain, if one of the states in Markov chain is proved aperiodic, the entire Markov chain can be proved aperiodic. In addition, if a state of the Markov Chain has the property of autoregression, the state can be proven aperiodic [20]. For state x of the MDP, there must exist a state transition

satisfying $f(x, u, x) > 0$, which indicates that state x is autoregressive. From the above analysis, we can conclude that the MDP is aperiodic. Therefore, the Markov chain that DFR-Sarsa(λ) depends on is aperiodic.

Finally, we prove that its reward and value function are bounded. Literature [1] shows that value function is a discounted accumulating reward function, which satisfies the equation $Q(x, u) = \sum_{i=0}^{\infty} \gamma^i \rho(x, u)$, $\gamma \in (0, 1)$. By Definition 1, we know that the reward function ρ is bounded, and it satisfies $0 \leq \rho(x, u) \leq C$, where C is a constant. Hence

$$Q(x, u) = \sum_{i=0}^{\infty} \gamma^i \rho(x, u) < \sum_{i=0}^{\infty} \gamma^i C = \lim_{i \rightarrow \infty} \frac{(1 - \gamma^i)}{1 - \gamma} C = \frac{C}{1 - \gamma}. \quad (24)$$

By Inequation (24), we can conclude that value function $Q(x, u)$ is bounded.

In summary, Lemma 3 is proved. \square

Condition 1. For each membership function i , there exists a unique state x_i that $\mu_i(x_i) > \mu_i(x)$, for all $x \neq x_i$, while the other membership functions in state x_i are 0; that is, $\mu_{i'}(x_i) = 0$, for all $i' \neq i$.

Lemma 4. *The basis functions of DFR-Sarsa(λ) are bounded, and the basis function vector is linearly independent.*

Proof. Firstly, we prove the basis functions are bounded. From $\phi_r(x) \in [0, 1]$ and $\mu_{\nu_{r,j}}(C(x)) \in [0, 1]$, we get

$$\|\phi_r(x) \mu_{\nu_{r,j}}(C(x))\|_{\infty} \leq 1, \quad (25)$$

where $\|\cdot\|_\infty$ represents infinite norm. Since the basis function of DFR-Sarsa(λ) is known as $\phi_r(\mathbf{x})\mu_{\nu_{r,j}}(C(\mathbf{x}))$ from (25), we get that the basis functions of DFR-Sarsa(λ) are bounded.

Secondly, we prove the basis function vector is linearly independent. In order to make the basis function vector linearly independent, let the basis functions meet Condition 1 [21], where the function form is shown in Figure 4. From literature [21] we know that, when Condition 1 is met, the basis function vector is linearly independent.

The requirement in Condition 1 can be relaxed appropriately by making the membership degree of $\mu_i(x_i)$ at state x_i a small value, for example, a Gaussian membership function with smaller standard deviation. Applying the membership function to DFR-Sarsa(λ), experimental results show that DFR-Sarsa(λ) is convergent, though the convergence still cannot be given theoretically.

In summary, Lemma 4 is proved. \square

Lemma 5. Step-size parameter α of DFR-Sarsa(λ) satisfies (26)

$$\sum_{t=0}^{\infty} \alpha_t = \infty, \quad \sum_{t=0}^{\infty} \alpha_t^2 < \infty. \quad (26)$$

Proof. Set step-size parameter of DFR-Sarsa(λ) $\alpha = 1/(t+1)$, where t is the time step. By Newton power series expansion, we get

$$\sum_{t=0}^{\infty} \alpha_t = \sum_{t=0}^{\infty} \left(1 + \frac{1}{2} + \cdots + \frac{1}{t} \right) = \ln(t+1) + r, \quad (27)$$

where $r = 0.577218$ is Euler's constant. Because $\ln t$ is an increasing function, it satisfies $\sum_{t=0}^{\infty} \alpha_t = \infty$ when $t \rightarrow \infty$.

Cosider

$$\sum_{t=0}^{\infty} \alpha_t^2 = \sum_{t=0}^{\infty} \left(1^2 + \left(\frac{1}{2} \right)^2 + \cdots + \left(\frac{1}{t} \right)^2 \right) < \frac{2t-1}{t} = 2 - \frac{1}{t}, \quad (28)$$

the inequality part in Inequation (28) can be proven by induction; thus $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$ is met when $t \rightarrow \infty$.

By (27) and Inequation (28), we get that the step-size parameter of DFR-Sarsa(λ) satisfies (26); thus we proved Lemma 5. \square

Theorem 6. Under the condition of Assumption 2, if DFR-Sarsa(λ) satisfies Lemma 3 to Lemma 5, the algorithm converges with probability 1.

Proof. Literature [18] gives the related conclusion that, under the condition of Assumption 2, when on-policy TD(λ) algorithms with linear function approximation meet certain conditions (Lemma 3 to Lemma 5), the algorithms converge with probability 1. DFR-Sarsa(λ) is just such an algorithm and it meets Assumption 2 and Lemma 3 to Lemma 5. So we get that DFR-Sarsa(λ) converges with probability 1. \square

4. Experiments

In order to verify DFR-Sarsa(λ)'s performance about the convergence rate, iteration steps after convergence, and the

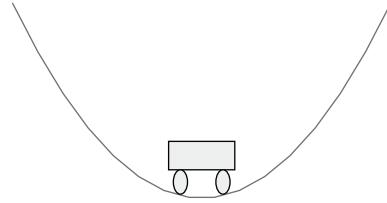


FIGURE 2: Mountain Car.

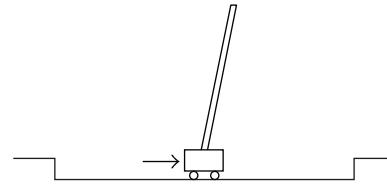


FIGURE 3: Cart-pole Balancing.

effectiveness of continuous action policy, we take two problems as experimental benchmarks: Mountain Car and Cart-pole Balancing. These two problems are classic episodic tasks with continuous state and action spaces in reinforcement learning, which are shown in Figures 2 and 3, respectively.

4.1. Mountain Car. Mountain Car is a representative problem with continuous state space, as shown in Figure 2. Suppose the underpowered car cannot accelerate up directly to reach the top of the right side. So, it has to move around more than once to get there. Modeling the task as an MDP, in which the state represented a two-dimensional variable: location and speed; that is, $x = [y, v]$. The action is the force that drives the car to move horizontally, which is bounded in $[-1, 1]$. In this problem, the system dynamics are described as follows:

$$\begin{aligned} v_{t+1} &= \text{bound}[v_t + 0.001 u_t + g \cos(3y_t)], \\ y_{t+1} &= \text{bound}[y_t + 1], \end{aligned} \quad (29)$$

where $\text{bound}(v_t) \in [-0.07, +0.07]$, $\text{bound}(y_t) \in [-1.5, +0.5]$, and $g = 0.0025$ is a constant related to gravity. In addition, time step is 0.1 s and the reward function is as follows:

$$r_t = \begin{cases} -1, & y < 0.5, \\ 0, & y \geq 0.5. \end{cases} \quad (30)$$

Equation (30) is a punishment reward function, where r_t means the reward received at time t .

In the simulation, the number of episodes is set to 1000. The maximum time step in each episode is also set to 1000. The initial state of the car is $y = -0.5$, $u = 0$. When the car arrives to the destination ($y = 0.5$) or the time steps exceed 1000, we finish this episode and begin a new one. The experiment will end after 1000 episodes.

In order to show the effectiveness of DFR-Sarsa(λ), we compare the algorithm with Fuzzy Sarsa proposed by Tokarchuk et al. [15], GD-Sarsa(λ) proposed by Sutton et al. [3], and Fuzzy Q(λ) proposed by Zajdel [22]. Additionally, the effect of eligibility trace on the convergence performance is also tested.

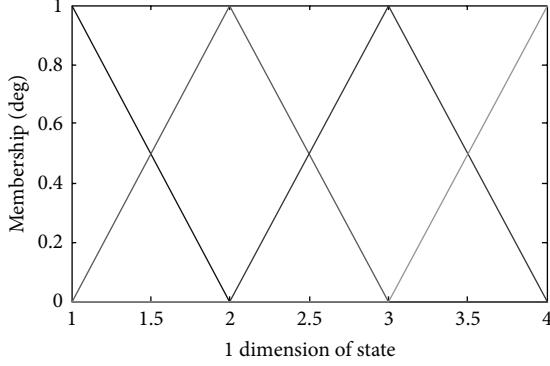


FIGURE 4: Triangular membership functions (except the domains of states are different, the form of membership functions of position, and velocity is described in Figure 4. Besides, the form of membership functions in Section 4.2 is also the same).

At present, there is no proper way to select parameters that make the four algorithms have their best performance, respectively. In order to make the comparison more reasonable, the parameters that exist in all of the four algorithms will be set at the same value, while the parameters that do not exist in all of the four algorithms will be set at the value from where it firstly comes.

We first set the parameters of DFR-Sarsa(λ): 20 triangular fuzzy sets whose cores are equidistant are used to partition each state variable, which results in 400 fuzzy rules. Similarly, use eight triangular fuzzy sets whose cores are equidistant to partition the continuous action space, where the number of fuzzy rules is 8. Set the other parameters $\varepsilon = 0.001$, $\alpha = 0.9$, $\lambda = 0.9$, and $\gamma = 1.0$. The form of fuzzy partition in Fuzzy Sarsa is the same as in DFR-Sarsa(λ). Other parameters are set to $\varepsilon = 0.001$, $\alpha = 0.9$, and $\gamma = 0.9$. GD-Sarsa(λ) uses 10 tilings of 9×9 to divide state space, where the parameters are set as the best experimental parameters given in literature [1]: $\varepsilon = 0.001$, $\alpha = 0.14$, $\lambda = 0.3$, and $\gamma = 1.0$. The form of fuzzy partition in Fuzzy Q(λ) Sarsa is also the same as in DFR-Sarsa(λ). Other parameters are set in accordance with literature [22] to $\varepsilon = 0.005$, $\alpha = 0.1$, $\lambda = 0.1$, and $\gamma = 0.995$.

DFR-Sarsa(λ), Fuzzy Sarsa, GD-Sarsa(λ), and Fuzzy Q(λ) are applied to Mountain Car. Figure 5 shows the average result in 30 independent simulation experiments. The x -coordinate indicates the number of episodes, and y -coordinate represents the average time steps the car drives from the initial state to the target. As can be seen from Figure 5, the convergence performance of DFR-Sarsa(λ) is better than those of the other three algorithms.

The detailed performance of the four algorithms is shown in Table 1 (the benchmark time is the average time of a single iteration of DFR-Sarsa(λ)).

In order to test the effectiveness of the proposed eligibility trace, DFR-Sarsa(λ) with eligibility trace and DFR-Sarsa without eligibility trace are both applied in Mountain Car. Figure 6 shows the convergence performance of these two algorithms. It can be seen that these two algorithms converge in the same average time steps, but the convergence speed of DFR-Sarsa(λ) is better than that of DFR-Sarsa.

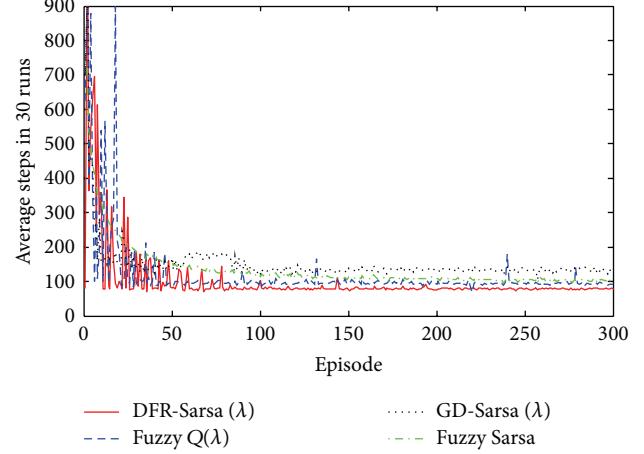


FIGURE 5: Comparisons on convergent efficiency of the four algorithms.

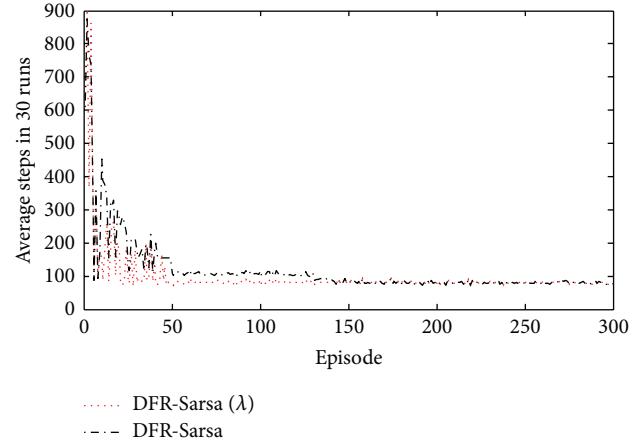


FIGURE 6: The effect of eligibility traces on convergent efficiency of DFR-Sarsa(λ).

TABLE 1: Performance comparison of the four algorithms in Mountain Car problem.

Algorithm	DFR-Sarsa (λ)	Fuzzy Q (λ)	GD-Sarsa (λ)	Fuzzy Sarsa
Average episodes	86	98	118	103
Average steps	79	101	134	112
Average time within an iterative step	100%	80%	30%	65%

4.2. Cart-Pole Balancing. Figure 3 shows a Cart-pole Balancing system, in which the cart can move left or right on the horizontal plane. A pole is hinged to the cart, which can rotate freely within a certain angle. The task is to move the cart horizontally to keep the pole standing in a certain range $[-\pi/2, \pi/2]$. Similarly, modeling the task as an MDP, the state is a two-dimensional variable, which is represented by the vertical angle of pole θ , and the angular velocity of the pole $\dot{\theta}$; that is, $x = [\theta, \dot{\theta}]$. These two state variables satisfy $\theta \in [-\pi/2, \pi/2]$ (rad) and $\dot{\theta} \in [-16\pi, 16\pi]$ (rad/s). The action

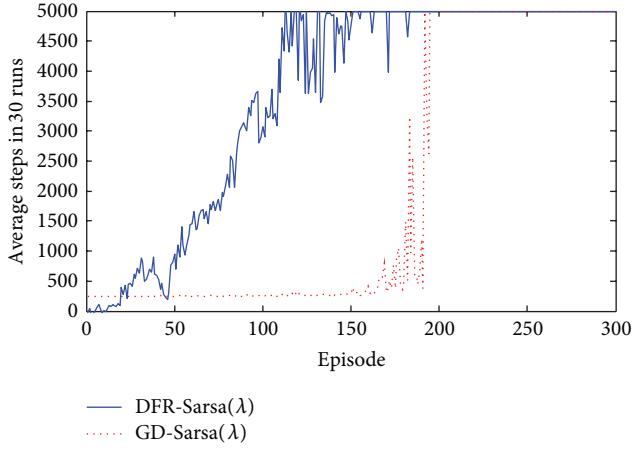


FIGURE 7: Comparisons on convergent efficiency of the two algorithms.

is the force exerted on the cart, which ranges from -50 N to 50 N . In addition, the force is added by a noise force which is uniformly distributed in $[-10 \text{ N}, 10 \text{ N}]$. The system dynamics are described as

$$\ddot{\theta} = \frac{g \sin(\theta) - aml(\dot{\theta})^2 \sin(2\theta)/2 - \alpha \cos(\theta) u}{4l/3 - aml \cos^2(\theta)}, \quad (31)$$

where $g = 9.8 \text{ m/s}^2$ is acceleration of gravity, $m = 2.0 \text{ kg}$ is the mass of pole, $M = 8.0 \text{ kg}$ is the mass of cart, $l = 0.5 \text{ m}$ is the length of pole, and constant $\alpha = 1/(m + M)$. The change of reward depends on the change of state. At each time step (0.1 s), when the angle of the pole with vertical direction is no more than $\pi/2$, reward 0 is received. While the angle is more than $\pi/2$, the reward is -1 , and the episode ends.

The parameter setting in this example is similar to the settings in Section 3.1, so we only give the difference here: 12 equidistant triangular fuzzy sets are used to partition the continuous action space, which leads to 144 fuzzy rules.

DFR-Sarsa(λ) and GD-Sarsa(λ) are executed on 30 independent simulations on Cart-pole Balancing; the results are shown in Figure 7, where the x -coordinate represents the number of episodes; the y -coordinate represents the average time steps. As can be seen from Figure 7, the convergence performance of DFR-Sarsa(λ) is also better than GD-Sarsa(λ).

The detailed performance of the two algorithms is shown in Table 2 (the benchmark time is the average time of a single iteration of DFR-Sarsa(λ)).

Figure 8 shows the results of GD-Sarsa(λ) and DFR-Sarsa(λ) on Cart-pole Balancing task, respectively. We have known that GD-Sarsa(λ) is based on discrete action policies, while DFR-Sarsa(λ) is based on continuous action policies. From Figure 8 we can see that the continuous action policy obtained by DFR-Sarsa(λ) can make the pole's angle change in only a small angle, while discrete action policy obtained by GD-Sarsa(λ) makes the pole's angle change in a large range. This fact indicates that policies obtained by DFR-Sarsa(λ) are much more stable than that of GD-Sarsa(λ). Thus, DFR-Sarsa(λ) is more suitable for applications which require more stable policies.

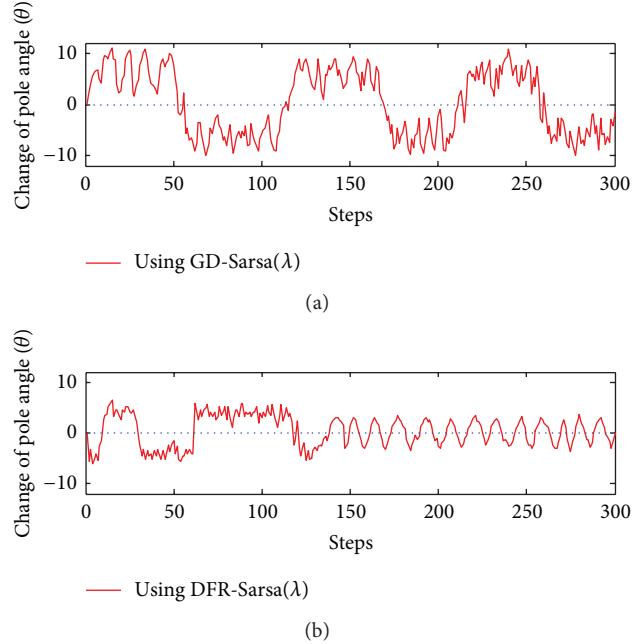


FIGURE 8: Changes of the pole angle θ when using the two mentioned algorithms, respectively.

TABLE 2: Performance comparison of the two algorithms in Cart-pole Balancing problem.

Algorithm	Episodes		Average time within an iterative step
	Minimum episodes	Average episodes	
DFR-Sarsa(λ)	135	155	100%
GD-Sarsa(λ)	179	204	46%

5. Conclusions

In allusion to the problem that classic reinforcement learning algorithms based on lookup table or function approximation converge slowly and are difficult to obtain continuous action policies, this paper presents an algorithm with eligibility trace based on double-layer fuzzy reasoning—DFR-Sarsa(λ). Firstly, the algorithm constructs two fuzzy reasoning layers to approximate Q -function, which are associated with state, action, and Q -value. Then, it uses gradient descent method to update eligibility trace and the consequent of fuzzy rules in the two FISs. Applying the proposed algorithm and other three similar relatively new algorithms to Mountain Car and Cart-pole Balancing system, experimental results show that, compared with reinforcement learning algorithms using only one fuzzy inference system, our algorithm requires fewer steps to convergence, though it increases the time complexity; compared with algorithms based on lookup table or some other function approximation methods, DFR-Sarsa(λ) has better convergence performance and can obtain a continuous action policy.

The performance of DFR-Sarsa(λ) relies on the two-layer fuzzy inference systems, while the performance of fuzzy

inference system mainly depends on the fuzzy sets and fuzzy rules. In this paper, the type of fuzzy sets and the number of rules are given as prior knowledge, and they are no longer changed during the learning process. In order to achieve a much better convergence performance, we will focus on using appropriate optimization algorithms to optimize the membership functions and adjust the fuzzy rules adaptively.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61070223, 61103045, 61070122, 61272005, and 61170314), the Natural Science Foundation of Jiangsu Province (BK2012616), the Foundation of Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University (93K172012K04).

References

- [1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, Mass, USA, 1998.
- [2] Q. Yan, Q. Liu, and D. Hu, “A hierarchical reinforcement learning method based on heuristic reward function,” *Journal of Computer Research and Development*, vol. 48, no. 12, pp. 2352–2358, 2011.
- [3] R. S. Sutton, D. McAllester, S. Singh et al., “Policy gradient methods for reinforcement learning with function approximation,” in *Proceedings of the 16th Annual Conference on Neural Information Processing Systems*, pp. 1057–1063, Denver, Colo, USA, 1999.
- [4] H. R. Maei and R. S. Sutton, “GQ(λ): a general gradient algorithm for temporal-difference prediction learning with eligibility traces,” in *Proceedings of the 3rd Conference on Artificial General Intelligence (AGI '10)*, pp. 91–96, Atlanta, Ga, USA, March 2010.
- [5] R. S. Sutton, C. Szepesvári, and H. R. Maei, “A convergent O(n) algorithm for off-policy temporal-difference learning with linear function approximation,” in *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS '08)*, pp. 1609–1616, Vancouver, Canada, December 2008.
- [6] A. A. Sherstov and P. Stone, “Function approximation via tile coding: automating parameter choice,” in *Proceedings of the 5th Symposium on Abstraction, Reformulation and Approximation*, pp. 194–205, New York, NY, USA, 2005.
- [7] M. R. Heinen and P. M. Engel, “An incremental probabilistic neural network for regression and reinforcement learning tasks,” in *Proceedings of the 20th International Conference on Artificial Neural Networks*, pp. 170–179, Berlin, Germany, 2010.
- [8] M. Geist and O. Pietquin, “Parametric value function approximation,” in *Proceedings of the IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL '11)*, pp. 9–16, New York, NY, USA, April 2011.
- [9] J. Pazis and M. G. Lagoudakis, “Learning continuous-action control policies,” in *Proceedings of the IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL '09)*, pp. 169–176, Nashville, Tenn, USA, April 2009.
- [10] A. Bonarini, A. Lazaric, F. Montrone, and M. Restelli, “Reinforcement distribution in fuzzy Q-learning,” *Fuzzy Sets and Systems*, vol. 160, no. 10, pp. 1420–1443, 2009.
- [11] P. Yves Glorennec, “Fuzzy Q-learning and dynamical fuzzy Q-learning,” in *Proceedings of the 3rd IEEE Conference on Fuzzy Systems*, pp. 474–479, Orlando, Fla, USA, June 1994.
- [12] C. H. Hsu and C. F. Juang, “Self-organizing interval type-2 fuzzy Q-learning for reinforcement fuzzy control,” in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '11)*, pp. 2033–2038, Anchorage, Alaska, USA, October 2011.
- [13] T. Horiuchi, A. Fujino, O. Katai, and T. Sawaragi, “Fuzzy interpolation-based Q-learning with continuous states and actions,” in *Proceedings of the 5th IEEE International Conference on Fuzzy System*, pp. 594–600, Los Angeles, Calif, USA, September 1996.
- [14] P. Y. Glorennec and L. Jouffe, “Fuzzy Q-learning,” in *Proceedings of the 6th IEEE International Conference on Fuzzy Systems*, pp. 659–662, Cambridge, Mass, USA, July 1997.
- [15] L. Tokarchuk, J. Bigham, and L. Cuthbert, “Fuzzy sarsa: an approach to linear function approximation in reinforcement learning,” in *Proceedings of the International Conference on Artificial Intelligence and Machine Learning*, pp. 1021–1026, Berlin, Germany, 2005.
- [16] H. S. Chang, M. C. Fu, J. Hu, and S. I. Marcus, *Simulation-Based Algorithms for Markov Decision Processes*, Springer, New York, NY, USA, 2007.
- [17] O. Castillo and P. Melin, *Type-2 Fuzzy Logic: Theory and Applications*, Springer, New York, NY, USA, 2008.
- [18] J. N. Tsitsiklis and B. Van Roy, “An analysis of temporal-difference learning with function approximation,” *Proceedings of the IEEE Transactions on Automatic Control*, vol. 42, no. 5, pp. 674–690, 1997.
- [19] P. Dayan, “The convergence of TD(λ) for general λ ,” *Machine Learning*, vol. 8, no. 3-4, pp. 341–362, 1992.
- [20] L. Cihua, *Stochastic Process*, Huazhong University of Science and Technology Press, Wuhan, China, 2008.
- [21] B. Lucian, B. Robert, S. Bart De, and E. Damien, *Reinforcement Learning and Dynamic Programming Using Function Approximation*, CRC Press, Boca Raton, Fla, USA, 2010.
- [22] R. Zajdel, “Fuzzy Q(λ)-learning algorithm,” in *Proceedings of the 10th International Conference on Artificial Intelligence and Soft Computing*, pp. 256–263, Berlin, Germany, 2010.

Research Article

Semophysical Modelling of the Nonlinear Dynamics of a Surface Craft with LS-SVM

David Moreno-Salinas,¹ Dictino Chaos,¹ Eva Besada-Portas,² José Antonio López-Orozco,² Jesús M. de la Cruz,² and Joaquín Aranda¹

¹ Departament of Computer Science and Automatic Control, National University Distance Education (UNED), Madrid, Spain

² Department of Computer Architecture and Automatic Control, Universidad Complutense de Madrid (UCM), Madrid, Spain

Correspondence should be addressed to David Moreno-Salinas; dmoreno@dia.uned.es

Received 12 July 2013; Accepted 31 October 2013

Academic Editor: Siddhivinayak Kulkarni

Copyright © 2013 David Moreno-Salinas 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 of the most important problems in many research fields is the development of reliable mathematical models with good predictive ability to simulate experimental systems accurately. Moreover, in some of these fields, as marine systems, these models play a key role due to the changing environmental conditions and the complexity and high cost of the infrastructure needed to carry out experimental tests. In this paper, a semophysical modelling technique based on least-squares support vector machines (LS-SVM) is proposed to determine a nonlinear mathematical model of a surface craft. The speed and steering equations of the nonlinear model of Blanke are determined analysing the rudder angle, surge and sway speeds, and yaw rate from real experimental data measured from a zig-zag manoeuvre made by a scale ship. The predictive ability of the model is tested with different manoeuvring experimental tests to show the good performance and prediction ability of the model computed.

1. Introduction

The availability of tools and methods to compute mathematical models for simulation purposes is of key importance, making the system identification field one of the highlights among the research topics in engineering and one of the most important stages in the control research area. Moreover, aspects that go from the high cost of practical implementations and tests in an open air environment to the complexity of the infrastructure needed to carry out experimental tests, and even the changing environmental conditions, call for the availability of these mathematical models with which new designs and ideas can be tested in simulation with high accuracy. In addition, the importance of the modelling stage is crucial since an inadequate model identification may yield large prediction errors. The literature on linear and nonlinear system identification is extensive and covers many areas of engineering research. For a short survey on some essential features in the identification area and a classification of methods the reader is referred to [1, 2].

Among the number of techniques and methods on system identification, semophysical modelling presents some interesting own characteristics. In this method, the prior knowledge about the application is used to develop a good model structure to be defined with raw measurements. The model structures defined are not physically complete, but allow for very suitable models to describe the behaviour of the systems involved, [3]. There exist several tools to be applied on semophysical modelling [4], and potential tools that can support the process of semophysical modelling are neural networks and support vector machines (SVM). The work at hand is focused on the semophysical modelling (based on SVM) of a surface craft to describe its dynamical behaviour in an experimental environment. The computation of the model is carried out with the least-squares support vector machines (LS-SVM) technique [5], which is one of the different types of SVM algorithms available in the literature.

One of the most popular and widely used techniques in the artificial intelligence (A.I.) field for system identification is the artificial neural networks, such as multilayer perceptron

(MLP); see, for example, [6]. This kind of techniques is robust and effective in many problems in identification and control. Despite this, they present some disadvantages such as the local minima, overfitting, large computation time to converge to the solution, and so forth, to name but a few. Some of these problems can be solved effectively using SVM since it provides a larger generalisation performance, offering a more attractive alternative for the system identification problem, as it is not based on the empirical error implemented in neural networks, but on the structural risk minimization (SRM). The basic idea of SVM is Vapnik-Chervonenkis (VC) theory which defines a measure of the capacity of a learning machine [7]. The idea is to map input data into a high-dimensional feature Hilbert space using a nonlinear mapping technique, that is, the kernel dot product trick [8], and to carry out linear classification or regression in feature space. The Kernel functions replace a possibly very high-dimensional Hilbert space without explicitly increasing the feature space [9]. SVM, both for regression and classification, has the ability to simultaneously minimize the estimation error in the training data (the empirical risk) and the model complexity (the structural risk) [10]. Moreover, SVM can be designed to deal with sparse data, where we have many variables but few data. Furthermore, the solution of SVM is globally optimal. The formulation of SVM for regression, that is, support vector regression (SVR), is very similar to the formulation of SVM for classification. For a survey on SVR the reader is referred to [11] and the references therein. The formulation of SVR shows how this technique is suitable to be used as a semophysical modelling tool to obtain the parameters of a mathematical model. In this sense, it is of practical interest to describe a nonlinear system from a finite number of input and output measurements.

Although there are not many results for system identification using SVM, we can find some interesting works such as the work in [12], where the authors make a study of the possible use of SVM for system identification, in [13] where an identification method based on SVR is proposed for linear regression models, or in [14] in which the application of SVM to time series modelling is considered by means of simulated data from an autocatalytic reactor. Other interesting works can be found in [15–18]. It is important to remark that most of the papers that study the problem of system identification using SVM deal only with simulation data. As mentioned above, among the different SVM techniques, we can find LS-SVM, [5]. This technique allows a nice simplification of the problem making it more tractable, as will be commented in higher detail in Section 2. We can also find some interesting works that deal with this problem, for example [19], where LS-SVM is used for nonlinear system identification for some simple examples of nonlinear autoregressive with exogenous input (NARX) input-output models. There also exist some other representative examples that deal with identification using LS-SVM; see, for example, [20–22].

In this paper LS-SVM is used for the semophysical modelling of a surface marine vessel. System identification of marine vehicles starts in the 70s with the works in [23], where an adaptive autopilot with reference model was presented, and in [24], where parametric linear identification techniques

were used to define the guidance dynamics of a ship using the maximum likelihood method. There exist many algorithms and tools to compute mathematical models that describe the dynamics of marine vehicles for different applications and scenarios. For instance, in [25] several parametric identification algorithms are used to design autopilots for different kinds of ships, in [26] the hydrodynamic characteristics of a ship are determined by a Kalman filter (KF), and in [27] an extended kalman filter (EKF) is used for the identification of the ship dynamics for dynamic positioning. The computation of these models usually needs a lot of time and practical tests to obtain enough information about the hydrodynamic characteristics of the vehicle and an important computational effort to define an accurate model, so it is clear that the identification task may become a complex and tedious task. Moreover, the operational conditions may affect the vehicle providing different models depending on these conditions, as studied in [28]. For some other interesting related works the reader is referred to [29–31] and the references therein.

For the above reason, in some practical situations, it is usual to employ simple vehicle models that, although they reproduce with less accuracy the dynamics of the vehicle, they show very good results and prediction ability for most of the standard operations; see, for example, [32] where the authors obtain a linear second-order Nomoto heading model with an added autoregressive moving average (ARMA) disturbance model for an autonomous in-scale physical model of a fast-ferry. They use a turning circle manoeuvre for the system identification. See also [33] where a nonlinear ship model is identified in towing tests in a marine control laboratory for automatic control purposes. Following this trend, in this paper the nonlinear Blanke model is identified [30, 34]. This model has a large prediction ability in the experimental setup for standard operations, as will be seen throughout this work, although the model is less precise than other models available in the literature, such as the Abkowitz model [35]. Furthermore, it can be obtained with semophysical modelling techniques based on SVM in a fast manner with relatively few data.

We can find some works that employ neural networks to define the dynamics of a surface marine vehicle, such as [36–38] or [39]. We can also find some interesting works that deal with the identification of marine vehicles by using SVM, for example, [40], where an Abkowitz model for ship manoeuvring is identified by using LS-SVM, and [41], where ϵ -SVM is employed for the computation of the same model. These two above works search to determine the hydrodynamic coefficients of a mariner class vessel with simple training manoeuvres; however, the identification of the mathematical model is made with data obtained from simulation, and the prediction ability of the model is also tested only in simulation. These works do not deal with real data. Furthermore, as far as the authors know, most of the works that deal with system identification using some SVR technique employ simulation data and numerical examples, where the models obtained are not tested on an experimental setup. Two exceptions are the works [42], in which the steering equations of a Nomoto second-order linear model with constant surge speed are identified using LS-SVM and tested

in an experimental setup with a scale ship model, and [43], in which an identification method based on SVM is proposed for modelling nonlinear dynamics of a torpedo AUV. In this reference the authors determine the hydrodynamic model with a series of captive model tests, and based on this experimental model, manoeuvring simulations are developed. Then SVM is used to identify the damping terms and Coriolis and centripetal terms by analysing the simulation data. In the work at hand, following a similar methodology to that explained in [40, 42], we seek to determine the nonlinear model of Blanke from raw data obtained from a physical scale ship and to validate the model with several experimental tests.

Therefore, the key contributions of the present paper are twofold: (i) the mathematical nonlinear Blanke model of a scale ship is computed from experimental data collected from a 20/20 degree zig-zag manoeuvre with the LS-SVM technique; (ii) the prediction ability of the mathematical model is tested on an open air environment with different manoeuvres carried out with the scale ship. These tests allow checking the connection between the mathematical model and the ship, showing how this nonlinear model predicts with large accuracy the actual behaviour of the surface vessel. In this sense, the model can be used to design control strategies to predict the ship behaviour on a simulation environment before its implementation on the real vehicle. It is important to keep in mind, for the experimental results obtained in this paper, that the analytical properties of SVM can be compromised in stochastic problems because the noise generates additional support vectors. However, if the noise ratio is good and the amplitude is limited, the SVM can solve the problem as if it was deterministic [12].

The document is organized as follows. In Section 2 LS-SVM is introduced. The nonlinear model of Blanke and the input and output data for the LS-SVM algorithm are stated in Section 3. In Section 4 the Blanke model obtained from the semophysical modelling is explicitly defined and its prediction ability is tested with some manoeuvres, namely, evolution circles and zig-zags. Finally, the conclusions and a brief discussion of topics for further research are included in Section 5.

2. Least Squares Support Vector Machines for Regression

For the sake of completeness and clarity, in this section LS-SVM is briefly introduced. The notation and concepts of this section follow the explanation in [5]. The interested reader can see also [11] for a report on support vector regression. The basic idea behind SVM is that, using nonlinear mapping techniques, the input data are mapped into a high-dimensional feature space where linear classification or regression is carried out. Consider a model in the primal weight space:

$$y(x) = \omega^T \varphi(x) + b, \quad (1)$$

where $x \in \Re^n$ is the input data, $y \in \Re$ is the output data, b is a bias term for the regression model, ω is a matrix of weights, and $\varphi(\cdot) : \Re \rightarrow \Re^{n_h}$ is the mapping to a high-dimensional Hilbert space, where n_h can be infinite. The optimization

problem in the primal weight space for a given training set $\{x_i, y_i\}_{i=1}^{N_s}$ yields

$$\min_{\omega, b, e} \mathcal{J}(\omega, e) = \frac{1}{2} \omega^T \omega + \gamma \frac{1}{2} \sum_{i=1}^{N_s} e_i^2 \quad (2)$$

subject to

$$y_i = \omega^T \varphi(x_i) + b + e_i, \quad (3)$$

where N_s is the number of samples, e_i are regression error variables, and γ is the regularisation parameter that determines the deviation tolerated from the desired accuracy. The parameter γ must be always positive. The minimization of $\omega^T \omega$ is closely related to the use of a weight decay term in the training of neural networks, and the second term of the right-hand side of (2) controls the tradeoff between the empirical error and the model complexity.

In the above problem formulation ω may become infinite dimensional, and then the problem in the primal weight space cannot be solved. In this situation, the Lagrangian must be computed to derive the dual problem:

$$\mathcal{L}(\omega, b, e, \alpha) = \mathcal{J}(\omega, e) - \sum_{i=1}^{N_s} \alpha_i \{\omega^T \varphi(x_i) + b + e_i - y_i\}, \quad (4)$$

where α_i , with $i = 1, \dots, N_s$, are the Lagrange multipliers. Now the derivatives of (4) with respect to ω , b , e_i , and α_i must be computed to define the optimality conditions:

$$\begin{aligned} & \frac{\partial \mathcal{L}(\omega, b, e, \alpha)}{\partial \omega, b, e, \alpha} \\ \rightarrow & \left\{ \begin{array}{l} \frac{\partial \mathcal{L}}{\partial \omega} = 0 \longrightarrow \omega = \sum_{i=1}^{N_s} \alpha_i \varphi(x_i) \\ \frac{\partial \mathcal{L}}{\partial b} = 0 \longrightarrow \sum_{i=1}^{N_s} \alpha_i = 0 \\ \frac{\partial \mathcal{L}}{\partial e_i} = 0 \longrightarrow \alpha_i = \gamma e_i \\ \frac{\partial \mathcal{L}}{\partial \alpha_i} = 0 \longrightarrow \omega^T \varphi(x_i) + b + e_i - y_i = 0. \end{array} \right. \end{aligned} \quad (5)$$

After straightforward computations, variables ω and e are eliminated from (5), and then the kernel trick is applied. The kernel trick allows us to work in large dimensional feature spaces without explicit computations on them [8]. Thus, the problem formulation yields

$$y(x) = \sum_{i=1}^{N_s} \alpha_i K(x, x_i) + b. \quad (6)$$

In (6) the term $K(\cdot, \cdot)$ represents the kernel function, which involves an inner product between its operands. This kernel must be positive definite and must satisfy the Mercer condition [44]. The equation defined in (6) may be applied now to compute the regression model.

Equation (6) is very similar to that which would be obtained for a standard SVM formulation. The main differences between both formulations are the equality constraints in (3) and the squared error term of the second term in the right-hand side of (2), implying a significant simplification of the problem.

3. Semophysical Modelling of the Nonlinear Model of Blanke

In marine systems, the experimental tests can become costly in time and money, due to the need of deployment, calibration, and recovery of complex systems at sea. Therefore, the number of experimental tests that may be carried out are partially constrained by this reason among others, like environmental conditions, transportation of equipment, and so forth, to name but a few. In this sense, the availability of mathematical models, which describe the real systems accurately, is of utmost importance because most of these experimental tests may be carried out in simulation, predicting with high accuracy the real behaviour of the real systems and saving a number of practical tests.

There exists a wide range of different marine systems that require mathematical models. The problem arises when a detailed and trustworthy mathematical ship model is needed, since it requires the identification of a multitude of hydrodynamic parameters; see [35]. This task can become hard and complex, with the need of multiple experimental tests [30].

In many practical scenarios, it is very usual to employ simple models that predict the behaviour of real ships with large accuracy in most of the standard operations, like the Nomoto models [45]. For example, in [42] the identification of a second-order linear model of Nomoto for control purposes is defined, although this model assumes linear ship dynamics and describes only the steering equations. This kind of model may be insufficient accurate for some scenarios due to its simplicity and thus its use would be seriously limited. It is necessary to compute a more general model to be applied in a wider variety of situations and control actions, and this work tries to overcome this limitation.

Therefore, in the present work a nonlinear manoeuvring model based on second-order modulus functions is employed. The model used is the one proposed by Blanke [34], which is a simplification of the Norrbin's nonlinear model, but with the most important terms for steering and propulsion loss assignment. This 3-degree-of-freedom (DOF) manoeuvring model is defined, following the definition in [30], as

$$\begin{aligned} (m - X_{\dot{u}}) \dot{u} &= X_{|u|u} |u| u + (m + X_{vr}) vr \\ &\quad + (mx_G + X_{rr}) r^2 + (1-t) T + X_{\delta\delta} \delta^2 + X_{\text{ext}} \\ (m - Y_{\dot{v}}) \dot{v} &+ (mx_G - Y_{\dot{r}}) \dot{r} = -(m - Y_{ur}) ur + Y_{uv} uv \\ &\quad + Y_{|v|v} |v| v + Y_{|v|r} |v| r \\ &\quad + Y_{\delta\delta} \delta + Y_{\text{ext}} \end{aligned}$$

$$\begin{aligned} (mx_G - N_{\dot{v}}) \dot{v} + (I_z - N_{\dot{r}}) \dot{r} &= -(mx_G - N_{ur}) ur + N_{uv} uv \\ &\quad + N_{|v|v} |v| v + N_{|v|r} |v| r \\ &\quad + N_{\delta\delta} \delta + N_{\text{ext}}, \end{aligned} \quad (7)$$

where u is surge speed, r is yaw rate, v is sway velocity, δ is the rudder angle, I_z is moment of inertia about the z -axis, m is mass, x_G is the x -axis coordinate of the centre of gravity, t is the thrust deduction number, T is propeller thrust, $X_{\delta\delta}$ is resistance due to rudder deflection, and $X_{\dot{u}}$, $X_{|u|u}$, X_{vr} , X_{rr} , X_{ext} , $Y_{\dot{v}}$, Y_{ur} , $Y_{\dot{r}}$, Y_{uv} , $Y_{|v|v}$, $Y_{|v|r}$, $Y_{\delta\delta}$, Y_{ext} , $N_{\dot{v}}$, $N_{\dot{r}}$, N_{ur} , N_{uv} , $N_{|v|v}$, $N_{|v|r}$, $N_{\delta\delta}$, and N_{ext} are added inertia hydrodynamic coefficients. For more details, the reader is referred to [30].

The interest of this particular model resides in that, despite its relative simplicity, the most important nonlinear terms of the ship dynamics are taken into account. Furthermore, it is possible to compute a dynamic model for control purposes from the experimental data without the need of computing the hydrodynamic derivatives that define all the ship characteristics and its complete behaviour. Therefore, (7) may be rewritten as

$$\begin{aligned} \dot{u} &= \frac{1}{m - X_{\dot{u}}} (X_{|u|u} |u| u + (m + X_{vr}) vr \\ &\quad + (mx_G + X_{rr}) r^2 + (1-t) T \\ &\quad + X_{\delta\delta} \delta^2 + X_{\text{ext}}), \\ \dot{v} &= \frac{I_z - N_{\dot{r}}}{\Theta} (- (m - Y_{ur}) ur + Y_{uv} uv + Y_{|v|v} |v| v \\ &\quad + Y_{|v|r} |v| r + Y_{\delta\delta} \delta + Y_{\text{ext}}) \\ &\quad - \frac{mx_G - Y_{\dot{r}}}{\Theta} (- (mx_G - N_{ur}) ur + N_{uv} uv \\ &\quad + N_{|v|v} |v| v + N_{|v|r} |v| r + N_{\delta\delta} \delta + N_{\text{ext}}), \\ \dot{r} &= \frac{m - Y_{\dot{v}}}{\Theta} (- (mx_G - N_{ur}) ur + N_{uv} uv + N_{|v|v} |v| v \\ &\quad + N_{|v|r} |v| r + N_{\delta\delta} \delta + N_{\text{ext}}) \\ &\quad - \frac{mx_G - N_{\dot{v}}}{\Theta} (- (m - Y_{ur}) ur + Y_{uv} uv + Y_{|v|v} |v| v \\ &\quad + Y_{|v|r} |v| r + Y_{\delta\delta} \delta + Y_{\text{ext}}), \end{aligned} \quad (8)$$

with $\Theta = (I_z - N_{\dot{r}})(m - Y_{\dot{v}}) - (mx_G - Y_{\dot{r}})(mx_G - N_{\dot{v}})$. Now we can proceed with the derivation of the semophysical model. For simplicity reasons, (8) is discretized with Euler's stepping method using a forward-difference approximation on the derivative:

$$\begin{aligned} \frac{u(k+1) - u(k)}{\Delta k} &= \frac{1}{m - X_{\dot{u}}} (X_{|u|u} |u(k)| u(k) \\ &\quad + (m + X_{vr}) v(k) r(k)) \end{aligned}$$

$$\begin{aligned}
& + (mx_G + X_{rr}) r(k)^2 + (1-t) T(k) \\
& + X_{\delta\delta} \delta(k)^2 + X_{\text{ext}}), \\
& \frac{v(k+1) - v(k)}{\Delta k} \\
& = \frac{I_z - N_r}{\Theta} (- (m - Y_{ur}) u(k) r(k) \\
& + Y_{uv} u(k) v(k) + Y_{|v|v} |v(k)| v(k) \\
& + Y_{|v|r} |v(k)| r(k) + Y_\delta \delta(k) + Y_{\text{ext}}) \\
& - \frac{mx_G - Y_r}{\Theta} (- (mx_G - N_{ur}) u(k) r(k) \\
& + N_{uv} u(k) v(k) + N_{|v|v} |v(k)| v(k) \\
& + N_{|v|r} |v(k)| r(k) + N_\delta \delta(k) + N_{\text{ext}}), \\
& \frac{r(k+1) - r(k)}{\Delta k} \\
& = \frac{m - Y_v}{\Theta} (- (mx_G - N_{ur}) u(k) r(k) \\
& + N_{uv} u(k) v(k) + N_{|v|v} |v(k)| v(k) \\
& + N_{|v|r} |v(k)| r(k) + N_\delta \delta(k) + N_{\text{ext}}) \\
& - \frac{mx_G - N_v}{\Theta} (- (m - Y_{ur}) u(k) r(k) \\
& + Y_{uv} u(k) v(k) + Y_{|v|v} |v(k)| v(k) \\
& + Y_{|v|r} |v(k)| r(k) + Y_\delta \delta(k) + Y_{\text{ext}}), \\
& v(k+1) \\
& = v(k) - \Delta k \Theta^{-1} ((I_z - N_r) (m - Y_{ur}) \\
& - (mx_G - Y_r) (mx_G - N_{ur})) \\
& \cdot u(k) r(k) \\
& + \Delta k \Theta^{-1} ((I_z - N_r) Y_{uv} - (mx_G - Y_r) N_{uv}) \\
& \cdot u(k) v(k) \\
& + \Delta k \Theta^{-1} ((I_z - N_r) Y_{|v|v} - (mx_G - Y_r) N_{|v|v}) \\
& \cdot |v(k)| v(k) \\
& + \Delta k \Theta^{-1} ((I_z - N_r) Y_{|v|r} - (mx_G - Y_r) N_{|v|r}) \\
& \cdot |v(k)| r(k) \\
& + \Delta k \Theta^{-1} ((I_z - N_r) Y_\delta - (mx_G - Y_r) N_\delta) \delta(k) \\
& + \Delta k \Theta^{-1} ((I_z - N_r) Y_{\text{ext}} - (mx_G - Y_r) N_{\text{ext}}), \\
& r(k+1) \\
& = r(k) - \Delta k \Theta^{-1} ((m - Y_v) (mx_G - N_{ur}) \\
& - (mx_G - N_v) (m - Y_{ur})) \\
& \cdot u(k) r(k) \\
& + \Delta k \Theta^{-1} ((m - Y_v) N_{uv} - (mx_G - N_v) Y_{uv}) \\
& \cdot u(k) v(k) \\
& + \Delta k \Theta^{-1} ((m - Y_v) N_{|v|v} - (mx_G - N_v) Y_{|v|v}) \\
& \cdot |v(k)| v(k) \\
& + \Delta k \Theta^{-1} ((m - Y_v) N_{|v|r} - (mx_G - N_v) Y_{|v|r}) \\
& \cdot |v(k)| r(k) \\
& + \Delta k \Theta^{-1} ((m - Y_v) N_\delta - (mx_G - N_v) Y_\delta) \delta(k) \\
& + \Delta k \Theta^{-1} ((m - Y_v) N_{\text{ext}} - (mx_G - N_v) Y_{\text{ext}}). \tag{10}
\end{aligned}$$

where Δk is considered to be the sampling time of the inertial measurement unit (IMU) on board the ship, and k and $k+1$ define two successive data measurements from the IMU. We have to rearrange the terms of (9) in a similar way as (6). Doing so, (9) leads to

$$\begin{aligned}
u(k+1) &= u(k) + \frac{\Delta k \cdot X_{|u|u}}{m - X_{\dot{u}}} |u(k)| u(k) \\
&+ \frac{\Delta k \cdot (m + X_{vr})}{m - X_{\dot{u}}} v(k) r(k) \\
&+ \frac{\Delta k \cdot (mx_G + X_{rr})}{m - X_{\dot{u}}} r(k)^2 \\
&+ \frac{\Delta k}{m - X_{\dot{u}}} (1-t) T(k) + \frac{\Delta k \cdot X_{\delta\delta}}{m - X_{\dot{u}}} \delta(k)^2 \\
&+ \frac{\Delta k \cdot X_{\text{ext}}}{m - X_{\dot{u}}},
\end{aligned}$$

Equation (10), following the notation introduced in [5] and in Section 2, can be rewritten in compact form as

$$y_{k\xi} = \Gamma_\xi x_{k\xi} \tag{11}$$

for $k = 1, \dots, N_s - 1$, where $\xi = u, v, r$, $y_{ku} = u(k+1)$, $y_{kv} = v(k+1)$, and $y_{kr} = r(k+1)$ are the output training data for the sampling time k and where the input training data are

$$x_{ku} = [u(k), |u(k)| u(k), v(k) r(k), r(k)^2, T(k), \delta(k)^2, 1]^T,$$

$$\begin{aligned}
x_{kv} &= [v(k), u(k)r(k), u(k)v(k), |v(k)|v(k), \\
&\quad |v(k)|r(k), \delta(k), 1]^T, \\
x_{kr} &= [r(k), u(k)r(k), u(k)v(k), |v(k)|v(k), \\
&\quad |v(k)|r(k), \delta(k), 1]^T
\end{aligned} \tag{12}$$

and with

$$\begin{aligned}
\Gamma_u &= \left[1, \frac{\Delta k \cdot X_{|u|u}}{m - X_{\dot{u}}}, \frac{\Delta k \cdot (m + X_{vr})}{m - X_{\dot{u}}}, \frac{\Delta k \cdot (mx_G + X_{rr})}{m - X_{\dot{u}}}, \right. \\
&\quad \left. \frac{\Delta k}{m - X_{\dot{u}}} (1 - t), \frac{\Delta k \cdot X_{\delta\delta}}{m - X_{\dot{u}}}, \frac{\Delta k \cdot X_{\text{ext}}}{m - X_{\dot{u}}} \right]^T, \\
\Gamma_v &= \left[1, -\Delta k \Theta^{-1} ((I_z - N_r)(m - Y_{ur}) \right. \\
&\quad \left. - (mx_G - Y_r)(mx_G - N_{ur})) \right., \\
&\quad \Delta k \Theta^{-1} ((I_z - N_r)Y_{uv} - (mx_G - Y_r)N_{uv}), \\
&\quad \Delta k \Theta^{-1} ((I_z - N_r)Y_{|v|v} - (mx_G - Y_r)N_{|v|v}), \\
&\quad \Delta k \Theta^{-1} ((I_z - N_r)Y_{|v|r} - (mx_G - Y_r)N_{|v|r}), \\
&\quad \Delta k \Theta^{-1} ((I_z - N_r)Y_\delta - (mx_G - Y_r)N_\delta), \\
&\quad \left. \Delta k \Theta^{-1} ((I_z - N_r)Y_{\text{ext}} - (mx_G - Y_r)N_{\text{ext}}) \right]^T, \\
\Gamma_r &= \left[1, -\Delta k \Theta^{-1} ((m - Y_{\dot{v}})(mx_G - N_{ur}) \right. \\
&\quad \left. - (mx_G - N_{\dot{v}})(m - Y_{ur})) \right., \\
&\quad \Delta k \Theta^{-1} ((m - Y_{\dot{v}})N_{uv} - (mx_G - N_{\dot{v}})Y_{uv}), \\
&\quad \Delta k \Theta^{-1} ((m - Y_{\dot{v}})N_{|v|v} - (mx_G - N_{\dot{v}})Y_{|v|v}), \\
&\quad \Delta k \Theta^{-1} ((m - Y_{\dot{v}})N_{|v|r} - (mx_G - N_{\dot{v}})Y_{|v|r}), \\
&\quad \Delta k \Theta^{-1} ((m - Y_{\dot{v}})N_\delta - (mx_G - N_{\dot{v}})Y_\delta), \\
&\quad \left. \Delta k \Theta^{-1} ((m - Y_{\dot{v}})N_{\text{ext}} - (mx_G - N_{\dot{v}})Y_{\text{ext}}) \right]^T.
\end{aligned} \tag{13}$$

The estimates of the elements in vectors (13) are obtained from LS-SVM regression. In fact, we are interested in the values of the vectors Γ_ξ themselves, regardless of the values of the different hydrodynamic coefficients that appear in (13), because these vectors will define the equations of motion of the ship, and we look for a model for control purposes. However, if we want to know the values of these hydrodynamic coefficients, additional manoeuvres should be carried out to identify some of them independently, and then, together with the model obtained from LS-SVM, those remaining would be computed.

The structure of the mathematical model is known in advance, and elements in vectors (13) are linear in the parameters, so linear regression can be applied and a linear

TABLE 1: Main parameters and dimensions of the real and the scale ships.

Parameter	Real ship	Scale ship
Length between perpendiculares (L_{pp})	74.400 m	4.389 m
Maximum beam (B)	14.200 m	0.838 m
Mean depth to the top deck (H)	9.050 m	0.534 m
Design draught (T_m)	6.300 m	0.372 m



FIGURE 1: Scale ship used in the experimental tests.

kernel $K(x_i, x_j) = (x_i \cdot x_j)$ may be used for the semiphysical modelling:

$$y_{k\xi} = \left(\sum_{i=1}^{N_s} \alpha_{i\xi} x_{i\xi} \right) \cdot x_{k\xi} + b_\xi \tag{14}$$

for $\xi = u, v, r$ and $k = 1, \dots, N_s$. Comparing (14) with (11), after the training process we have

$$\Gamma_\xi = \sum_{i=1}^{N_s} \alpha_{i\xi} x_{i\xi}, \tag{15}$$

where the bias terms b_ξ must be equal to or approximately 0. The support vectors obtained allow to define the parameters of the Blanke model immediately from (13).

4. Experimental Results

The data used for the training of the LS-SVM algorithm were obtained by carrying out a 20/20 degree zig-zag manoeuvre, since it is a simple manoeuvre but enough to define the main characteristics of the ship dynamics. Once the model is defined with the above zig-zag data, its prediction ability must be compared with the real behaviour of the ship for the same commanded input data, namely, surge speed and rudder angle.

The vehicle used for the experimental tests is a scale model in a 1/16.95 scale; see Figure 1. The scale ship, hereinafter referred to as the ship, has the dimensions shown in Table 1, where the dimensions of the real ship that it represents are also shown.

4.1. Semiphysical Modelling of the Surface Craft. The 20/20 degree zig-zag manoeuvre to obtain the training data is carried out with a commanded surge speed of 2 m/s, during

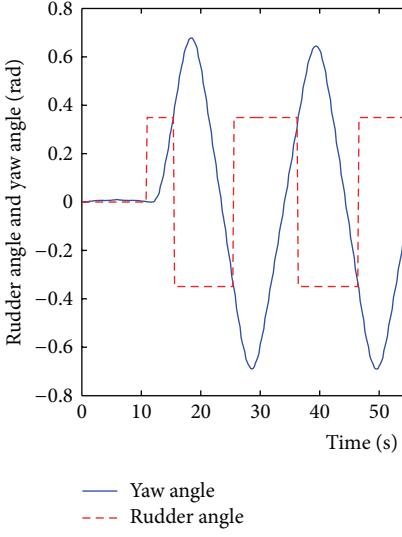


FIGURE 2: 20/20 degree zig-zag manoeuvre. Yaw angle (solid line) and rudder angle (dashed line).

90 seconds. The sampling time is 0.2 seconds, so 450 samples are measured. Figure 2 shows the commanded rudder angle (dashed line) and the corresponding yaw angle (solid line) defined by the vehicle during the 20/20 degree zig-zag manoeuvre. The training data are the commanded control signals or inputs (rudder angle and surge speed) and the data measured from the IMU on board the ship or outputs (effective surge speed, sway speed, and yaw rate). For the sake of clarity on the results shown, the one sigma confidence levels of the measured data from the IMU are heading: 0.05 deg, attitude: 0.025 deg, position: 0.5 m, and velocity: 0.04 m/s.

Now the LS-SVM algorithm for regression may be trained with these input and output data to compute the vectors defined in (13). Different values of the regularisation parameter γ were tested, and $\gamma = 10^4$ was selected as the best candidate. Following the comments made by Blanke [34], the term $(mx_G + X_{rr})$ is considered to be zero since it will be very small for most ships; thus the nonlinear model, after the training process, yields

$$\begin{aligned} \dot{u} &= -0.0321 |u| u - 2.7053 v r + 0.0600 - 0.2257 \delta^2 + 0.06 \\ \dot{v} &= -0.4531 u r - 0.5284 u v + 0.5354 |v| v \\ &\quad - 0.4121 |v| r + 0.0520 \delta + 0.0007 \\ \dot{r} &= -1.1699 u r - 0.6696 u v + 2.3001 |v| v \\ &\quad + 3.9335 |v| r - 0.5503 \delta - 0.0054. \end{aligned} \quad (16)$$

Note that the term $(1 - t)T(k)$ of (8) is constant in (16) since the commanded surge speed is constant for all the experiences carried out in the present work. Once the model is well defined, we must check if it fits correctly the training data; that is, it is necessary to compare the training data with the results obtained with (16) for the same input signals. In Figure 3 the comparison of the semiphysical modelling

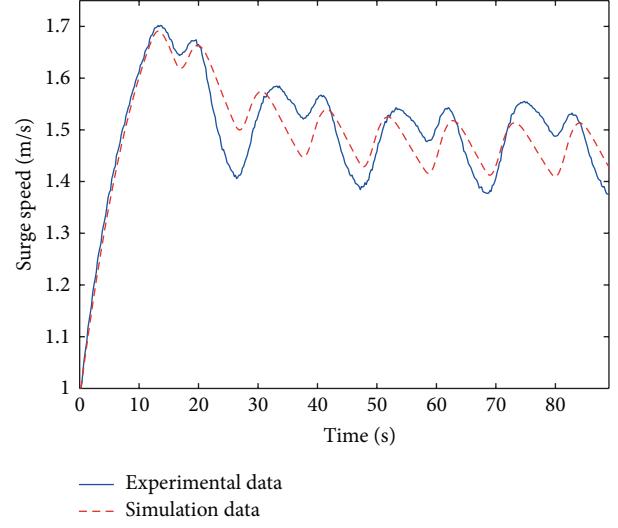


FIGURE 3: Surge speed measured in the zig-zag manoeuvre with the ship (solid line) and in simulation (dashed line).

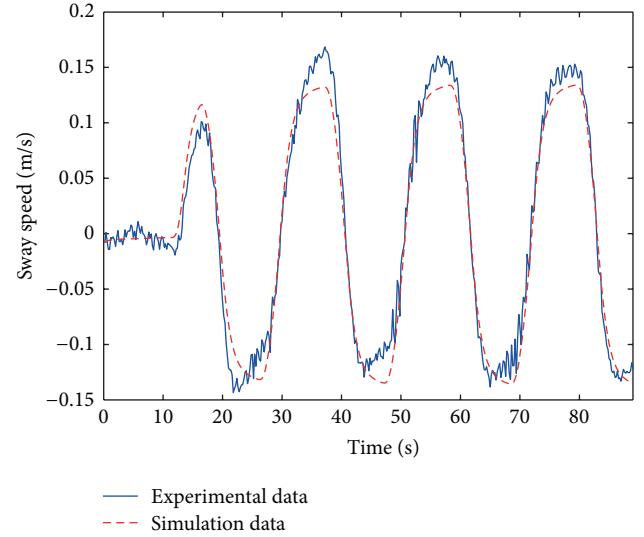


FIGURE 4: Sway speed measured in the zig-zag manoeuvre with the ship (solid line) and in simulation (dashed line).

results with the experimental data for the surge speed is shown. It is important to notice that the scale used in Figure 3 has been chosen to show clearly the difference between the simulated and real surge speeds, but we can see how the maximum error between both speeds is less than 0.1 m/s, and hence, the simulation results are very similar to the real ones.

Similarly, in Figure 4 the sway speed measured from the IMU on board the ship is shown together with the sway speed obtained from the Blanke model defined in (16). Notice how the results are also very similar. Moreover, the large similarity between the real and simulated sway speeds is even more interesting because the sway speed cannot be directly controlled due to the fact that the ship studied is an underactuated vehicle; that is, we have more degrees of freedom

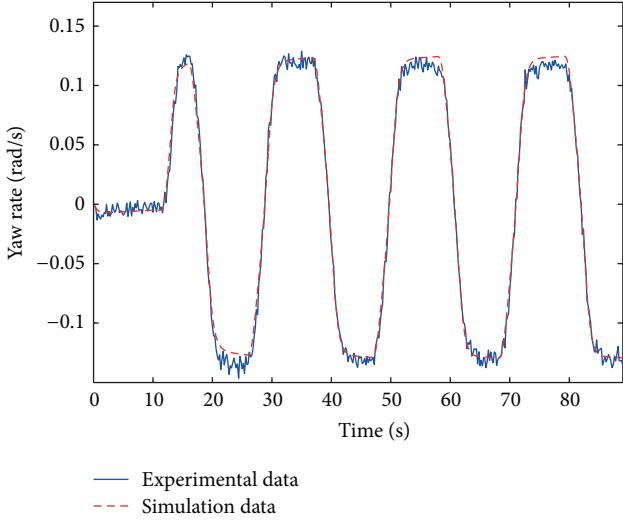


FIGURE 5: Yaw rate obtained in the zig-zag manoeuvre with the ship (solid line) and in simulation (dashed line).

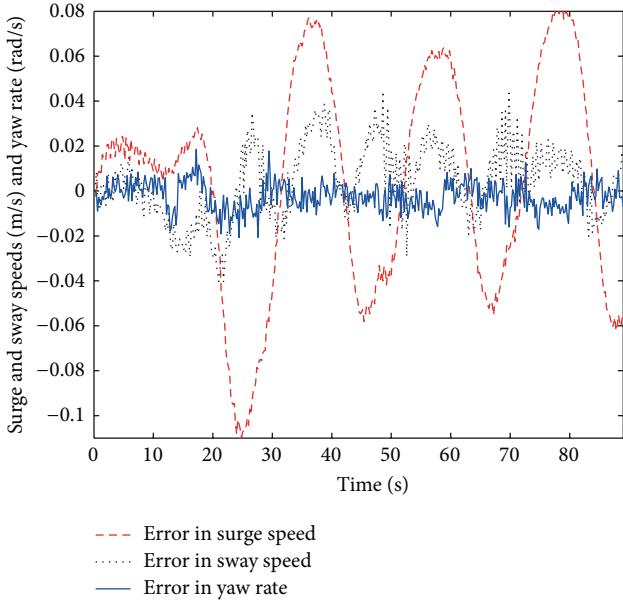


FIGURE 6: Approximation errors in the surge speed (dashed line), sway speed (dotted line), and yaw rate (solid line).

(DOF) than control actions, the latter acting directly on the surge speed and the rudder angle.

Finally, in Figure 5 the IMU and simulated yaw rates are shown. Notice how both curves are practically the same, showing that the model has a dynamical behaviour very close to the actual one of the ship.

For comparison purposes, in Figure 6 the approximation errors for the surge speed, the sway speed, and the yaw rate are shown. It can be seen how the errors are very small and their average values are very close to zero. The standard deviation of the error in the surge speed is 0.0486 m/s and in the sway speed is 0.0171 m/s. For the yaw rate the standard deviation

is 0.0066 rad/s, so it is clear that the simulation model has a behaviour very close to the real one.

4.2. Predictive Ability of the Model. The predictive ability of the model must be tested with different tests and manoeuvres. For this purpose two different manoeuvres are now undertaken. These tests are some turning manoeuvres (evolution circles) and a 10/10 degree zig-zag manoeuvre. The initial values of the effective surge speed, sway speed, and yaw rate used in the simulation tests are the same as those of the real ones to show clearly the connection between the real and the simulated systems.

4.2.1. Test 1: Evolution Circles. The first validation test consists in two turning manoeuvres (evolution circles) for commanded rudder angles of ± 20 deg. The test was run during 240 seconds for each of the turning manoeuvres. In Figures 7(a) and 7(b) we can check the effective surge speed for the ship (solid line) and for the simulation model (dashed line) during these experimental tests for commanded rudder angles of -20 deg and $+20$ deg, respectively. Notice the similar behaviour of both speeds and how the simulated surge speed is smoother than the real one because the simulated model is not affected by noise or disturbances.

In Figures 8(a) and 8(b) the sway speeds for the ship and the simulation model are shown for the commanded rudder angles of -20 deg and $+20$ deg, respectively. It can be seen how both speeds are very similar, although, as mentioned above, the simulated one is free of noise and disturbances.

The yaw rate for the simulated and the actual systems can be studied in Figures 9(a) and 9(b) for the two manoeuvres, where it is shown that the simulation model and the ship have a similar behaviour. In Figures 9 and 7 we can also notice that the real system behaviour is not exactly symmetric; the turnings are slightly larger for negative rudder angles. This nonsymmetrical behaviour is possibly also the reason for the different (small) errors in surge and sway speeds, which vary depending on the turning angle. Despite the above mentioned, the results obtained from the simulated model are very similar to the actual ones, and their difference is not significant. Moreover, the nonsymmetrical behaviour may be caused by environmental conditions, like currents or winds, or by structural characteristics, like the trimming of the ship. This problem does not arise with the semiphysical model since it does not incorporate environmental disturbances that are always present in an experimental setup. Including the possibility of modelling the environmental disturbances would be some interesting future work.

In Figures 10(a) and 10(b) the approximation errors between the real and the predicted surge speed, sway speed, and yaw rate are shown. Notice how the yaw rate error is larger for negative rudder angles as mentioned above. Despite the commented deviation, the errors are small and their average values are close to zero, providing a more than satisfactory prediction of the real dynamical behaviour of the ship. In this sense, the standard deviation of the predicted surge speed with respect to the real one is 0.1786 m/s in the first manoeuvre and 0.04 m/s in the second one. For the sway speed the standard deviations are 0.0303 m/s and 0.0236 m/s,

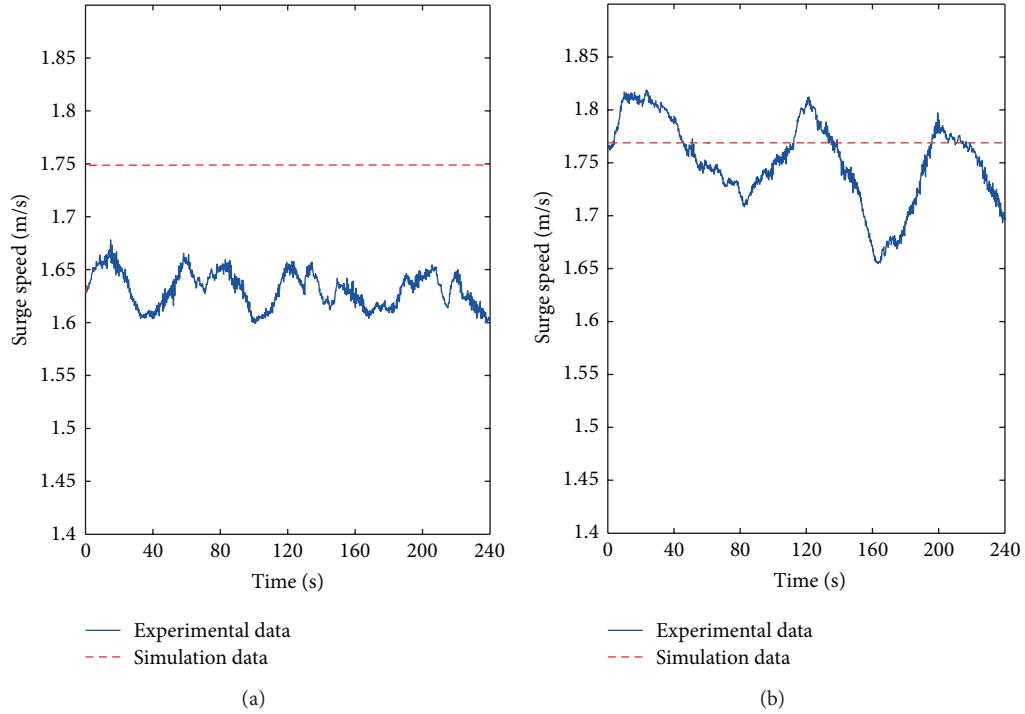


FIGURE 7: Surge speed obtained in two turning manoeuvres with the ship (solid line) and in simulation (dashed line), (a) -20 deg and (b) $+20$ deg.

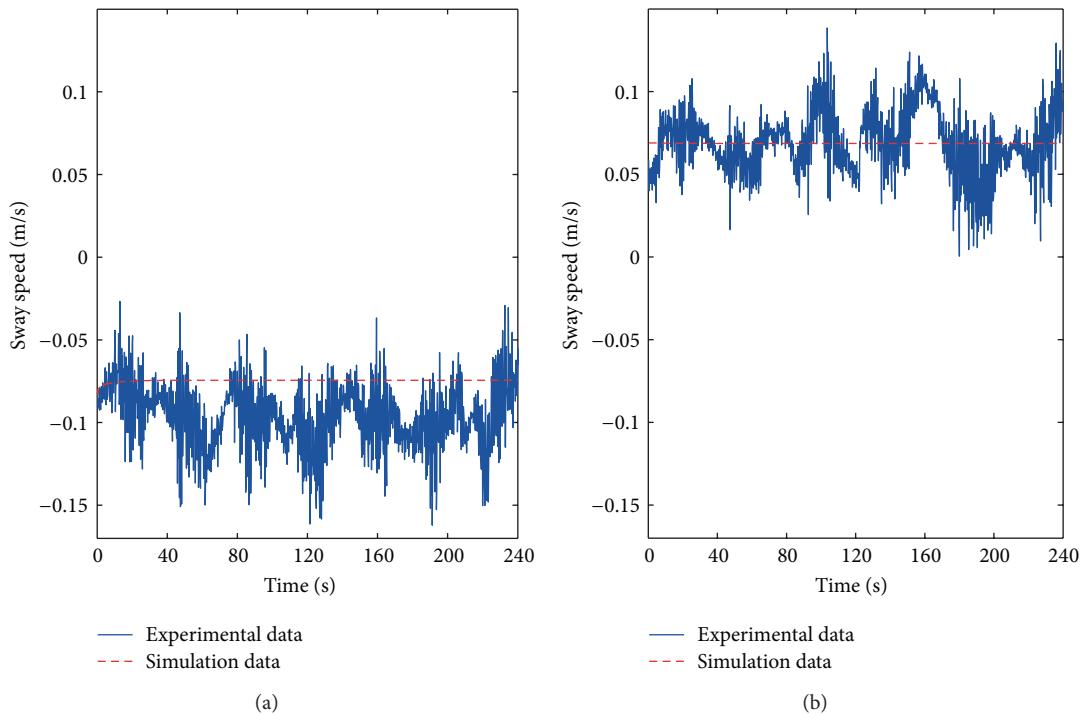


FIGURE 8: Sway speed obtained in two turning manoeuvres with the ship (solid line) and in simulation (dashed line), (a) -20 deg and (b) $+20$ deg.

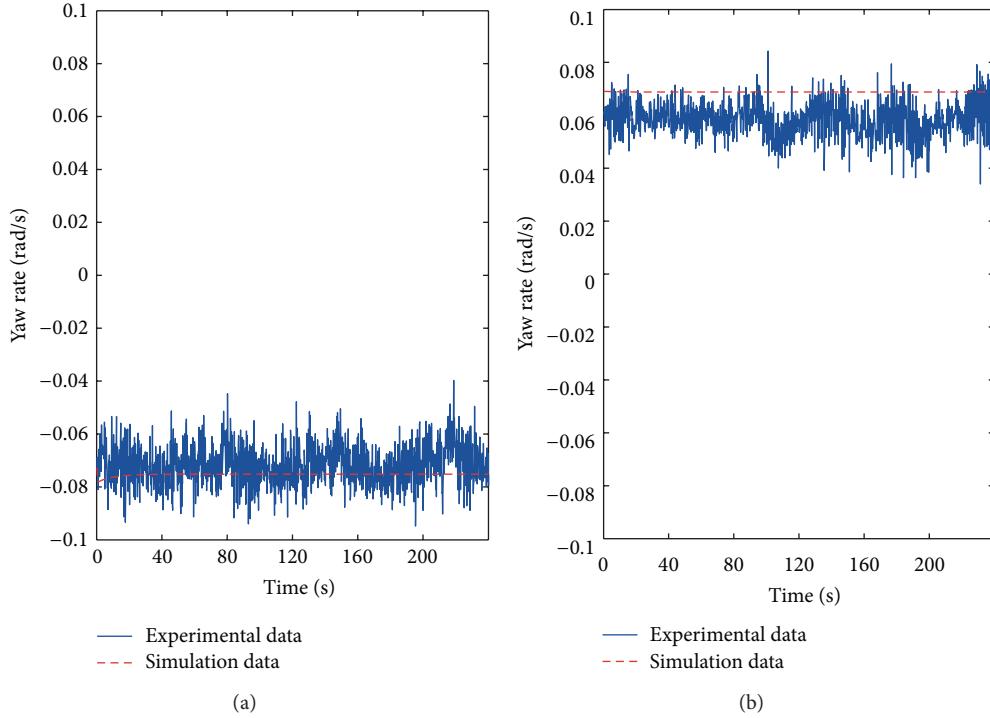


FIGURE 9: Yaw rate obtained in two turning manoeuvres with the ship (solid line) and in simulation (dashed line), (a) -20 deg and (b) $+20$ deg.

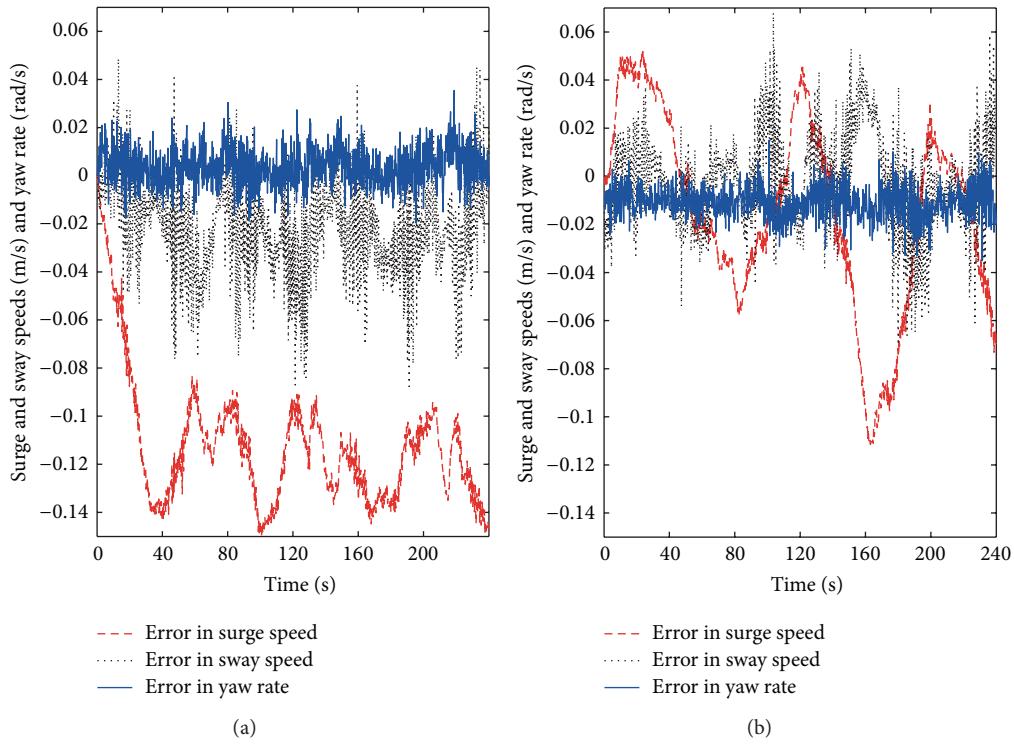


FIGURE 10: Approximation errors in the surge speed (dashed line), sway speed (dotted line), and yaw rate (solid line) for the turning manoeuvres, (a) -20 deg and (b) $+20$ deg.

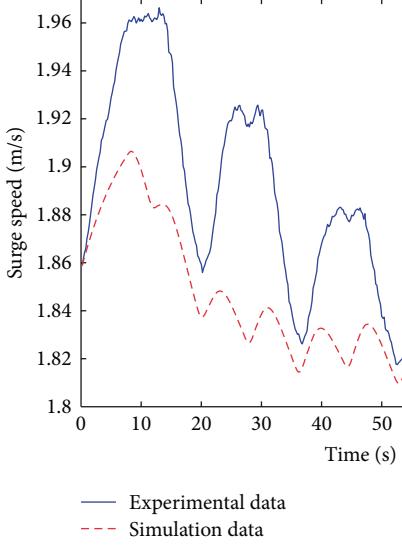


FIGURE 11: Surge speed obtained in a 10/10 degree zig-zag manoeuvre with the ship (solid line) and in simulation (dashed line).

respectively. Finally the yaw rate standard deviations are 0.0160 rad/s and 0.0129 rad/s, respectively. These standard deviations give us a clear image of the slightly nonsymmetrical behaviour of the real ship, showing also how the simulated model has a dynamical behaviour very close to that seen in the real ship.

4.2.2. Test 2: 10/10 Degree Zig-Zag Manoeuvre. In this second test a 10/10 degree zig-zag manoeuvre is carried out to prove the prediction ability of the model. The manoeuvre is run during 90 seconds. In Figure 11 the surge speed during the zig-zag test is shown for both the simulated model and the ship. Notice again that the scale used in Figure 11 has been chosen to show the difference between both speeds and that the maximum error is around 0.06 m/s. Therefore, both speeds are very similar, and the approximation error is very small, as the details in Figure 14 show.

In Figure 12 the sway speed for both systems is shown, and the similarity between both outputs is again easy to check. Finally, in Figure 13 the yaw rate shows that the simulation model obtained with LS-SVM regression has a dynamical behaviour very close to that of the real ship.

In Figure 14 the approximation errors in the surge speed, sway speed, and yaw rate are shown. The standard deviation of the error in the surge speed for this case is 0.0466 m/s, in the sway speed is 0.0239 m/s, and in the yaw rate is 0.0097 rad/s. Hence, the model predicts again the behaviour of the real ship with large accuracy, validating the model obtained with the LS-SVM regression algorithm.

Therefore, it is clear that the nonlinear mathematical model defined for a surface marine vehicle with LS-SVM provides a satisfactory result which predicts with large accuracy the nonlinear dynamics of the experimental system and that it is suitable to be used for control purposes. Thus, this technique has the potential to be implemented for different kinds of marine vehicles in a simple and fast manner, avoiding

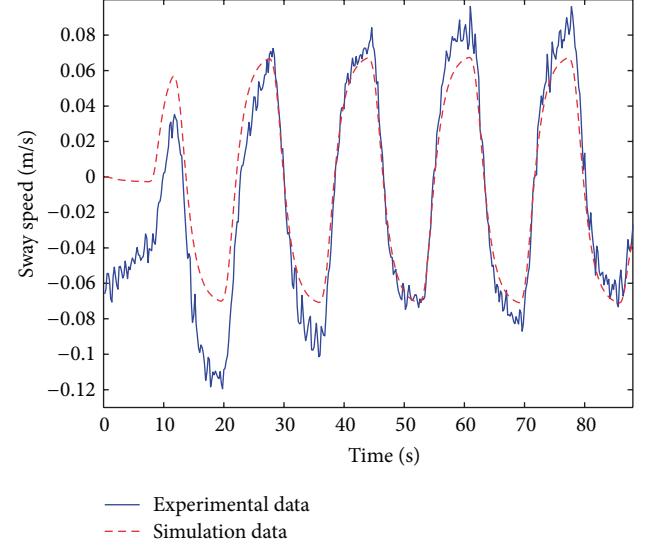


FIGURE 12: Sway speed obtained in a 10/10 degree zig-zag manoeuvre with the ship (solid line) and in simulation (dashed line).

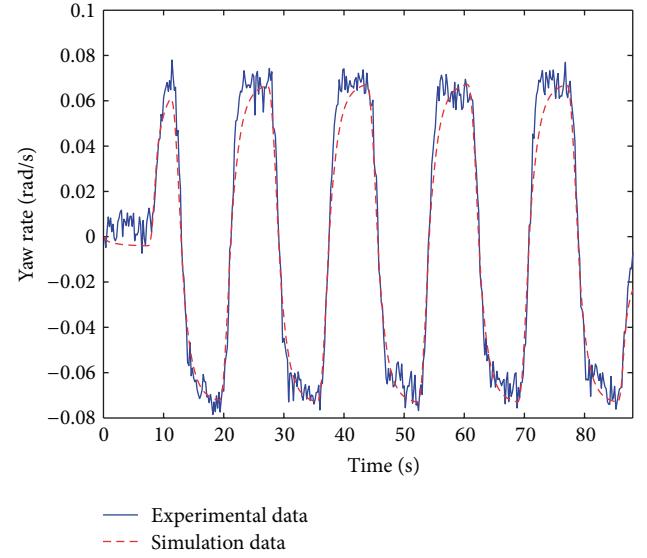


FIGURE 13: Yaw rate obtained in a 10/10 degree zig-zag manoeuvre with the ship (solid line) and in simulation (dashed line).

many practical tests to define a reliable mathematical model and providing a very large prediction ability.

It would be interesting as future research to compare the results obtained in this work with the results that would be obtained using extreme learning machines (ELM) [46], as this technique overcomes some drawbacks that neural networks present, and it also reduces significantly the computation time [47].

5. Conclusions and Future Work

In this work, the nonlinear ship model of Blanke has been computed using experimental data obtained from a zig-zag

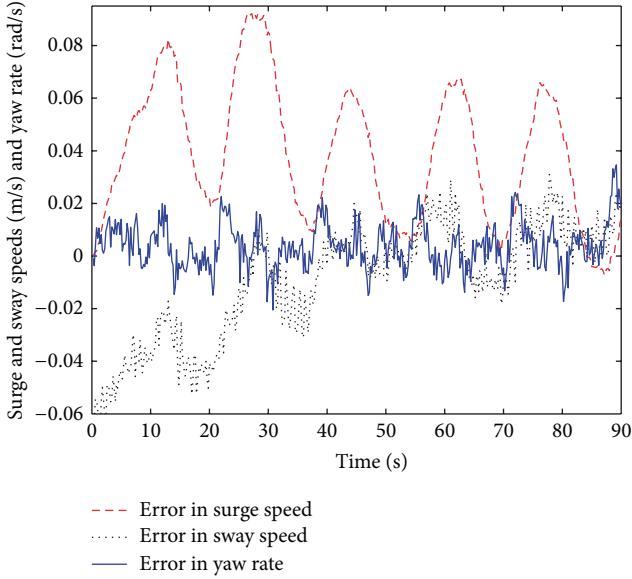


FIGURE 14: Approximation errors in the surge speed (dashed line), sway speed (dotted line), and yaw rate (solid line).

manoeuvre test. A semophysical modelling technique based on a least squares support vector machines algorithm has been applied to determine the parameters of the nonlinear model using the rudder angle, surge and sway speeds, and yaw rate as training data. It was shown that the model obtained fits the training data in a nice manner, showing the simulated system a behaviour very similar to that of the real ship. Furthermore, the prediction ability of the model was validated carrying out several experimental tests, like turning manoeuvres and zig-zags, demonstrating that the mathematical model can reproduce the actual ship dynamics with large accuracy in different manoeuvres. In addition, the model computed is suitable to be used for testing control algorithms in simulation, avoiding the execution of a large number of experimental tests.

Future work will aim at (i) extending the methodology developed to deal with models whose structures are not known in advance to capture all the features of the real ship, incorporating disturbances and environmental conditions, (ii) studying the performance of control algorithms for path following and tracking with the ship model defined in comparison with the results obtained for the real vehicle, and (iii) comparing the results obtained in this work with other different identification techniques, like the extreme learning machines (ELM).

Acknowledgments

The authors wish to thank the Spanish Ministry of Science and Innovation (MICINN) for support under Projects DPI2009-14552-C02-01 and DPI2009-14552-C02-02. The authors wish to thank also the National University Distance Education (UNED) for support under Project 2012V/PUNED/0003.

References

- [1] L. Ljung, *System Identification: Theory for the User*, Prentice-Hall, Upper Saddle River, NJ, USA, 1999.
- [2] L. Ljung, "Identification of Nonlinear Systems," in *Proceedings of the International Conference on Control, Automation, Robotics and Vision*, 2006.
- [3] D. E. Rivera, "Teaching semophysical modeling to ChE students using a brine-water mixing tank experiment," *Chemical Engineering Education*, vol. 39, no. 4, pp. 308–315, 2005.
- [4] P. Lindskog and L. Ljung, "Tools for semophysical modelling," *International Journal of Adaptive Control and Signal Processing*, vol. 9, no. 6, pp. 509–523, 1995.
- [5] J. A. K. Suykens, T. van Gestel, J. de Brabanter, B. de Moor, and J. Vandewalle, *Least Squares Support Vector Machines*, World Scientific, Singapore, 2002.
- [6] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.
- [7] V. Vapnik and Z. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Doklady Akademii Nauk USSR*, vol. 4, no. 181, 1968.
- [8] M. Aizerman, E. Braverman, and L. Rozonoer, "Theoretical foundations of the potential function method in pattern recognition learning," *Automation and Remote Control*, vol. 25, pp. 821–837, 1964.
- [9] B. Schölkopf and A. J. Smola, *Learning With Kernels*, MIT press, Cambridge, Mass, USA, 2002.
- [10] V. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.
- [11] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.
- [12] P. M. L. Drezet and R. F. Harrison, "Support vector machines for system identification," in *Proceedings of the International Conference on Control*, pp. 688–692, September 1998.
- [13] S. Adachi and T. Ogawa, "A new system identification method based on support vector machines," in *Proceedings of the IFAC Workshop Adaptation and Learning in Control and Signal Processing*, L'Aquila, Italy, 2001.
- [14] G. T. Jemwa and C. Aldrich, "Non-linear system identification of an autocatalytic reactor using least squares support vector machines," *Journal of The South African Institute of Mining and Metallurgy*, vol. 103, no. 2, pp. 119–125, 2003.
- [15] W. Zhong, D. Pi, and Y. Sun, "SVM based nonparametric model identification and dynamic model control," in *Proceedings of the First International Conference on Natural Computation (ICNC '05)*, pp. 706–709, August 2005.
- [16] V. Verdult, J. A. K. Suykens, J. Boets, I. Goethals, and B. de Moor, "Least squares support vector machines for kernel cca in non-linear state-space identification," in *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems (MTNS '04)*, Leuven, Belgium, July 2004.
- [17] W. Zhong, H. Ge, and F. Qian, "Model identification and control for nonlinear discrete-time systems with time delay: a support vector machine approach," in *Proceedings of International Conference on Intelligent Systems and Knowledge Engineering (ISKE '07)*, Chengdu, China, October 2007.
- [18] S. Tötterman and H. T. Toivonen, "Support vector method for identification of Wiener models," *Journal of Process Control*, vol. 19, no. 7, pp. 1174–1181, 2009.

- [19] X.-D. Wang and M.-Y. Ye, "Nonlinear dynamic system identification using least squares support vector machine regression," in *Proceedings of International Conference on Machine Learning and Cybernetics*, pp. 941–945, Shanghai, China, August 2004.
- [20] I. Goethals, K. Pelckmans, J. A. K. Suykens, and B. de Moor, "Identification of MIMO Hammerstein models using least squares support vector machines," *Automatica*, vol. 41, no. 7, pp. 1263–1272, 2005.
- [21] Z. Yu and Y. Cai, "Least squares wavelet support vector machines for nonlinear system identification," in *Proceedings of the Second International Symposium on Neural Networks: Advances in Neural Networks (ISNN '05)*, pp. 436–441, June 2005.
- [22] L. Wang, H. Lai, and T. Zhang, "An improved algorithm on least squares support vector machines," *Information Technology Journal*, vol. 7, no. 2, pp. 370–373, 2008.
- [23] J. van Amerongen and A. J. Urdink Ten Cate, "Model reference adaptive autopilots for ships," *Original Research Article Automatica*, vol. 11, no. 5, pp. 441–449, 1975.
- [24] K. J. Åström and C. G. Källström, "Identification of ship steering dynamics," *Automatica*, vol. 12, no. 1, pp. 9–22, 1976.
- [25] C. G. Källström and K. J. Åström, "Experiences of system identification applied to ship steering," *Automatica*, vol. 17, no. 1, pp. 187–198, 1981.
- [26] M. A. Abkowitz, "Measurement of hydrodynamic characteristics from ship maneuvering trials by system identification," *Transactions of Society of Naval Architects and Marine Engineers*, vol. 88, pp. 283–318, 1981.
- [27] T. I. Fossen, S. I. Sagatun, and A. J. Sørensen, "Identification of dynamically positioned ships," *Modeling, Identification and Control*, vol. 17, no. 2, pp. 153–165, 1996.
- [28] T. Perez, A. J. Sørensen, and M. Blanke, "Marine vessel models in changing operational conditions—a tutorial," in *Proceedings of the 14th IFAC Symposium on System Identification*, Newcastle, Australia, 2006.
- [29] M. Caccia, G. Bruzzone, and R. Bono, "A practical approach to modeling and identification of small autonomous surface craft," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 2, pp. 133–145, 2008.
- [30] T. I. Fossen, *Marine Control Systems: Guidance, Navigation, and Control of Ships, Rigs and Underwater Vehicles*, Marine Cybernetics, Trondheim, Norway, 2002.
- [31] J. M. de La Cruz, J. Aranda, and J. M. Giron, "Automática Marina: una revisión desde el punto de vista de control," *Revista Iberoamericana de Automática e Informática Industrial*, vol. 9, pp. 205–218, 2012.
- [32] F. J. Velasco, E. Revestido, L. Eopez, and E. Moyano, "Identification for a heading autopilot of an autonomous in-scale fast ferry," *IEEE Journal of Oceanic Engineering*, vol. 38, no. 2, pp. 263–274, 2013.
- [33] R. Skjetne, Ø. N. Smogeli, and T. I. Fossen, "A nonlinear ship manoeuvering model: identification and adaptive control with experiments for a model ship," *Modeling, Identification and Control*, vol. 25, no. 1, pp. 3–27, 2004.
- [34] M. Blanke, *Ship propulsion losses related to automated steering and prime mover control [Ph.D. thesis]*, The Technical University of Denmark, Lyngby, Denmark, 1981.
- [35] M. A. Abkowitz, "Lectures on ship hydrodynamics steering and manoeuvrability," Tech. Rep. Hy-5, Hydro and Aerodynamics Laboratory, Denmark, 1964.
- [36] M. R. Haddara and Y. Wang, "Parametric identification of manoeuvring models for ships," *International Shipbuilding Progress*, vol. 46, no. 445, pp. 5–27, 1999.
- [37] M. R. Haddara and J. Xu, "On the identification of ship coupled heave-pitch motions using neural networks," *Ocean Engineering*, vol. 26, no. 5, pp. 381–400, 1998.
- [38] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [39] A. B. Mahfouz, "Identification of the nonlinear ship rolling motion equation using the measured response at sea," *Ocean Engineering*, vol. 31, no. 17–18, pp. 2139–2156, 2004.
- [40] W. L. Luo and Z. J. Zou, "Parametric identification of ship maneuvering models by using support vector machines," *Journal of Ship Research*, vol. 53, no. 1, pp. 19–30, 2009.
- [41] X.-G. Zhang and Z.-J. Zou, "Identification of Abkowitz model for ship manoeuvring motion using ϵ -support vector regression," *Journal of Hydrodynamics*, vol. 23, no. 3, pp. 353–360, 2011.
- [42] D. Moreno-Salinas, D. Chaos, J. M. de la Cruz, and J. Aranda, "Identification of a surface marine vessel using LS-SVM," *Journal of Applied Mathematics*, vol. 2013, Article ID 803548, 11 pages, 2013.
- [43] F. Xu, Z.-J. Zou, J.-C. Yin, and J. Cao, "Identification modeling of underwater vehicles' nonlinear dynamics based on support vector machines," *Ocean Engineering*, vol. 67, Article ID 002980, pp. 68–76, 2013.
- [44] J. Mercer, "Functions of positive and negative type and their connection with the theory of integral equations," *Philosophical Transactions of the Royal Society A*, vol. 209, pp. 415–446, 1909.
- [45] K. Nomoto, T. Taguchi, K. Honda, and S. Hirano, "On the steering qualities of ships," Tech. Rep., International Shipbuilding Progress, 1957.
- [46] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 985–990, July 2004.
- [47] R. Rajesh and J. Siva Prakash, "Extreme learning machines—a review and state-of-the-art," *International Journal of Wisdom Based Computing*, vol. 1, no. 1, 2011.

Research Article

An Ensemble Method for High-Dimensional Multilabel Data

Huawen Liu,^{1,2} Zhonglong Zheng,¹ Jianmin Zhao,¹ and Ronghua Ye¹

¹ College of Mathematics, Physics and Information Engineering, Zhejiang Normal University, Jinhua 321004, China

² NCMIS, Academy of Mathematics and Systems Science, CAS, Beijing 100190, China

Correspondence should be addressed to Huawen Liu; hwliu@zjnu.edu.cn

Received 1 August 2013; Accepted 5 October 2013

Academic Editor: Zhiqiang Ma

Copyright © 2013 Huawen Liu 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.

Multilabel learning is now receiving an increasing attention from a variety of domains and many learning algorithms have been witnessed. Similarly, the multilabel learning may also suffer from the problems of high dimensionality, and little attention has been paid to this issue. In this paper, we propose a new ensemble learning algorithms for multilabel data. The main characteristic of our method is that it exploits the features with local discriminative capabilities for each label to serve the purpose of classification. Specifically, for each label, the discriminative capabilities of features on positive and negative data are estimated, and then the top features with the highest capabilities are obtained. Finally, a binary classifier for each label is constructed on the top features. Experimental results on the benchmark data sets show that the proposed method outperforms four popular and previously published multilabel learning algorithms.

1. Introduction

Data classification is one of the major issues in data mining and machine learning. Generally speaking, it consists of two stages, that is, building classification models and predicting labels for unknown data. Depending on the number of labels tagged on each data, the classification problems can be divided into single-label and multilabel classification [1]. In the former, the class labels are mutually exclusive and each instance is tagged with only one class label. On the contrary, each instance may be tagged with more than one class label simultaneously. The multilabel classification problems are ubiquitous in real-world applications, such as text categorization, image annotation, bioinformatics, and information retrieval [1, 2]. For example, the movie “*avatar*” may be tagged with *action*, *science fiction*, and *love* types.

Now, many multilabel classification algorithms have been witnessed. Roughly speaking, they can be grouped into two categories, that is, algorithm adaption and problem transformation [1]. The first kind of technique extends traditional single-label classifiers, such as kNN, C4.5, SVM, and AdaBoost, by modifying some constraint conditions to handle multilabel data. Typical examples include AdaBoost.MH [3], BRkNN [4], and LPkNN [4]. For instance, Zhang and

Zhou [5] proposed MLkNN and applied tscene classification, while Clare and King [2] employed C4.5 to deal with multilabel data by altering the discriminative formula of information entropy.

The second technique of multilabel learning transforms multilabel data into corresponding single-label ones and then handle them one by one using the traditional methods. An intuitive approach is to treat the multilabel problem as a set of independent binary classification problems, one for each class label [6, 7]. However, they often have not considered the correlations among the class labels and may suffer from the problem of unbalanced data, especially when there are a large number of the class labels [8]. To cope with these problems, several strategies have been introduced. For example, Zhu et al. [9] explored the label correlation with maximum entropy, while Cai and Hofmann [10] captured the correlation information among the labels by virtue of a hierarchical structure.

Analogous to traditional classification, multilabel learning may also encounter the problems, such as over-fitting and the curse of dimensionality, raised from high dimensionality of data [11, 12]. To alleviate this problem, an effective solution is to perform dimension reduction or feature selection on data in advance. As a typical example, Ji et al. [13] extracted

a common subspace shared among multiple labels by using ridge regression. One common characteristic of these methods is that they make use of only one feature set to achieve the learning purpose under the context of multilabel data. However, in reality, only one feature subset can not represent the properties of different labels exactly. Therefore, it is necessary to choose different features for each label during the multilabel learning stage. A representative example of such kind is LIFT [14].

In this paper, we propose a new multilabel learning algorithm. The main characteristic of our method is that during the procedure of constructing binary classifiers different feature subsets will be exploited for each label. More specifically, given a class label, the features with high discriminative capabilities with respect to the label are chosen and then used to train a binary classifier. This means that the selected features have local properties. Note that they may have lower discriminative capabilities with respect to other class labels. Other binary classifiers can also be constructed in a similar manner. Finally, all binary classifiers are assembled into an overall one, which will be used to predict or classify the labels of unknown data.

The rest of this paper is organized as follows. We describe the details of the proposed method in Section 2. Experimental results conducted to evaluate the effectiveness of our method are presented in Section 3. Finally, conclusions and future works are given in the end.

2. Binary Classification with Feature Selection

Assume that $L = \{l_1, \dots, l_m\}$ denotes the finite set of labels in a multilabel learning task. Let $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_n, \mathbf{y}_n)\}$ be a multilabel data set. It consists of n independently identically distributed (iid) samples. $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{id})^T \in X$ and $\mathbf{y}_i = (y_{i1}, y_{i2}, \dots, y_{im})^T \in Y$ are the d -dimensional feature and m -dimensional label vectors of the i th sample, respectively. y_{ij} takes a value of 1 or 0, indicating whether i th sample is associated with the j th label or not.

2.1. Binary Classification. According to the formal description of \mathcal{D} , we know that the multilabel data is a general form of traditional single-label data, whereas \mathbf{y}_i only involves one single label. Thus, a natural and intuitive solution for multilabel learning is to transform the multilabel data into its corresponding single-label data and then train classifiers on the generated data. There are many transformation strategies. *Copy*, *selection*, and *ignore* are three typical transformation techniques [1]. Besides, the power set of labels is also introduced in the literature, where every \mathbf{y}_i is often taken as a new class label.

Before giving the principle of binary classification, let us introduce the concepts of positive and negative samples of labels.

Definition 1. Given a multilabel data set \mathcal{D} with n samples associated with m labels L , for each class label $l_k \in L$, its

positive samples $P(l_k)$ and negative samples $N(l_k)$ are defined as follows:

$$P(l_k) = \{\mathbf{x}_i \mid (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}, y_{ik} = 1\}, \quad (1)$$

$$N(l_k) = \{\mathbf{x}_i \mid (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}, y_{ik} = 0\}. \quad (2)$$

From this definition, we know that, given a label l_k , all examples of the original data set are positive if they are associated with the class label l_k and negatively otherwise. Moreover, $P(l_k) \cap N(l_k) = \emptyset$ and $P(l_k) \cup N(l_k) = X$.

Binary relevance (BR), also known as one-against-all method, is the most popular and most commonly used transformation method for multilabel learning in the literature [1]. It learns m different binary classifiers independently, one for each different label in L . Specifically, it transforms the original data set \mathcal{D} into m data sets \mathcal{D}_i , $i = 1, \dots, m$. Each data set \mathcal{D}_i consists of the positive samples $P(l_i)$ and negative samples $N(l_i)$ with respect to l_i . Based on the new data set \mathcal{D}_i , a binary classifier f_i for the label l_i can be built using the off-the-shelf learning methods, for example, kNN and SVM. After obtaining m binary classifiers for all labels, the prediction of BR for a new sample x is the union of the labels l_k that are positively predicted by the m classifiers; that is, $f(x) = [f_1(x), f_2(x), \dots, f_m(x)]^T$, where $f_i(x)$ takes a value of 0 or 1, indicating x is predicted positively or negatively by the classifier f_i .

BR is a straight forward transformation method and widely used as a baseline in comparison with multilabel learning algorithms. However, the drawback of BR is that it does not take correlations among the labels into account and treats all labels independently. In addition, it also suffers from the class imbalance problem. In multilabel data, the number of positive samples $P(l_k)$ is significantly less than the number of negative samples $N(l_k)$ for some labels due to the typical sparsity of labels. To alleviate this problem, feature selection should be performed on the data sets in advance.

2.2. Feature Selection. The purpose of feature selection is to select significant features to represent data from the original space without losing information greatly. It has been extensively studied in the traditional learning. However, little work of feature selection has been done in the context of multilabel learning. Currently, there are many criteria available to measure the interestingness of features [15]. Here, we exploit the concept of density distribution of data to represent the interestingness of features.

Definition 2. Given a data set \mathcal{D}_k with n samples, the density of value distribution of the i th feature is defined as

$$\rho_i^k(\mathcal{D}_k) = \frac{\sum_{u=1}^n \sum_{v=1}^n \text{sim}(x_{ui}, x_{vi})}{n \cdot \min_{1 \leq u, v \leq n} \text{sim}(x_{ui}, x_{vi})}, \quad (3)$$

where x_{ui} denotes the i th feature value of x_u and sim is a similar function between two values.

In (3), the sim function is often taken as the form of inverse Euclidean distance. If the positive samples $P(l_k)$ and negative samples $N(l_k)$ are considered in (3), we can get the positive and negative densities of features.

Definition 3. Given the positive samples $P(l_k)$ and negative samples $N(l_k)$, the positive and negative densities of the i th feature are defined as

$$\rho_{i+}^k = \rho_i^k(P(l_k)), \quad (4)$$

$$\rho_{i-}^k = \rho_i^k(N(l_k)). \quad (5)$$

The positive density ρ_{i+}^k , as well as the negative density ρ_{i-}^k , can effectively represent the specific characteristic of data. The larger the value of ρ_{i+}^k (or ρ_{i-}^k), the better discriminative capability to distinguish positive (or negative) samples from others.

Based on this principle, we adopt these two criteria to choose significant features during the learning stage. Specifically, for each feature a_i in \mathcal{D}_k , we calculate its positive density ρ_{i+}^k and negative density ρ_{i-}^k , respectively. Then, the positive densities of all features will be ranked in a decreasing order, and the top t features with high positive densities will be selected. Similar situation can be done for the negative densities. Finally, the features with high positive and negative densities will be used to train desirable binary classifiers.

How many features should be selected for classification is still an open problem. Here, we empirically determine the number of selected features with a concept called m_k -minimum density, which is defined in the following.

Definition 4. Let \mathcal{D}_k be the data set, and let $P(l_k)$ and $N(l_k)$ be the positive and negative samples of the i th feature, respectively. The m_k -minimum density of \mathcal{D}_k with respect to l_k is

$$m_k = r \cdot \min(|P(l_k)|, |N(l_k)|), \quad (6)$$

where $r \in [0, 1]$ and $|\cdot|$ is the set cardinality.

The m_k -minimum density can effectively measure the information amount that one feature has. If the density is larger than the m_k -minimum density, the corresponding feature has enough information to represent the characteristics of data. As a result, the feature will be chosen during the stage of feature selection. In other words, after calculating ρ_{i+}^k and ρ_{i-}^k , we retain the features with ρ_{i+}^k or ρ_{i-}^k larger than m_k and discard the others. Note that the parameter r in Definition 4 is to control the number of selected features. The larger value of r is, the more features would be chosen. In our empirical experiments, the classifier achieved good performance when r was set to 0.1.

2.3. The Proposed Method. Based on the analysis above, we propose a new multilabel learning algorithm. The framework of our algorithm is shown as Algorithm 1. The proposed method works in a straightforward way and can be easily understood. It consists of two major stages, that is, learning and prediction stages. In the training stage, a new data set will be generated for each class label by obtaining its positive and negative samples. Subsequently, we estimate the interestingness of features in the data set, so as to retain significant features for classification. Finally,

a binary classifier is constructed with a baseline learning method. Given a new sample, its class labels can be predicted by testing it with all binary classifiers.

3. Empirical Study

To validate the performance of our proposed method, we made a comparison of EMCFS with three popular multilabel learning algorithms. They are ML- k NN, LIFT, and Rank-SVM, standing for different kinds of learning types. In addition, we took a linear support vector machine as the baseline binary classification algorithm and assigned 0.1 to the parameter r . Other parameters were set as their default values suggested by authors. For example, the number of the nearest neighbors in ML- k NN was 10 and the distance measure is the Euclidean distance [5]. In the Rank-SVM classifier, the degree of polynomial kernels was 8 and the cost parameter c was assigned as one [16].

3.1. Data Sets. To validate the effectiveness of our method roundly, our experiments have been conducted on four data sets, including *emotions*, *medical*, *corel16k* (sample 1), and *delicious*. They are often used to verify the performance of multilabel classifiers in the literature and are available at <http://mulan.sourceforge.net/datasets.html>. Table 1 summarizes their general information, where the *cardinality* and *density* columns refer to the average number of class labels of the samples and its fraction by the number of labels. These multilabel data sets vary from the quantities of labels and differ greatly in the sizes of samples and features [17].

3.2. Experimental Results. There are lots of evaluation criteria available to evaluate the performance of multilabel classifiers. In this work, average precision, hamming loss, one error, and coverage have been adopted to assess the effectiveness of the proposed method. Their detailed descriptions can be found in several literatures such as those in [1, 18].

Table 2 reports the average precision of the multilabel classifiers on the data sets. In this table, each row denotes an observation on the data sets. The best result comparable with others in the same row is highlighted in boldface, where the larger the value, the better the performance. From the table, one may observe that the proposed method, EMCFS, works quite well and is comparable to others in most cases with the average precision. For example, on the *delicious* data set (the last row in the table), the precision of EMCFS is 29.5%, which is the best one among the others.

Apart from the average precision, we also compared EMCFS to the others from the perspective of the hamming loss, one error, and coverage. Tables 3, 4, and 5 present the averaged performance of the learning algorithms in terms of these three criteria, respectively, where the smaller the value, the better the performance. The best results are also highlighted in boldface.

According to algorithms the results in these tables, we know that similar to the average precision, EMCFS is also superior to other regarding the aspects of hamming loss, one error and coverage. Although EMCFS achieved slightly poor

```

Input:  $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$ : The training multilabel data set;
       $r$ : The parameter of  $m_k$ ;
       $U = x_1, \dots, x_s$ : The test data set without labels;
Output:  $Y = \{y_1, \dots, y_s\}$ : The label set of  $U$ ;
Training stage
For each label  $l_k$  in  $L$  do
    Obtain  $P(l_k)$  and  $N(l_k)$  of  $l_k$  from  $\mathcal{D}$  according to Def.(1)(2);
    Calculate  $m_k$  with  $P(l_k)$  and  $N(l_k)$  according to Def. (6);
    For each feature  $a_i$  in  $\mathcal{D}_k = P(l_k) \cup N(l_k)$  do
        Calculate  $\rho_{i+}^k$  and  $\rho_{i-}^k$  according to Def.(4)(5);
        Select the features whose  $\rho_{i+}^k$  or  $\rho_{i-}^k$  is larger than  $m_k$ ;
        Train a binary classifier  $f_k$  on  $\mathcal{D}_k$  with the selected features;
Endfor
Prediction stage
For each sample  $o_i$  in  $U$  do
     $y_i = \emptyset$ ;
    For each classifier  $f_k$  do
         $y_i = y_i \cup \{f_k(o_i)\}$ , where  $f_k$  returns 0 or 1;
Endfor

```

ALGORITHM 1: Ensemble multilabel classifier using feature selection (EMCFS).

TABLE 1: The brief description information of data sets in experiments.

Data set	Domain	Sample	Feature	Label	Cardinality	Density
Emotions	Music	593	72	6	1.87	0.31
Medical	Text	978	1449	45	1.25	0.03
Corel16k	Image	13766	500	153	2.86	0.02
Delicious	Web	16105	500	983	19.02	0.02

TABLE 2: A comparison of average precision of four classifiers on data sets (%).

Data set	EMCFS	ML-kNN	LIFT	Rank-SVM
Emotions	80.5	76.3	78.9	77.6
Medical	89.9	78.2	86.9	87.2
Corel16k	32.3	27.6	31.5	29.7
Delicious	29.5	25.7	28.2	22.4

TABLE 3: A comparison of hamming loss of four classifiers on data sets.

Data set	EMCFS	ML-kNN	LIFT	Rank-SVM
Emotions	0.189	0.214	0.206	0.218
Medical	0.009	0.016	0.012	0.014
Corel16k	0.017	0.928	0.017	0.019
Delicious	0.019	0.998	0.021	0.025

coverage on the *corel16k* data set, the performance is 140.428, which is slightly worse than the best one. However, it is not the worst in comparison with ML-kNN.

4. Conclusions

In this paper, we propose a new ensemble multilabel learning method. The central idea of our method is that, for each label,

TABLE 4: A comparison of one error of four classifiers on data sets.

Data set	EMCFS	ML-kNN	LIFT	Rank-SVM
Emotions	0.253	0.285	0.264	0.293
Medical	0.140	0.267	0.180	0.194
Corel16k	0.671	0.852	0.674	0.783
Delicious	0.401	0.482	0.433	0.665

TABLE 5: A comparison of coverage of four classifiers on data sets.

Data set	EMCFS	ML-kNN	LIFT	Rank-SVM
Emotions	2.148	2.561	2.302	2.335
Medical	1.256	2.504	1.403	1.852
Corel16k	140.428	151.853	139.592	136.527
Delicious	662.746	674.613	667.513	671.652

it exploits different features to build learning models. The advantage is that the classifiers are constructed on the features with strong local discriminative capabilities. Generally, the proposed method consists of three steps. Firstly, for each label, a new data set is generated by identifying the positive and negative samples. Then, the interestingnes's of features will be estimated and the features with high density will be retained to train a learning model. Finally, all binary classifiers built with the selected features will be integrated into an overall one. Experimental results on four multilabel

data sets show that the proposed method can potentially improve performance and outperform other competing and popular methods.

Acknowledgments

The authors are grateful to the anonymous referees for their valuable comments and suggestions. This work is partially supported by the National NSF of China (61100119, 61170108, 61170109, 61272130, and 61272468), the NSF of Zhejiang province (Y1110483), Postdoctoral Science Foundation of China (2013M530072), and the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) (201204214).

References

- [1] G. Tsoumakas, I. Katakis, and I. Vlahavas, “Mining multi-label data,” in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds., pp. 667–686, 2010.
- [2] A. Clare and R. King, “Knowledge discovery in multi-label phenotype data,” in *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 42–53, 2001.
- [3] R. E. Schapire and Y. Singer, “BoosTexter: a boosting-based system for text categorization,” *Machine Learning*, vol. 39, no. 2, pp. 135–168, 2000.
- [4] G. Tsoumakas, I. Katakis, and I. Vlahavas, “Random k-labelsets for multilabel classification,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 1079–1089, 2011.
- [5] M.-L. Zhang and Z.-H. Zhou, “ML-KNN: a lazy learning approach to multi-label learning,” *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.
- [6] L. Rokach, “Ensemble-based classifiers,” *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1–39, 2010.
- [7] M. L. Zhang and Z. H. Zhou, “A review on multi-label learning algorithms,” *IEEE Transactions on Knowledge and Data Engineering*, no. 99.
- [8] G. M. Weiss and F. Provost, “Learning when training data are costly: the effect of class distribution on tree induction,” *Journal of Artificial Intelligence Research*, vol. 19, pp. 315–354, 2003.
- [9] S. Zhu, X. Ji, W. Xu, and Y. Gong, “Multi-labelled classification using maximum entropy method,” in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 274–281, 2005.
- [10] L. Cai and T. Hofmann, “Hierarchical document categorization with Support Vector Machines,” in *Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management (CIKM ’04)*, pp. 78–87, November 2004.
- [11] H. Liu, X. Wu, and S. Zhang, “A new supervised feature selection method for pattern classification,” *Computational Intelligence*, 2013.
- [12] B. Bonev, F. Escolano, D. Giorgi, and S. Biasotti, “Information-theoretic selection of high-dimensional spectral features for structural recognition,” *Computer Vision and Image Understanding*, vol. 117, no. 3, pp. 214–228, 2013.
- [13] S. Ji, L. Tang, S. Yu, and J. Ye, “A shared-subspace learning framework for multi-label classification,” *ACM Transactions on Knowledge Discovery from Data*, vol. 4, no. 2, article 8, 2010.
- [14] M. L. Zhang, “LIFT: multi-label learning with label-specific features,” in *Proceedings of the 22nd international joint conference on Artificial Intelligence (IJCAI ’11)*, pp. 1609–1614, 2011.
- [15] X. Li, J. Wang, J. Zhou, and M. Yin, “A perturb biogeography based optimization with mutation for global numerical optimization,” *Applied Mathematics and Computation*, vol. 218, no. 2, pp. 598–609, 2011.
- [16] A. Elisseeff and J. Weston, “A kernel method for multi-labelled classification,” in *Proceedings of the Neural Information Processing Systems*, pp. 681–687, 2002.
- [17] X. Li and M. Yin, “An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure,” *Advances in Engineering Software*, vol. 55, pp. 10–31, 2013.
- [18] X. Li and M. Yin, “Application of differential evolution algorithm on self-potential data,” *PloS ONE*, vol. 7, no. 12, Article ID e51199, 2012.

Research Article

BMCloud: Minimizing Repair Bandwidth and Maintenance Cost in Cloud Storage

Chao Yin,¹ Changsheng Xie,^{1,2} Jiguang Wan,^{1,2} Chih-Cheng Hung,^{3,4} Jinjiang Liu,⁵ and Yihua Lan⁵

¹ School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

² Wuhan National Laboratory for Optoelectronics, Wuhan 430074, China

³ School of Computer and Information Engineering, Anyang Normal University, Anyang 455000, China

⁴ Center for Biometrics Research, Southern Polytechnic State University, Atlanta, GA, USA

⁵ School of Computer and Information Technology, Nanyang Normal University, Nanyang 473061, China

Correspondence should be addressed to Changsheng Xie; cs_xie@mail.hust.edu.cn

Received 11 July 2013; Accepted 8 September 2013

Academic Editor: William Guo

Copyright © 2013 Chao Yin 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.

To protect data in cloud storage, fault tolerance and efficient recovery become very important. Recent studies have developed numerous solutions based on erasure code techniques to solve this problem using functional repairs. However, there are two limitations to address. The first one is consistency since the Encoding Matrix (EM) is different among clouds. The other one is repairing bandwidth, which is a concern for most of us. We addressed these two problems from both theoretical and practical perspectives. We developed BMCloud, a new low repair bandwidth, low maintenance cost cloud storage system, which aims to reduce repair bandwidth and maintenance cost. The system employs both functional repair and exact repair while it inherits advantages from the both. We propose the JUDGE_STYLE algorithm, which can judge whether the system should adopt exact repair or functional repair. We implemented a networked storage system prototype and demonstrated our findings. Compared with existing solutions, BMCloud can be used in engineering to save repair bandwidth and degrade maintenance significantly.

1. Introduction

With the rapid growth of data production in companies, the requirement of storage space grows very largely as well. This growth leads to the emergence of cloud storage. Cloud storage is a concept which is an extension and development from cloud computing. This system collects application software in order to work together and provide systems of data storage and business access features through grids or distributed file systems [1]. Cloud storage is produced by distributed storage technology and virtualization technology, and it is the latest development of distributed storage technology. Cloud storage provides effective solutions for network mass data storage. Also this system provides on-demand pay services, which reduces not only the threshold for the user but also the payment. Some companies have been involved in cloud storage systems, such as Amazon, Sun, Google, and Yahoo.

According to forecasts from the International Data Corporation (IDC), the size of the global cloud computing and cloud storage market has increased from \$16 billion in 2008 to \$42 billion in 2012 [2], which is the proportion of the total IT investment from 4.2% to 5%. In addition, the IDC predicts that cloud computing and cloud storage investments will be 25% of the annual IT investment in 2012, and that will rise to 30% or more in 2013. The advisory body Merrill (Merrill Lynch) foresees that this will be \$160 billion market by 2011.

In order to improve the reliability of cloud storage, some companies have developed their own solutions, such as HDFS [3] and GFS [4]. The two cloud storage systems use duplicate technology to have reliability guaranteed. Of course, this method can solve the problem of data loss and data error. But there is a lot of storage space wasted by using this method and this will bring consistency issues. For these reasons, the

erasure code has become one of the most popular choices for storage to improve its utilization and avoid the problem of consistency. A good erasure code technology can not only improve the availability and reliability of the system but can also improve the efficiency of data access.

We hereby focus on the recovery problem for a family of network coding. In the current situation, cloud coding is a popular direction to prevent large-scale cloud node failure. While bandwidth consumption is an important performance signal in cloud storage system, we always want to repair data using the minimum bandwidth and the fastest repair speed. The model for a cloud file system using erasure codes is inspired by NCCloud [5], which is the implementable design for the functional minimum-storage regenerating code [6]. Other cloud file systems such as Khan et al. [7] are also function repair systems. In functional repair, the system needs to keep the value of the repair matrix to ensure consistency. Following each functional repair, that matrix needs to be synchronized to each cloud in order to ensure the consistency across the system. The number of recovery matrices increases with the number of recoveries, which will affect the overall system reliability and consistency. Therefore we should establish a system with an exact repair in case the system experiences data loss or data error.

In this paper, we have proposed Bandwidth and Maintenance minimum Cloud system (BMCloud) which has low bandwidth consumption and low overhead in terms of maintenance. The system has both functional repair and exact repair when data loss or data error are experienced. When a part of data in cloud is lost, the system can recover with exact repair to degrade the overhead of maintenance for the future. More importantly, it has the ability to consume less bandwidth when recovering. While almost all the data in the cloud breaks, the system can recover it with a functional repair. We have developed a JUDGE_STYLE rule to judge whether the system should use exact repair to recover or not.

The system includes a proxy which can calculate and transfer data through the clouds and can maintain the system consistency. In the exact repair function, the proxy itself does not require arithmetic processing of the data or the data cache, and it does not need to provide calculation and storage capabilities. Since most of the calculation work is loaded on the cloud nodes in this function, which will be described in Section 4 with more details, the proxy would not be a bottleneck in terms of calculation and storage.

The contributions of this paper are described as follows.

- (i) We have developed E Code algorithm to recover data in exact repair. It can improve recovery bandwidth performance and ensure data integrity.
- (ii) We have developed the JUDGE_STYLE algorithm, which can judge whether the system should use exact repair or functional repair.
- (iii) We have implemented the system BMCloud. When the number of the fail strips in the cloud is less than or equal to 4, or between 4 and 10 but we can recover data in exact repair, BMCloud could significantly improve repair bandwidth by 53.9% and 41.8% compared to

current solutions (RDP, NCCloud, etc.). In addition, it can degrade maintenance cost.

The rest of this paper is organized as follows. Section 2 describes related work in erasure code. The architecture and the design of BMCloud are introduced in Section 3. Section 4 is the prototype and algorithms. Section 5 is the experimental result and evaluation. Section 6 is the conclusions.

2. Related Works and Motivation

2.1. RAID Code. Since Patterson et al. [8] brought the concept of RAID into storage systems, erasure coding has been one of the most popular choices to supply high reliability and high performance storage services with acceptable spatial and monetary cost. Examples such as LAN, RAID0, RAID1, and RAID5 are very small and their integration versions are widely used. These can be used to recover systems when a disk error occurs. With the reliability requirements improved, single-fault-tolerant could not meet the needs of the consumers, so two-fault-tolerant comes into play. There are a lot of two-fault-tolerant codes storage systems, such as EVENODD [9], RDP [10], and X-code [11]. A typical RAID-6 storage system is composed of $k + 2$ disk drives. The first k disk drives are used to store original data, and the last two are used as parity disk drives. All the disk drives are in two check chains. When one or two disk drives lose data, we can recover data from the two check chain. Wan et al. [12] have used the raid code to improve the system performance and save the energy.

Maximum Distance Separable (MDS) can tolerate maximum failures with a given amount of redundancy. It is widely used in RAID-6 code, so that most of the RAID-6 codes are MDS codes. The most important feature of MDS is the ability to verify that the length of the chain is equal to the length of all the disks, which means that all disks are involved in checking, so you can reach the maximum disk utilization. But there is another coding which is different to MDS. In these cases, the length of the chain is less than the length of all the disks, such as in M-code [13]. Therefore due to careful design of the parity chain arrangement, the length of its check chain is shorter than that of the conventional RAID-6 code, which makes its I/O consumption smaller.

2.2. Network Coding. With the development of the network, especially the development of the cloud in recent years, erasure code is also transferred from the disk array level to the cloud level. In network coding, there are three versions of repair which include exact repair, functional repair, and exact repair of systematic parts.

Functional repair has been researched to the greatest extent of the three [14–16]. Dimakis et al. [17] have proposed regenerating codes for functional repairs. They have created a serial theory of the threshold function with informational flow graph. This function describes the relationship between node numbers, surviving nodes numbers, communicating bits, and repair bandwidth. They also studied two extremal

points on the optimal trade-off curve between Minimum-Bandwidth Regenerating (MBR) Codes [18] and minimum-storage regenerating (MSR) codes [19, 20]. In MSR codes, the system stores the average bits at each node while ensuring the MDS-code property, so that they are equivalent to standard MDS codes. While in MBR codes, because we are concerned with the minimum bandwidth, MDS codes and non-MDS codes can be used.

After Rashmi et al. [21] had deeply researched MBR and MSR codes, they proposed the model of twin-code framework, which is the trade-off between MBR codes and MSR codes. Under this framework, the nodes are partitioned into two types and encoded using two codes in a manner that reduces the problem of node repair to that of erasure-decoding of the constituent codes. Depending upon the choice of the two codes, the framework can be used to avail one or more of the following advantages: simultaneous minimization of storage space and bandwidth repair, low complexity of operation, fewer disk reads at helper nodes during repair, and error detection and correction.

Exact repair is also proven theoretically. With the idea of interference alignment [22, 23], the exact repair also includes Exact-MBR Codes and Exact-MSR Codes as does functional repair. The idea is to align multiple interference signals in a signal subspace whose dimensions are smaller than the number of interferers.

Different from the theory analysis above, NCCloud has proposed the implementable design for the functional repair. The system has divided each cloud into two parts and uses its algorithm, so that time for data reads can be reduced by 25% compared to time taken to reconstruct the whole file. There are cloud storage systems that provide a scalable platform for storing massive data over multiple storage nodes via erasure coding [24, 25].

3. BMCloud

3.1. Design Goals. In this paper, we want to develop a cloud system applied as a deep archive and make some progress on the base of an existing system. We specifically design BMCloud under a thin cloud assumption—that the remote data center storing the backups does not provide any special backup services, as Cumulus [26] has proposed. The following factors are taken into account.

3.1.1. Low Bandwidth Consumption. Due to the importance of bandwidth consumption, we put it first on this list. Network code has made some advancements on this issue, but we believe that there is still plenty of room for improvement. That is why the bandwidth is one of the most important factors to consider. We wanted to develop an extra layer on functional repair to create a hybrid system, and so finally we chose E Code, which is a RAID-6 code and has excellent I/O properties. In BMCloud, we elaborately apply E Code on the cloud platform and reserve its excellent properties in I/O (see details in Section 3). E Code can also help the system in locating errors and repairing them efficiently. This creation fills in the blanks in the cloud storage system. In BMCloud,

cloud nodes are required to have computing abilities, limited to XOR operations, which could easily compute concurrently. However, the response time will not increase due to the reduction in the size of the file's need to transport.

3.1.2. Low Overhead of Maintenance. In BMCloud, we want to improve the functional repair model in some aspects, such as bandwidth and computing cost.

Existing systems such as NCCloud, which use functional repair models, need to regenerate EM to repair failed nodes and that may cause some problems that we should not ignore. Since each fault will lead to whole node data regeneration and EM updates, the maintenance overhead of EM may rise significantly, because it will take many computing resources to generate new EM. After a certain number of faults, the maintenance overhead of the whole system may become unacceptable.

In BMCloud, E Code can deal with faults with size of one or two F-MSR blocks and repair them exactly. Every fault repaired by E Code will not need to regenerate and update EM. This property guarantees that the EM will stay stable for a relatively long time. Furthermore, repairing computing cost of E Code is much less than F-MSR. So the system will not incur cost as high as that for computing and bandwidth resources to maintain stability. Since faults in small scale constitute to the majority of regular faults, E Code can be very helpful.

From that, we know that even though we added an extra layer in our system, there is still room for improvement in terms of better computing and bandwidth performance.

3.1.3. Preservation of Regenerating Code Properties. Though faults on a large scale seldom occur, cloud systems must have a mechanism to deal with these lethal problems. F-MSR code is an excellent approach to a solution and has an acceptable bandwidth cost. We preserve the fault tolerance requirement and repair traffic with F-MSR (with up to a small constant overhead) as compared to the conventional repair method in erasure codes.

In BMCloud, we improve the performance of bandwidth, stability, and other aspects on the premise of the preservation of the properties of F-MSR code.

3.1.4. Flexibility. In BMCloud, we elaborately designed the repair model which made it capable to provide the most economical and stable solutions for faults on different scales. Furthermore, BMCloud has excellent expandability. The number of nodes in E Code can be any number larger than 4, and this property helps the E Code layer connect to F-MSR seamlessly. If a better system using the functional repair model came up, it would be very convenient to deploy E Code on it, so the mechanism of BMCloud can be applied in many situations.

3.2. Notations. For an (n, k) -F-MSR code, we define the $n^*(n-k)$ code chunks encoded from $k^*(n-k)$ native chunks as F-MSR chunks. Normally, an F-MSR chunk would be placed on a different cloud node. The encode coefficient matrix in

TABLE 1: Notations of BMCloud.

Notations	Description
F-MSR chunk	A data unit in F-MSR code
Stripe	A coding group containing a collection of strips in E Code
Strip	A data unit of a stripe in E Code
E Code area	Two neighboring cloud nodes in one stripe
Encode matrix (EM)	Coefficient matrix in the F-MSR code
Extended F-MSR chunk	An code chunk after E Code encoding and extension

the F-MSR is defined as the encoding matrix (EM), which is the main part in F-MSR metadata. Each F-MSR chunk will be divided into $(n - 1)^* n$ strips.

In E Code, we define any two cloud nodes adjacent to each other as an E Code area. An exception occurs for the first cloud node, which is grouped with the last node in an E Code area. To a specific file, an E Code area contains two F-MSR chunks.

A stripe is a concept borrowed from traditional RAID codes, and they are an independent coding group of strips. Any stripe in E Code belongs to specific E Code area.

After E Code coding, an additional strip would be appended to a stripe and more strips in each F-MSR chunk. We define the code chunk after extension as an extended F-MSR chunk. Table 1 shows the notations of BMCloud.

3.3. The Design of BMCloud

3.3.1. Software Architecture of BMCloud. In order to solve the concerns mentioned in Section 3.1, we developed BMCloud. BMCloud is a dual-layer system based on F-MSR shown as Figure 1. The first layer contains the classic F-MSR code, providing the tolerance ability on a cloud level. The second layer implements E Code. Data on every cloud node is firstly encoded by F-MSR code and then encoded by E Code before unloading to the cloud node.

BMCloud consists of three modules as coding, storage, and protocol, and defines four workflows as download, update (upload), delete, and repair. Basic functions in file systems and data structure definitions, as well as consistency control and other utility functions, are also included in the system.

BMCloud is mounted on Linux with FUSE (filesystem in userspace). Basic functions in a filesystem are supported by user-defined codes. Our system implements reading and writing, rename, link, creating a new folder, changing file attributes, and other basic functions only because a perfectly functioned filesystem is not BMCloud's point. BMCloud applies Hadoop zookeeper distributed applications to provide coordination services and ensure the system's strict consistency. In the part of workflows, the system mainly defines the encoding, decoding, update (upload), download, repair, and delete operation workflows. This part is the implementation section about encoding and decoding operation and repair strategy in FMSR and E Code. The underlying part has three modules. Coding module provides a variety of basic encoding methods, including FMSR, RAID0, RAID1, and RS

coding, in which we focused on the use of FMSR, a functional repair coding method. Storage module is designed to adapt to different network environments and cloud vendors to provide an interface for basic I/O. Protocol module is a specialized extension module for E Code, defining generation and transmission of the parity blocks in E Code.

3.3.2. E Code Algorithm. Figure 2 is an example of an E Code encoded process. The deep gray squares are the parity strips of E Code. The other 12 strips are data strips which are partitions of F-MSR chunks. In Figure 2, we can see that every data strip belongs to two data links. If n is the number of the nodes, when $n > 3$, E Code can provide double-fault tolerance to protect data in nodes.

Define the row sequence and the column sequence in a strip in the left cloud node as x and y , respectively. So all the left strips in a stripe are equal to $(x + y) \bmod (n + 1)$, which is defined as L . When r represents the row sequence in the right cloud node, we see that $L = (2n - 1 - r) \bmod (n + 1)$.

Therefore, we know that

$$L = (x + y) \bmod (n + 1) = (2n - 1 - r) \bmod (n + 1). \quad (1)$$

We use an algorithm to improve recovery bandwidth performance and ensure data integrity. The details are given in Algorithm 1.

3.3.3. The Design of Functional Repair. F-MSR code is a code using the functional repair method, which is an important foundation of BMCloud. In F-MSR code, the system utilizes EM to record the mathematical relation between the original data and encoded data. From the EM, the system can retrieve the original data. When faults occur, the system can repair them by calculating a linear transformation on EM and regenerating new encoded data on the cloud nodes.

The encoding process preserves the F-MSR code block. It just adds a parity block of E Code to the data, so that in daily use, there is no need for the system to decode the block twice. The added parity block will only be used in the repair process.

The reason why we choose E Code is that it introduces abundance by adding parity blocks. This method maintains the contents of the data.

The E Code layer can improve the stability of the system while not delaying the response time of the system.

3.3.4. The Design of Exact Repair. The deep gray squares are the parity strips of E Code. The other 12 strips are data strips which are partitions of F-MSR chunks. In Figure 2, we can

```

1: Requirements:
2: Native chunks;
3: Encoding request;
4: Main parameters ( $n, k, s$ ) in F-MSR code and E Code;
5: Step 1. System initialization
6: Step 2. F-MSR ( $n, k$ ) encoding
7: (a): Generate random encoding coefficient vectors;
8: (b): If MDS is satisfied
9: then generate an encoding matrix from then coding coefficient vectors;
10: else return to a;
11: (c): Compute the product of encoding matrix and native
12: chunks as F-MSR chunk;
13: Step 3. E Code ( $n, s$ ) encoding
14: (a): Divide each F-MSR chunk into stripes and strips;
15: (b): For stripe A in F-MSR chunks
16: Calculate the parity strip of this stripe;
17: Save the parity strip in certain position
18: (c): Consolidate the strips into extended F-MSR chunks

```

ALGORITHM 1: Algorithm E Code: encode algorithm of the exact layer.

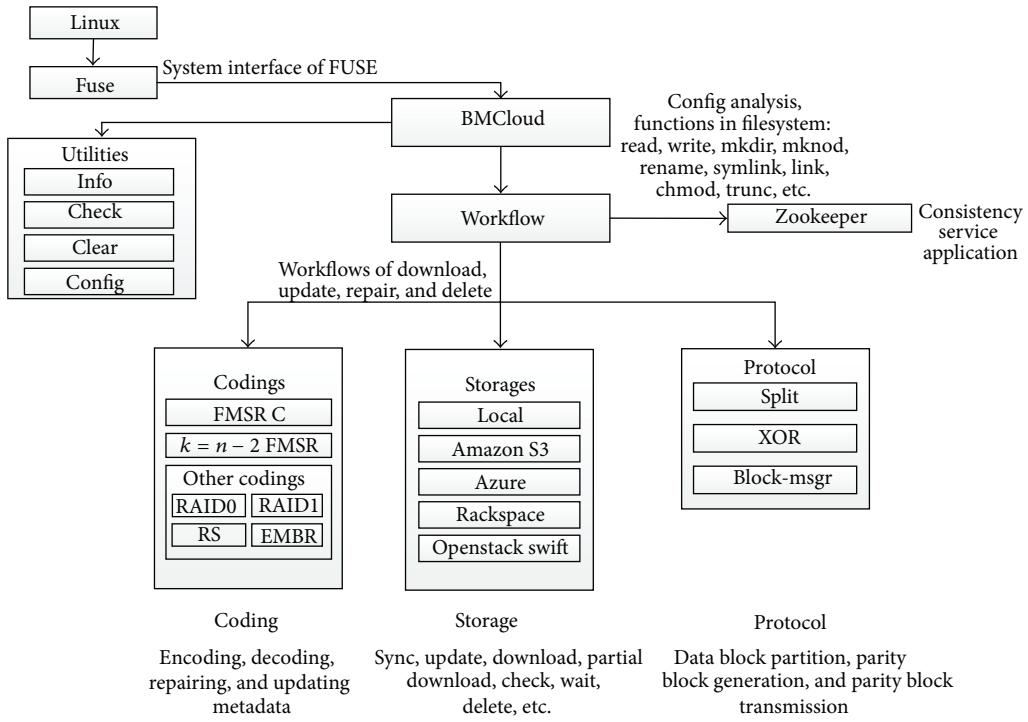


FIGURE 1: Software architecture of BMCloud.

see that every data strip belongs to two data links. If n is the number of the nodes, when $n > 3$, E Code can provide double-fault tolerance to protect data in nodes.

The dividing method provides system repair abilities in smaller granularity. That enables BMCloud to avoid repairing whole nodes in most situations. In other words, BMCloud avoids the cost of calculating the data of a whole node and updating bandwidth costs of EM.

3.3.5. Proxy Architecture. The prototype system of BMCloud is constructed of a proxy and several cloud servers (cloud nodes) in heterogeneous environments. Figure 3 shows a coding group of clouds, which provides storage capacities, redundancy, and computing.

As a controller, a proxy functions to coordinate data transmission between several cloud nodes and to maintain the system's consistency. In the idealized mode, the proxy

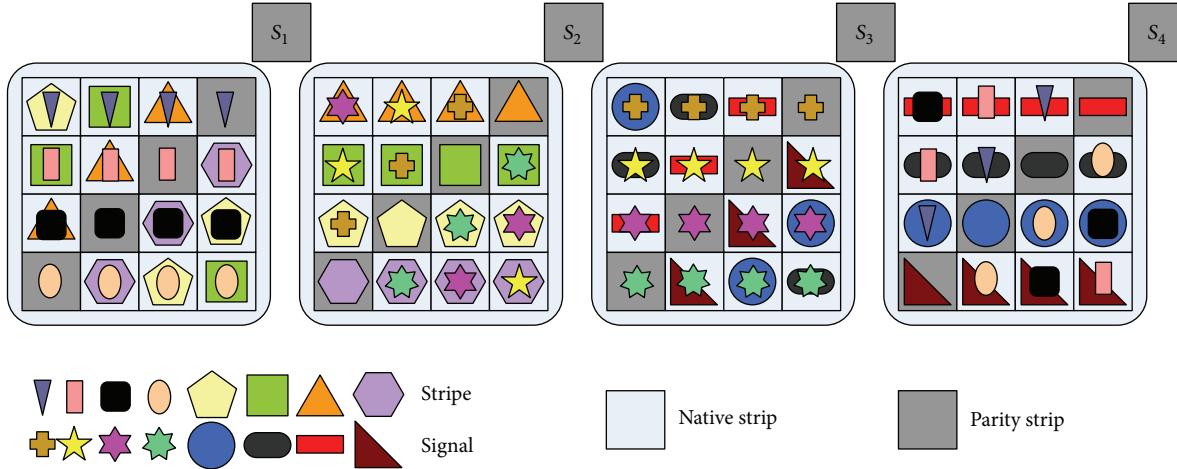


FIGURE 2: An example of an E Code encoded process.

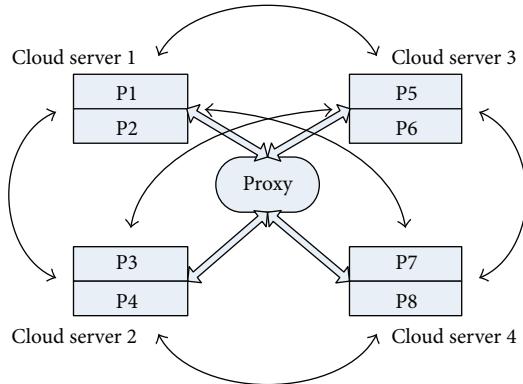


FIGURE 3: The communication ring of cloud servers.

has no need to compute or store any data, which strengthens the system's scalability. In a large-scale archive environment, proxy process requests remain limited and affordable to current servers or cloud servers. Therefore proxy will not become the bottleneck in this prototype.

Data packages could be directly transmitted between cloud nodes, cutting the bandwidth cost from multiple transmissions. Cloud nodes may need computing capacity to encode and decode the data, undertake the recoveries, and respond to instructions from proxies or requests from other cloud nodes.

We employ a distributed structure in which all the cloud nodes and proxy functions have ability to calculate independently, so that it is necessary to arrange a unified protocol to control and maintain the system.

As with single-fault recovery, there are two equivalent strategies, recovery with the last node or recovery with the next node. In the former case, there will be two kinds of orders from the proxy, the ones to the previous node and the ones to the recovering node. In order to prompt this, we will send an order formatted as follows for prenode (k): $0 \leq k \leq n-2$ or $k = n$. K is the stripe sequence. This order

will trigger a repeated XOR operation in the corresponding stripe and the result will be sent to the next node, as the recovery node. Before this order, the proxy should send an order to the recovering node, defined as recnode (x, y) . x and y are, respectively, the row and column sequence of the fault. When the recovering node receives this order, it will wait for the coming strips from the previous node. After it is received, the recovering node will operate XOR on the strip and other strips in the row with fault and the result will be the strip needing recovery.

4. The Implement Algorithms

4.1. Encode Algorithms

4.1.1. Encode with F-MSR. Encode the original data with F-MSR code. In this process, the encoded data is divided and distributed to several cloud nodes.

4.1.2. Divide the F-MSR-Encoded Data into E Code Chunks and Add Parity Chunks. Every F-MSR chunk is divided into $(n - 1)^*n$ strips in this step. The strips will be mapped into certain positions. Strips on the diagonal will be reserved for parity strips in step 3.

4.1.3. Calculate Parity Blocks and Upload Data. Calculate the data of the parity strips and insert it into the reserved space in the second step. Figure 2 is an example of a block location. After the calculation, the extended F-MSR chunk will be uploaded to the cloud.

Furthermore, since BMCloud is developed on the foundation of F-MSR code, the system has the cloud level fault tolerance ability. When one cloud node fails, the system can repair the data from the surviving node.

Every neighboring cloud node pair is defined as an E Code area. Any stripe belongs to one and only one E Code Area. In an E Code area, the cloud node on the left owns

$(n - 1)$ strips in the stripe, which is leaning towards the top-right corner and the right cloud node has n strips in the stripe showing as a row in Figure 2.

4.2. JUDGE_STYLE Algorithm. In BMCloud, most strips are located on two data-links. In order to improve the recovery ability of BMCloud, we added a parity strip S_n which is the checksum of all the parity blocks on each node (in order to make the Figures more clear, we have hidden S_n in the following figures.). When a fault occurs, the system can search the data link which owns the failed strip and repairs it with the surviving part of the data link. We now take an overview of the related work on the recovery of faults in different scales. We propose the JUDGE_STYLE algorithm, which can judge what kind of style the system will be used. We classify existing recovery solutions into three families, namely exact recovery, enumeration recovery, and functional recovery. Respectively, they are designed for faults of small, medium, and large scales.

4.2.1. Exact Recovery. E Code in our system has excellent recovery abilities. When the number of the failed strips is less than or equal to 4, BMCloud will recover the data with an exact recovery policy. All faults in this scale can be repaired by the E Code layer exactly. In extended-F-MSR chunks, we divide the strips into two families: data strips and parity strips. In these cases, we use vectors (m, n) to represent the situation where there are m failed data strips and n failed parity strips.

(a) $(0, 4)$. Since there are 4 failed parity strips for different data links, which all have only one failed strip, it is obvious that these 4 parity strips can be easily repaired. In the following situations, we will skip all similar relationships.

(b) $(1, 3)$. In this case, there are 3 failed parity strips and 1 failed data strip. If the one failed data strip is in the same data link with the one parity strip, then the other two parity strips can be repaired and the system can repair the rest of the parity strip by using a S1 parity strip. Finally only the data strips can be repaired.

(c) $(2, 2)$. In this situation, we will give an example of every possible approach. We use the symbol $S(m, n)$ to represent the n th strip in the m th cloud.

(i) $S(1,3), S(1,6), S(1,10), S(1,13)$: they can be easily repaired because they are all in a data link, which has only one failed strip.

(ii) $S(1,3), S(1,6), S(1,7), S(1,10)$: $S(1,3)$ and $S(1,10)$ can be repaired first then, the recovery trace of $S(1,6)$ and $S(1,7)$ is obvious.

(iii) $S(1,2), S(1,3), S(1,4), S(1,7)$: $S(1,7)$ first, and from S1 we can repair $S(1,4)$, then $S(1,2)$ can be recovered by a data-link-square, and $S(1,3)$ can by a data-link-upside-down triangle.

(iv) $S(1,3), S(1,4), S(1,6), S(1,7)$: from four involved data-links, we get the equitation set as follows:

$$\begin{aligned} S(1,3) + S(1,4) &= R_1, \\ S(1,3) + S(1,6) &= R_2, \\ S(1,4) + S(1,7) &= R_3, \\ S(1,6) + S(1,7) &= R_4. \end{aligned} \quad (2)$$

R_{1-4} is the result the system calculates from survived strips in the involved data links. So from the equitation set we can calculate the data of the four failed strips.

(d) $(3,1)$ and $(4,0)$. These situations are similar to the situations in $(0,4)$ and $(1,3)$, the repair trace is simple and easy to find so we will not introduce it in detail.

4.2.2. Enumeration Recovery. When the numbers of the failed strips are between 4 and 10. The E Code layer can handle these faults except for in some special situations. So first, BMCloud will scan all the failed strips and check their relationships. Then, the system will try to use a greedy algorithm to repair the fault. The details are in Algorithm 2.

4.2.3. Functional Recovery. When the number of failed strips is larger than 10, the scale of the fault overcomes the upper bound of repair ability of the E Code layer. So BMCloud will repair the fault by the F-MSR layer. The data of the whole node will be regenerated by F-MSR code from the data on surviving cloud servers. After regenerating the new F-MSR chunk, BMCloud will recalculate the parity strips and add them into the F-MSR chunk to restore and extend the F-MSR chunk onto a new cloud sever. In the meanwhile, the related parity strip on the related cloud server will be updated.

5. Evaluation Methodology

5.1. Cost Analyze. Table 2 shows the monthly price plans for three major vendors of cloud storage as of January 2013. We used the price of Azure [27] and assume that the storage usage was within 1 TB/month; data transferred out was more than 1 GB/month but less than 10 TB/month. From the analysis in Section 4, we can save 33.33% of the download traffic during storage repair when $n = 4$. The storage size and the number of chunks being generated per file object of BMCloud are 33.33% larger than RAID-6 when $n = 4$. Since the price of storage is much lower than the bandwidth, the redundancy of BMCloud is acceptable to the user.

However, in the analysis, we have ignored three practical considerations: the computing cost, the size of metadata, and the number of requests issued during repair, because we considered these values negligible in real-life applications.

Computing cost: in real-life applications, the MTTF (99.99999999%) of the Business Cloud is very long, so it will cost few computing resources to guarantee high availability.

Metadata size: in BMCloud, regardless of the size of the data files, the F-MSR metadata size is always within 160 B.

```

1: Requirements:
2:   Native chunks;
3:   Repair request;
4:   Main parameters  $(n, k, s)$  in F-MSR code and E Code;
5:    $k$  stands for the number of fail strips
6: Step 1. Basic information collection
7: Step 2. Greedy repair
8:   (a): Scan failed strip  $S_i$  ( $1 < i < k$ );
9:   (b): If  $S_i$  can be recovered by a simple data-link
10:    then restore  $S_i$ , go to Step 1.
11:    else  $i = i + 1$ ;
12:   (c): List the equation set of the rest of the failed strips
13:      from the related data-links
14:   (d): If the equation set is soluble
15:      then restore all the failed strips
16:      else got Step 3.
17: Step 3. Functional Repair
18:   (a): Repair the whole node with F-MSR code;
19:   (b): Recalculate the related parity strip and update all
20:      the EMs on every cloud server;

```

ALGORITHM 2: Algorithm enumeration recovery.

TABLE 2: Monthly price plans (in US dollars) for Amazon S3 (US Standard) and Windows Azure Storage, as of January, 2013.

	S3	Azure
Storage (per GB)	\$0.064	\$0.062
Data transfer in (per GB)	free	free
Data transfer out (per GB)	\$0.120	\$0.119
PUT, POST (per 10 K requests)	\$0.100	\$0.010
Get (per 10 K requests)	\$0.010	\$0.010

In evaluation, we used a 512 MB file to test the response time of BMCloud, and compared to the size of the test file, the metadata size will usually negligible. In real-life applications, the size of the data file usually overcomes 1 GB, so the effects caused by metadata on system are too small and do not need evaluation.

Number of Requests. From Table 2, we know that the charge of requests is relatively low compared to storage and bandwidth. RAID-6 and F-MSR differ in the number of requests when recovering data during repairs. Suppose that we store a file of size 4 MB with $n = 4$ and $k = 2$. BMCloud may need special protocol to support this function. When we repair cloud data, there is an agreement that it is merged before the data is transferred over. But if you do not follow these steps, the Get Operation will put all the data transmitted over a result of the flow rate increase.

5.2. Response Time Analysis. In this part, we deploy our BMCloud prototype in a real local cloud environment to evaluate the system performance of the response time. This cloud storage environment is chosen to carry out this analysis in order to evaluate the performance without the effects

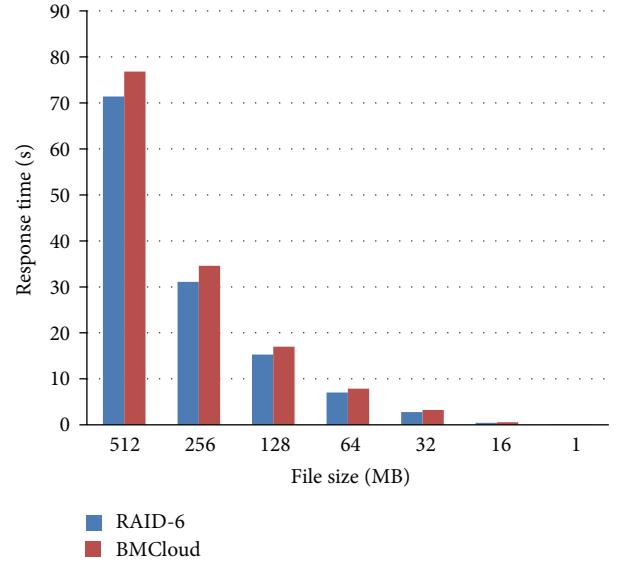


FIGURE 4: File upload response times of BMCloud on a local cloud.

of network fluctuations. We may continue the analysis on commercial clouds as a future work. All results are averaged over 50 runs.

The experiment is implemented on a storage platform based on OpenStack swift. The proxy of the system is installed on a laptop with Intel Core i5-580 and 8 GB RAM. This machine is connected to an OpenStack swift platform attached to a number of storage servers with Xeon E5606 CPU and 8 GB DDR3 RAM. We create 6 containers on this platform, and 4 containers act as 4 cloud nodes and other 2 containers act as spare nodes, to constitute the testing environment of F-MSR ($n = 4, k = 2$) and BMCloud system.

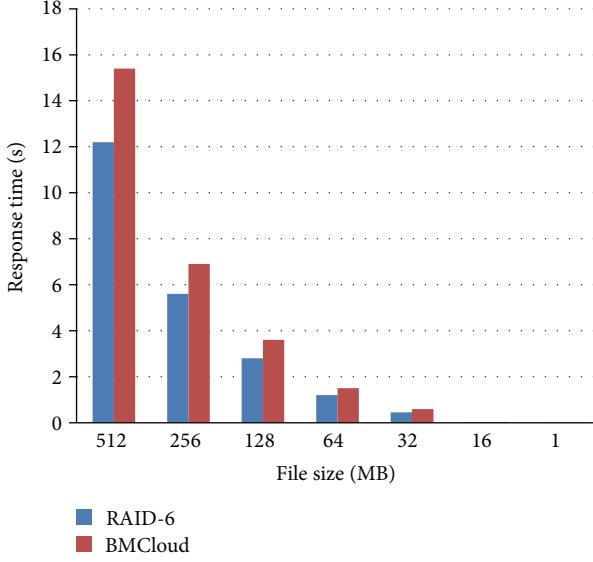


FIGURE 5: File download response times of BMCloud on a local cloud.

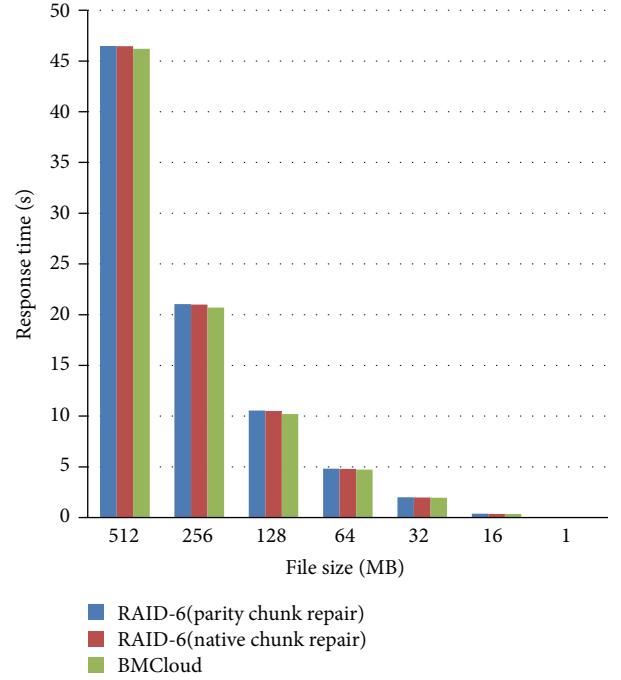


FIGURE 6: Recovery response times of BMCloud on a local cloud.

We test the response time of the three most important operations in the workflow of BMCloud: file upload, file download, and file recovery. For each operation workflow, we keep a record of the detailed time costs in every type of operation. We use random files sized from 1MB to 512 MB as the test data set. RAID-6 Reed-Solomon code is chosen as a control group. There are two types of recovery situations, considering whether the failed node is native or parity.

Figures 4, 5 and 6 show response time in 3 main operations workflows, upload, download, and recovery in BMCloud and RAID-6 RS code. BMCloud performs over RAID-6 on response time in file upload and file download.

Figures 7 and 8 show detailed component of the response time in the case of the 512 MB file. We can clearly see that the data transfer time contributes to the main part of the response time for all 3 operation workflows and time costs in both methods are quantitatively similar. In contrast to RAID-6 code, F-MSR code properties make BMCloud present a significant encoding/decoding overhead when a file is uploaded or downloaded. When uploading a 500 MB file, RAID-6 takes 2.496 s to encode and BMCloud takes 12.065 s; when downloading a 500 MB file, BMCloud takes 4.512 s to decode and RAID-6 needs no decoding when native nodes are available. However, uploads and downloads are infrequent operations in an archive storage environment. Moreover, network fluctuation in real environments will balance the difference between RAID-6 and BMCloud.

In the recovery process, BMCloud shows a shorter response time than RAID-6. BMCloud needs to download less data during repairs than RAID-6 and NCCloud. There are two kinds of recovery methods in BMCloud, exact repair, and functional repair. Exact repair sharply curtails repair bandwidth and hence repair response time, showing our main advantage. In repairing a 512 MB file, NCCloud spends

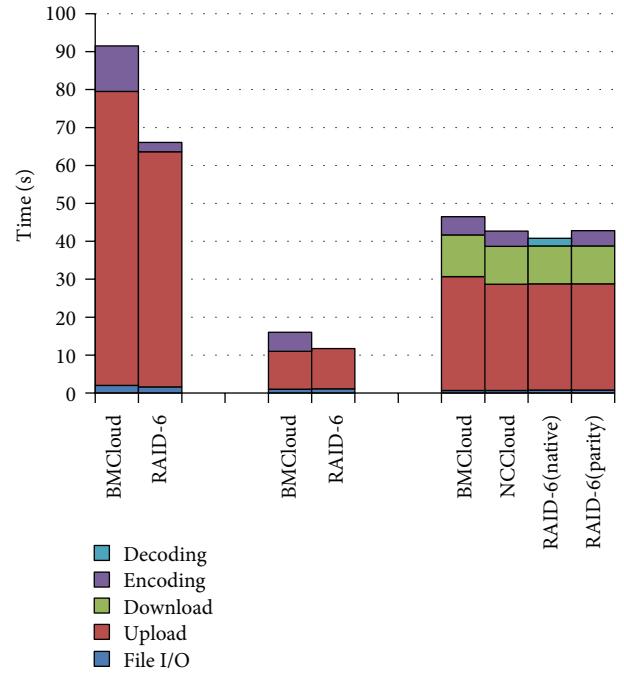


FIGURE 7: 512 MB file response time detailed analysis (Functional).

9.593 s in download; the native-chunk repair while RAID-6 spends 12.124 s and BMCloud spends only 5.583 s for all the data. The response time of BMCloud is 41.8% and 53.9% better than that of NCCloud and Raid-6, respectively. On the other hand, BMCloud spends 11.467 s in functional repair, which is a little less than NCCloud and RAID-6.

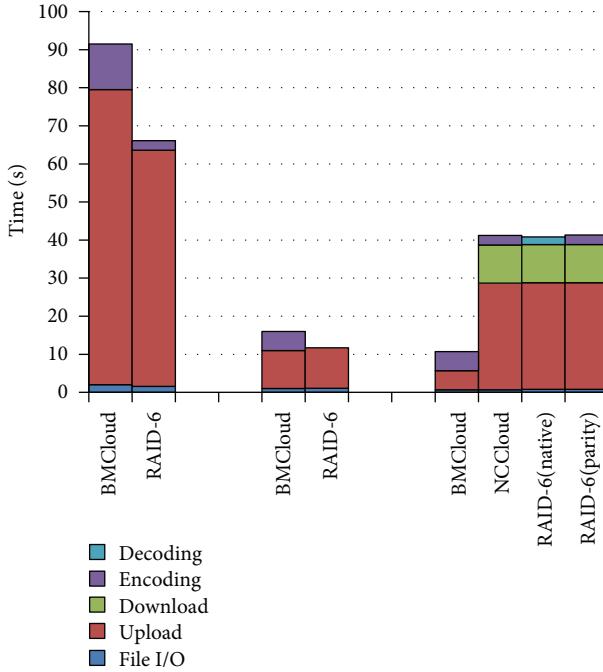


FIGURE 8: 512 MB file response time detailed analysis (exact).

We have not tested our system on a commercial cloud since our test environment is limited. But from NCCloud [5], we can see that network fluctuation plays a bigger role in determining the response time. So it will be proved that BMCloud will have no performance in functional repair and have good performance when it is in exact repair.

6. Conclusions

In this paper, we developed a low repair bandwidth, low maintenance cost cloud storage system named BMCloud. It has the exact repair algorithm E Code to degrade repair bandwidth and it also provides functional repair to recover data in any condition. The JUDGE_STYLE algorithm can help BMCloud decide in which cases exact repair will be used and in which cases functional repair will be used.

We implemented the system and conducted experiments which prove that BMCloud is effective in degrading repair bandwidth and maintenance costs. The result shows that the response time of BMCloud is 41.8% and 53.9% better than those of NCCloud and Raid-6 when the system operates in the recovery methods of exact repair.

There are still much more work to be done in the future which mainly take two directions. First, we will focus on the recovery of a single-disk failure and do some experiments to verify the performance of BMCloud. Second, we will take a twin-code model into account and make the system more suitable for cloud storage.

In conclusion, we believe that BMCloud is an attractive cloud storage system: one that offers low repair bandwidth, while achieving low maintenance cost.

Acknowledgment

This work is supported in part by the National Basic Research Program of China under Grant no. 2011CB302303, the NSF of China under Grant no. 60933002, and the National High Technology Research and Development Program of China under Grant no. 2013AA013203.

References

- [1] Wikipedia. Cloud computing, 2011, http://en.wikipedia.org/wiki/Cloud_computing.
- [2] N. Leavitt, "Is cloud computing really ready for prime time?" *Computer*, vol. 42, no. 1, pp. 15–25, 2009.
- [3] B. Calder, J. Wang, A. Ogus et al., "Windows azure storage: a highly available cloud storage service with strong consistency," in *Proceedings of the 23rd ACM Symposium on Operating Systems Principles (SOSP '11)*, pp. 143–157, October 2011.
- [4] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The google file system," in *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP '03)*, pp. 29–43, October 2003.
- [5] Y. Hu, H. Chen, P. Lee, and Y. Tang, "NCCloud: applying network coding for the storage repair in a cloud-of-clouds," in *Proceedings of the USENIX FAST*, 2012.
- [6] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4539–4551, 2010.
- [7] O. Khan, R. Burns, J. S. Plank, W. Pierce, and C. Huang, "Rethinking erasure codes for cloud file systems: minimizing I/O for recovery and degraded reads," in *Proceedings of the USENIX FAST*, 2012.
- [8] D. A. Patterson, P. Chen, G. Gibson, and R. H. Katz, "Introduction to redundant arrays of inexpensive disks (RAID)," in *Proceedings of the ACM SIGMOD International Conference*, pp. 112–117, February 1989.
- [9] M. Blaum, J. Brady, J. Bruck, and J. Menon, "Evenodd: an efficient scheme for tolerating double disk failures in raid architectures," *IEEE Transactions on Computers*, vol. 44, no. 2, pp. 192–202, 1995.
- [10] P. Corbett, B. English, A. Goel et al., "Row-diagonal parity for double disk failure correction," in *Proceedings of the USENIX FAST*, 2004.
- [11] L. Xu and J. Bruck, "X-code: MDS array codes with optimal encoding," *IEEE Transactions on Information Theory*, vol. 45, no. 1, pp. 272–276, 1999.
- [12] J. Wan, C. Yin, J. Wang, and C. Xie, "A new high-performance, energy-efficient replication storage system with reliability guarantee," in *Proceedings of the 28th Conference on Mass Storage Systems and Technologies*, 2012.
- [13] S. Wan, Q. Cao, C. Xie, B. Eckart, and X. He, "Code-M: a non-MDS erasure code scheme to support fast recovery from up to two-disk failures in storage systems," in *Proceedings of the 40th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN '10)*, pp. 51–60, July 2010.
- [14] Y. Wu, A. G. Dimakis, and K. Ramchandran, "Deterministic regenerating codes for distributed storage," in *Proceedings of the Allerton Conference on Control, Computing and Communication*, Monticello, Ill, USA, 2007.
- [15] Y. Wu, "Existence and construction of capacity-achieving network codes for distributed storage," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 2, pp. 277–288, 2010.

- [16] A. Jiang, “Network coding for joint storage and transmission with minimum cost,” in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '06)*, pp. 1359–1363, July 2006.
- [17] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, “Network coding for distributed storage systems,” *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4539–4551, 2010.
- [18] K. V. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, “Exact regenerating codes for distributed storage,” in *Proceedings of the Allerton Conference on Control, Computing and Communication*, Urbana, Ill, USA, 2009.
- [19] Y. Wu and A. G. Dimakis, “Reducing repair traffic for erasure coding-based storage via interference alignment,” in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '09)*, pp. 2276–2280, Seoul, Republic of Korea, July 2009.
- [20] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, “Explicit codes minimizing repair bandwidth for distributed storage,” in *Proceedings of the IEEE Information Theory Workshop*, 2010.
- [21] K. V. Rashmi, N. B. Shah, and P. V. Kumar, “Enabling node repair in any erasure code for distributed storage,” in *Proceedings of the Allerton Conference on Control, Computing and Communication*, Urbana, Ill, USA, 2011.
- [22] V. R. Cadambe and S. A. Jafar, “Interference alignment and degrees of freedom of the K-user interference channel,” *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3425–3441, 2008.
- [23] C. Suh and D. Tse, “Interference alignment for cellular networks,” in *Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing*, pp. 1037–1044, Urbana, Ill, USA, September 2008.
- [24] R. Bhagwan, K. Tati, Y. Cheng, S. Savage, and G. Voelker, “Total recall: system support for automated availability management,” in *Proceedings of the NSDI*, 2004.
- [25] J. Kubiatowicz, D. Bindel, Y. Chen et al., “OceanStore: an architecture for global-scale persistent storage,” in *Proceedings of the 9th International Conference Architectural Support for Programming Languages and Operating Systems*, pp. 190–201, November 2000.
- [26] M. Vrable, S. Savage, and G. Voelker, “Cumulus: Filesystem backup to the cloud,” in *Proceedings of the USENIX FAST*, 2009.
- [27] G. DeCandia, D. Hastorun, M. Jampani et al., “Dynamo: Amazon’s highly available key-value store,” in *Proceedings of the 21st ACM Symposium on Operating Systems Principles (SOSP '07)*, pp. 205–220, October 2007.

Research Article

Structural Attack to Anonymous Graph of Social Networks

Tieying Zhu, Shanshan Wang, Xiangtao Li, Zhiguo Zhou, and Riming Zhang

School of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, China

Correspondence should be addressed to Rimeng Zhang; zhangrm280@nenu.edu.cn

Received 19 June 2013; Revised 7 October 2013; Accepted 18 October 2013

Academic Editor: Siddhivinayak Kulkarni

Copyright © 2013 Tieying Zhu 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.

With the rapid development of social networks and its applications, the demand of publishing and sharing social network data for the purpose of commercial or research is increasing. However, the disclosure risks of sensitive information of social network users are also arising. The paper proposes an effective structural attack to deanonymize social graph data. The attack uses the cumulative degree of n -hop neighbors of a node as the regional feature and combines it with the simulated annealing-based graph matching method to explore the nodes reidentification in anonymous social graphs. The simulation results on two social network datasets show that the attack is feasible in the nodes reidentification in anonymous graphs including the simply anonymous graph, randomized graph and k -isomorphism graph.

1. Introduction

A social network is a social relation structure which is made up of a set of social entities and their social ties or interactions. The research of social network analysis can be traced back to the contributions of Moreno who discusses the dynamics of social interactions within groups of people [1]. Nowadays, with the emerging of online social networks and services, the social relationship in reality has been extended to the virtual network world. Billions of users use online social website making friends, sharing pictures, micrologging and so on. The social structure hidden in the social network data are valuable for social analysis for the purpose of commerce or academy. For example, the user behavior and interests derived from social data are important for all the commercial recommendation systems [2, 3]. At the same time, more and more attentions are being paid to the privacy preservation problems in the process of using social networks and sharing social data, since data publishing and exchange increase the risk of disclosure and leakage of personal information of social network users [4, 5].

Social networks are usually modeled as graphs, in which the vertices represent social entities and the edges represent the social links or social ties. The properties of entities, such as age, gender, and SIN. can be represented as the attributes of vertices, and the properties of links between

entities, such as the tightness of social ties, can be shown as the edge label or weight. Therefore, the natural and simple way to prevent the disclosure of personal information of social users is to remove the user portfolios, such as names and SIN, or replace them with random identifications. But the simple method cannot prevent the disclosure of personal sensitive information. The earliest privacy event causing public attention is the publishing of email data set of Enron Corpus [6]. Although the original purpose is for legal investigation, the regularity of email communications among employees within the company, even the organization structure of Enron Corpus, can be inferred from the email data. Other personal information disclosure events include the AOL Company publishing anonymized user search data for the research in search engine [7], and Netflix Company publishing user movie scoring data for improving the movie recommendation systems [8]. All of the intended purposes of these data publishing issues are not to leak users' information, but it results in the privacy risks.

On the other hand, many privacy-preserving methods have been put forward and examined including k -anonymity based privacy preservation via edge modification, probabilistic privacy preservation via edge randomization and privacy preservation via clustering and generalization (see the recent review papers [9–11]). Besides these methods, the differential privacy method, which depends on specific

privacy guarantees and aims to make users in released data computationally indistinguishable from most of the other users in that data set, are paid more and more attention recently [12, 13].

In the paper, we present a structural attack method to deanonymize social graph data, called n -hop neighbor Feature for Node Reidentification (n -hop neighFNR). The method relies only on the network structure. It uses the cumulative degree of n -hop neighbors as the regional feature and combines with the simulated annealing-based graph matching method. With the aid of auxiliary graph, it can reidentify the nodes in anonymized social graphs. The simulations on two data sets including Karate clubs [14] and email networks of URV [15] show it is feasible on deanonymizing social graphs including the simple anonymous graph, randomized anonymous graph and k -isomorphism graph.

The rest of the paper is organized as follows. Section 2 presents the related work, and Section 3 describes the definition of n -hop neighbor feature and the node reidentification algorithm, followed by the experiments results on data sets in Section 4. Finally, we conclude the paper in Section 5.

2. Related Work

In the graph data of social networks, nodes usually correspond to the users in social networks and edges correspond to the relationship between users. The privacy attack to graph data of social networks aims to obtain the sensitive information including identity, friendship and other personal information that is hidden in social networks.

Backstrom et al. [16] firstly proposes the active and passive attack to simple anonymous social graphs. These two methods try to identify the target in the released social graph. The difference between them is whether the attackers change the graph data before data publishing. In active attacks, the adversary can create a certain number of Sybil nodes and edges linked to the target and embed these node and edges into the graph before data publish, then find these “Sybil” nodes together with the targeted users in the anonymized network. In passive attacks, attackers try to discover a target using their knowledge of local structure of the network around the target.

Different privacy attacks depend on different background knowledge. Zhou et al. [10] generalize some possible background knowledge that can be used in the privacy attacks. The background information includes degree, attribute of nodes, special links with the target node, neighbors, embedded sub-graphs, and other properties of graphs such as betweenness, closeness centrality and etc. Some literature [17–22] discusses different background knowledge and the corresponding privacy protection methods. Literature [17] proposes the known degree attack and corresponding k -degree anonymous graph for solving the problem. Literature [18] presents a degree trace attack which traces the change of certain node degree in the evolution graphs to reidentify the target node. Other privacy attacks are based on the structure of 1-hop neighbor [19] or neighbor subgraph and the corresponding privacy preserving methods are usually

based on k -anonymity methods in structure, such as k -automorphism [20], k -isomorphism [21] and k -symmetry model [22]. Compared with these k -anonymity methods, edge Randomization is a generalized privacy preservation methods which is not specific to the privacy attacks.

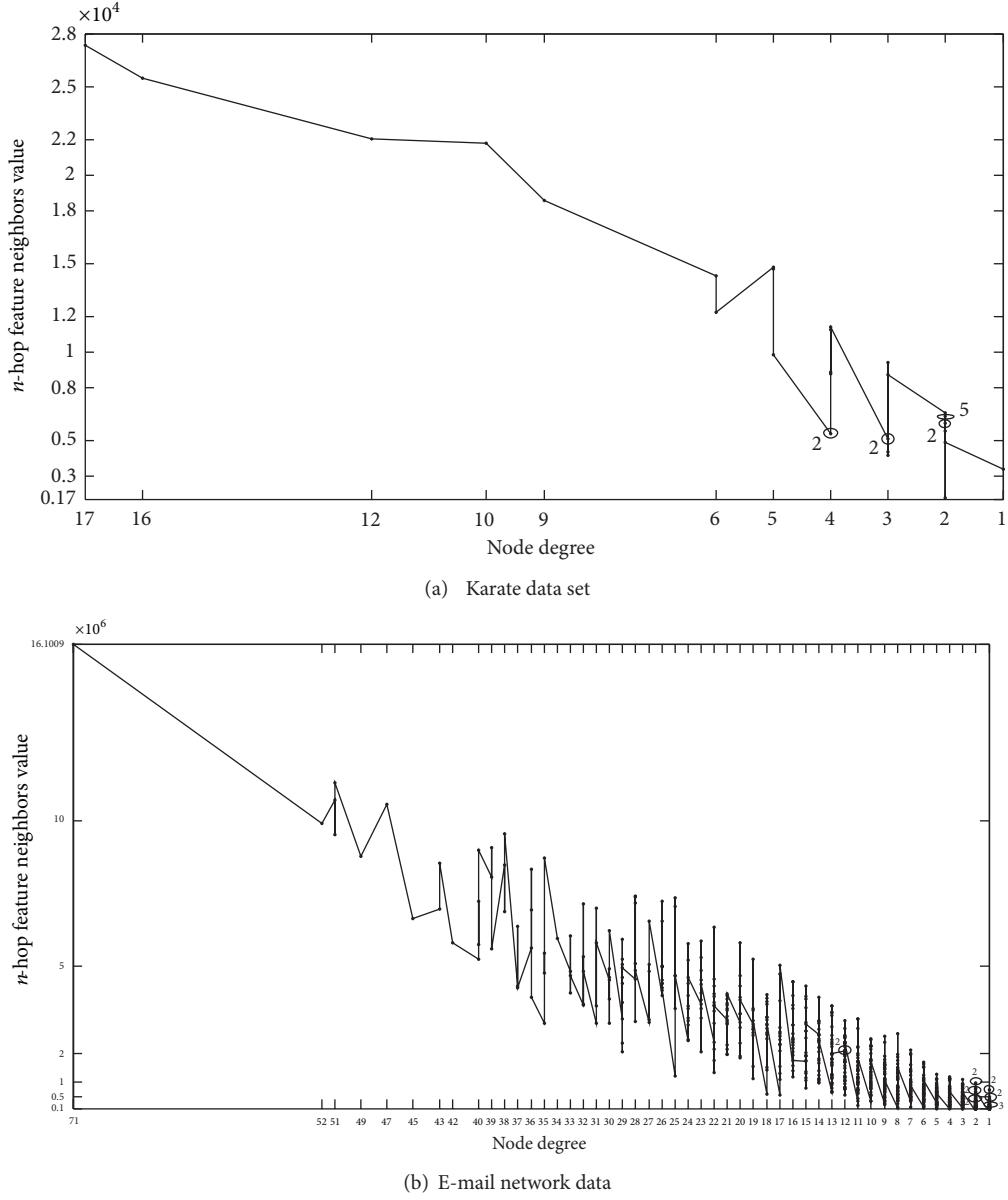
The de-anonymizing social network based on the auxiliary graph is a feasible attack which can recognize nodes from the large scale social networks. The auxiliary graph which can be obtained by crawling, is used to match the anonymized graph and reidentify the targets on the viewpoint of graph structure. Literature [23] proposes this method and use it to reidentify a third of the users who have accounts on both Twitter and Flickr with small error rate. In such attack, there are usually two phases: the recognition of certain amount of seed nodes in the anonymous graph, and then the propagation process to match the rest of nodes in the auxiliary graph with the targets in the anonymous graph on the basis of the known seeds. Rattigan [24] employs the crawling data of Yahoo! Music data as the auxiliary graph and attends to recognize the artist in the data set of KDD Cup 2011. Although there is no ground-truth graph to show the accuracy of this work, it still shows the feasibility of such attack. More recently, Narayanan et al. [25] use the crawling Flickr graph to deanonymize much of the competition test set of a machine learning contest in Kaggle challenge. They use the neighbor similarity between node pairs as a structural feature and combine it with simulated annealing-based graph matching method to reidentify a small number of nodes with largest degree. These recognized nodes can be used as the seed nodes in the first phase of de-anonymizing attack.

In privacy attacks mentioned above, the degree, neighbor structure, neighbor similarity of certain node pair are all metrics used to match or recognize the target nodes in anonymized graphs. In graph mining, Henderson et al. [26] present the concept of recursive feature, which combines node-based local features with egonet-based neighborhood features to capture the regional information of an individual node in large graphs. It can be applied in de-anonymization tasks on evolution graphs or partially anonymized graphs. Influenced by these ideas, the paper proposes a structural attack, which combines the cumulative n -hop neighbors’ degree feature with the simulated annealing algorithm, to reidentify the nodes in anonymized graphs.

Many computational intelligent algorithms for optimization problems are proposed like in literature [27–31]. In this paper, we use simulated annealing method [32] to match the auxiliary graph with anonymized graph, although other intelligent algorithms can be used to replace simulated annealing method.

3. n -Hop Neighbor Feature for Node Reidentification

3.1. n -Hop Neighbor Feature. A social network can be modeled as a graph $G = (V, E)$, where V is the set of nodes and E is the set of connections. In this paper, undirected graph is used, although social networks can be directed graphs if the direction of connections is considered. In a graph structure,

FIGURE 1: Discrimination of n -hop neighbor feature; $n = 4$.

the node degree is the basic local feature of a node. When considering the relation of a node and its 1-hop neighbors, the related metrics are computed in the range of egonet. For example, in literature [25], the cosine similarity between a pair of nodes is defined as: $|x \cap y| / \sqrt{|x| \cdot |y|}$, while x and y are the neighbor sets of a node pair, respectively.

In this paper, we considered the cumulative degree feature of a node in the range of n -hop neighbors. For a node i , its cumulative degree is defined as the sum of n -hop neighbors' degree of a node and denoted as follows:

$$\sum_{j_1 \in \text{neigh}(i)} \sum_{j_2 \in \text{neigh}(j_1)} \dots$$

$$\sum_{j_{n-1} \in \text{neigh}(j_{n-2})} \sum_{j_n \in \text{neigh}(j_{n-1})} \text{degree}(j_n). \quad (1)$$

The n -hop neighbor feature is a regional feature and captures the node's properties better than the node degree since for the nodes with same degree would have different value of cumulative value n -hop neighbor degree. It qualifies the connections of a node with other nodes and shows the importance of a node in the n -hop range and even the whole network.

In order to show the discrimination of the n -hop cumulative feature, we analyze two data sets: karate club data and email network of URV, in terms of the value of degree and n -hop cumulative feature; here, $n = 4$. In karate data set, there

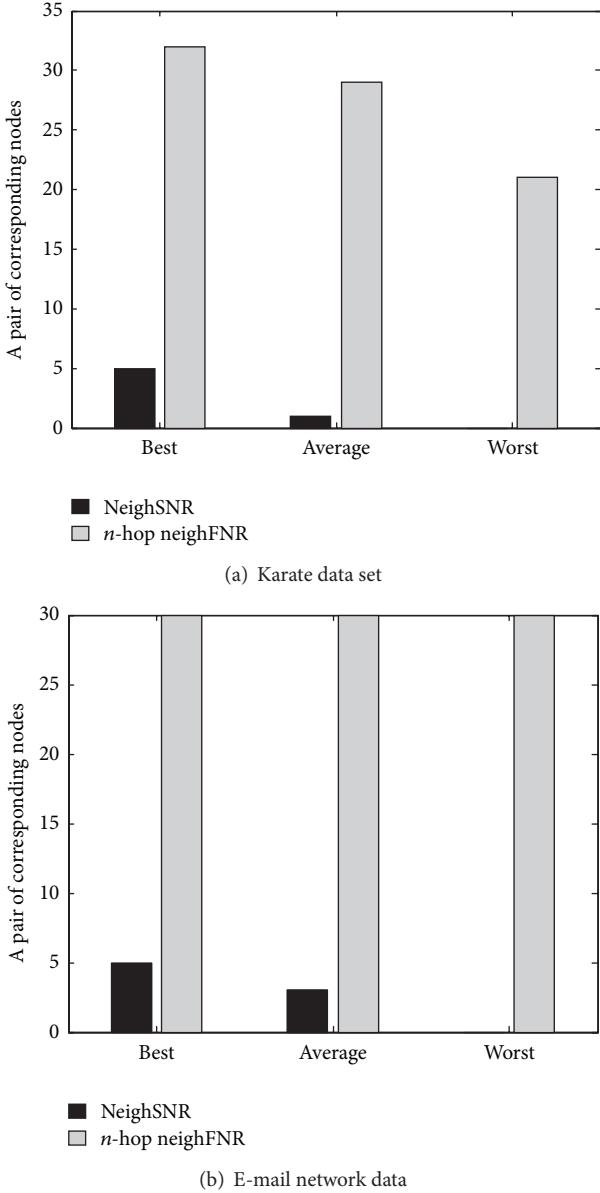


FIGURE 2: The results on simple anonymous graph for karate data set and e-mail network data.

are 34 nodes and 156 edges, while there are 1133 nodes and 5451 edges in email data set. Figures 1(a) and 1(b) shows that n -hop neighbor features discriminates nodes much better than the feature of degree on both two data sets. In these figures, x presents the degree of nodes in decreasing order, and y presents the corresponding 4-hop neighbor feature value. In a graph, there may be some nodes with the same degree value, and the nodes with the same degree value may also have the same or different 4-hop neighbor features; therefore we use the number besides the circle to denote how many nodes have the same n -hop neighbors feature value. For example, in Figure 1(a), in the 5 nodes with same degree 4, 3 nodes have different n -hop neighbor feature value and only 2 nodes have same n -hop neighbor feature value. Furthermore,

there are 11 nodes with the same degree 2, while there are 5 nodes with same n -hop neighbor feature value of 6376 and another 2 nodes with the other n -hop neighbor feature of 5977. These figures show that most of the nodes with the same degree have different 4-hop neighbor feature value on the two data sets. Specially, Figure 1(b) shows that most of the nodes with large degree value in email network data can be discriminated by 4-hop neighbor feature value. Although the discrimination becomes worse for some lower degree value, like 3 or 2, it is still better than the degree feature.

3.2. n -Hop Neighbor Feature for Node Reidentification. The n -hop neighbor feature can capture the regional information of a node in the graph. This paper combines n -hop neighbor feature with simulated annealing algorithm and proposes n -hop neighbor feature for node reidentification algorithm (n -hop neighborFNR) to deanonymize social networks by the aid of auxiliary graph.

For two graphs G_A and G_T , G_A is the auxiliary graph, in which the identities of nodes are already known, and G_T is the anonymous target graph. G_A and G_T can be thought as the induced graph from the same graph G . Usually the auxiliary graph G_A can be obtained by the crawling. G_T is the anonymous social data for publishing. The process of privacy attack can be thought as a matching process between the nodes of G_A and G_T . We use the original data sets as G_A , and three kinds of anonymous graphs as G_T , including the simple anonymous graph, which removes the identification of nodes, the randomize anonymous graph, which is obtained by randomly adding one edge followed by deleting another edge and repeating the process for k times, as shown in [33], and k -isomorphism graph, as shown in [21].

We combine the n -hop neighbor feature with the simulated annealing methods to match the two graphs, G_A and G_T . The Euclidian distance between two sets of n -hop neighbor feature value of node pairs is defined to measure the quality of a candidate mapping, so that we can optimize the matching over all the possible mapping. The Euclidian distance is defined as: $\sqrt{\sum_{i=1}^n (x_i - y_i)^2}$, in which x_i and y_i is the cumulative degree value of node i 's n -hop neighbors in G_A and G_T , respectively.

Algorithm 1 shows the method in n -hop neighbor feature for node reidentification. In the algorithm, Δp is the change of Euclidian distance in different matching processes. T is the temperature, which will be cooled with a rate of α ; $\alpha = 0.9$. The initiating value of temperature depends on the nodes in the graph; $T = 100 * n$. The ending of simulated annealing algorithm is determined by the threshold T_{\min} . c is constant, which is also dependent on the nodes, and $c = 20 * n$.

4. Experiment Results

In order to show the feasibility and effectiveness of n -hop neighbor feature, we compare it with the neighbor similar feature used in literature [25], which is called neighborFNR, to deanonymize the simply anonymous graphs, randomized graphs, and k -isomorphism graphs. In e-mail data set, we select 30 nodes with the largest degree

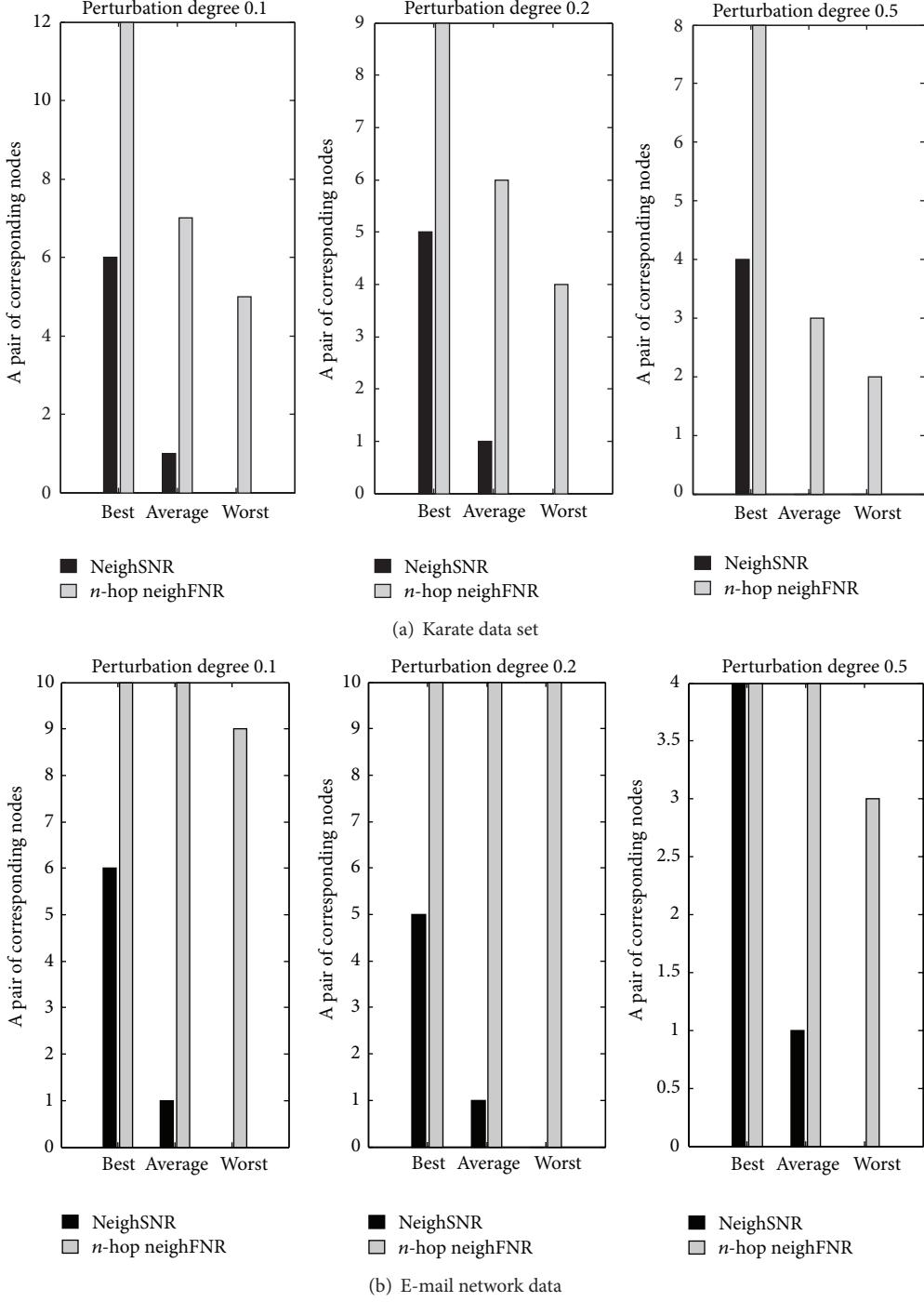


FIGURE 3: The results on randomized anonymous graph for karate data set and e-mail network data.

as the targets. The randomized graphs are obtained by adding or deleting edges randomly and repeatedly. The perturbation degree is defined as the percentage of the number of adding/deleting edges and we use 10%, 20%, and 50% as the perturbation degree, respectively. The k -isomorphism graph is obtained by adding or deleting edges to satisfy the definition of k -isomorphic: A graph G is k -isomorphic if G consists of k disjoint subgraphs g_1, \dots, g_k ,

where g_i and g_j are isomorphic for $i \neq j$. In the experiment, we use 2-isomorphic graph as the anonymous graph [21].

Figures 2(a) and 2(b) show the recognition results in simple anonymous graph. Our method n -hop neighFNR is much better than neighSNR. It can recognize 32 out of 34 nodes for karate data set in the best case and all the 30 largest degree nodes for email network data.

```

Input:  $G_A$  and  $G_T$ : the auxiliary graph and target anonymous graph
Output: node mapping between  $G_A$  and  $G_T$ 
Initialize the original mapping and temperature  $T$ ;
while (Temperature  $T >$  Threshold  $T_{\min}$ )
    Exchange two nodes position to make a new mapping;
        if (the Euclidian distance of the new mapping < the Euclidian distance of old mapping)
            Accept the changing of nodes position;
        else
            Accept it with a probability  $e^{-\Delta p/c*T}$ ;
        endif
        Decrease the temperature  $T$  with a rate;
endwhile

```

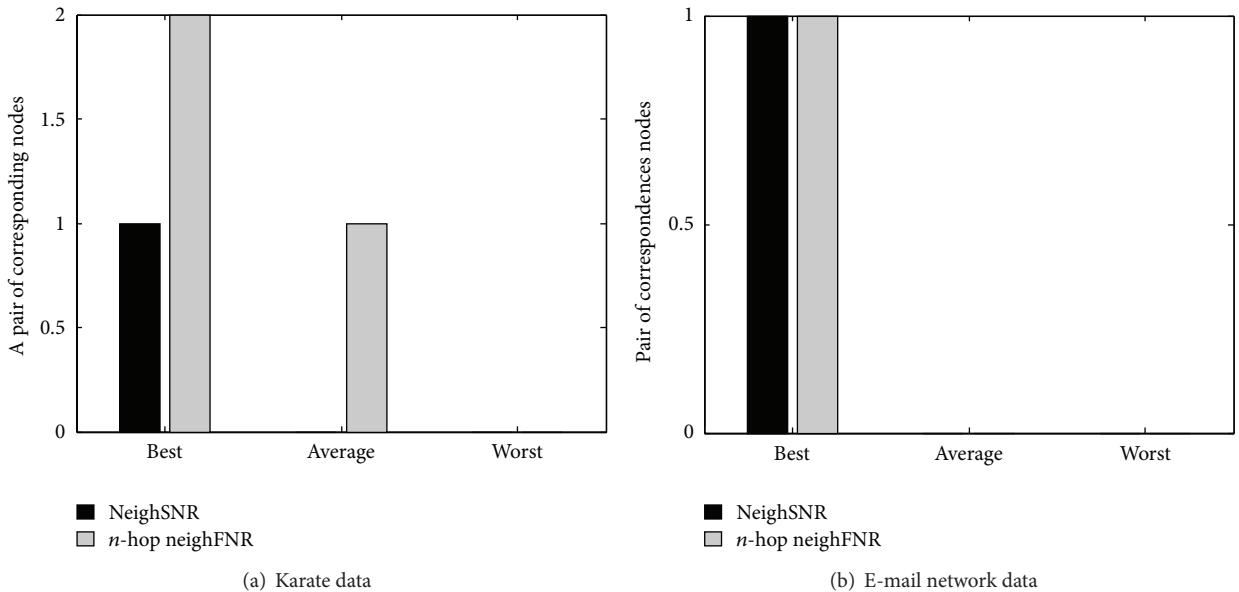
ALGORITHM 1: n -hop neighbor feature for node reidentification.

FIGURE 4: The results on 2-isomorphism anonymous graph for karate data set and e-mail network data.

Figures 3(a) and 3(b) show the results of the number of reidentification node in randomized anonymous graph with different perturbation degree for karate data and email network data. Although with the increasing of perturbation degree, the number of reidentification nodes decrease, our algorithm n -hop neighFNR outperforms NeighSNR in general. For karate data set, 12 node pairs are matched in the best case when our method is employed and perturbation degree is 10%. When perturbation degree increases to 50%, 8 node pairs are re-identified using our algorithm. For NeighSNR method, 4 node pairs are recognized when perturbation degree is 50% in the best case. For email network data, when perturbation degree is 10%, 10 nodes pairs are matched. When perturbation degree increases to 50%, both the n -hop neighFNR and NeighSNR method recognize 4 nodes pairs in the best case. In the average and worst cases, n -hop neighFNR also outperforms NeighSNR on both of two data sets.

Figures 4(a) and 4(b) show the de-anonymizing results on k -isomorphism graphs, in which $k = 2$. In the graph of Karate

data, 16 edges are added and 36 edges are deleted in order to generate 2-isomorphism anonymous graph. In the graph of Email data, 2428 edges are added and 2454 edges are deleted. Both NeighSNR and n -hop neighFNR does not work well on 2-isomorphism graphs, since these two 2-isomorphism anonymous graphs are obtained by perturbation of about 50% edges of the corresponding original graphs, and the k -isomorphism method enforces k -security for protecting the nodes and links in anonymous graph [21].

5. Conclusion

The paper presents n -hop neighbor feature to capture node characteristics in a graph. It uses the sum of degree of n -hop neighbors of a node as a regional feature. When combining with simulated annealing algorithm, it can be used as a structural attack to de-anonymize social networks. The experiments on two data sets show it is very effective for de-anonymizing the simple anonymous graph and feasible

for the randomized graph. The research provides insights for the privacy-preserving problem of social networks and the design of privacy-preserving algorithms. The future work we should do is to evaluate the effectiveness of our algorithm on large-scale real networks.

Acknowledgments

This work was supported in part by the Special Fund for Fast Sharing of Science Paper in Net Era by CSTD (FSSP 2012) and the Technical Development Plan of Jilin Province of China (No. 201101003).

References

- [1] J. L. Moreno, *Who Shall Survive?* Beacon House, Beacon, NY, USA, 1934.
- [2] I. Konstas, V. Stathopoulos, and J. M. Jose, "On social networks and collaborative recommendation," in *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)*, pp. 195–202, July 2009.
- [3] M. Maia, J. Almeida, and V. Almeida, "Identifying user behavior in online social networks," in *Proceedings of the 1st Workshop on Social Network Systems (SocialNets '08)*, pp. 1–6, Glasgow, UK, March 2008.
- [4] J. Becker and H. Chen, "Measuring privacy risk in online social networks," in *Proceedings of the Web 2.0 Security & Privacy Workshop (W2SP '09)*, pp. 1–8, Oakland, Calif, USA, 2009.
- [5] I. A. Tsoukalas and P. D. Siozos, "Privacy and anonymity in the information society: challenges for the European Union," *TheScientificWorldJournal*, vol. II, pp. 458–462, 2011.
- [6] J. Adibi and J. Shetty, "Discovering important nodes through graph entropy the case of Enron email database," in *Proceedings of the 3rd International Workshop on Link Discovery*, pp. 74–81, Chicago, Ill, USA, 2005.
- [7] S. Hansell, "AOL removes search data on vast group of web users," *New York Times*, 2006.
- [8] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proceedings of the IEEE Symposium on Security and Privacy (SP '08)*, pp. 111–125, May 2008.
- [9] X. Wu, X. Ying, K. Liu, and L. Chen, "A survey of algorithms for privacy-preservation of graphs and social networks," in *Managing and Mining Graph Data*, pp. 421–453, 2009.
- [10] B. Zhou, J. Pei, and W. S. Luk, "A brief survey on anonymization techniques for privacy preserving publishing of social network data," *SIGKDD Explorations*, vol. 10, no. 2, pp. 12–22, 2009.
- [11] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: a survey of recent developments," *ACM Computing Surveys*, vol. 42, no. 4, article 14, 2010.
- [12] C. Dwork, "Differential Privacy," in *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP '06)*, pp. 1–12, Venice, Italy, 2006.
- [13] C. Task and C. Clifton, "A guide to differential privacy theory in social network analysis," in *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM '12)*, pp. 411–417, 2012.
- [14] <http://www-personal.umich.edu/~mejn/netdata/>.
- [15] <http://deim.urv.cat/~aarenas/data/welcome.htm>.
- [16] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography," in *Proceedings of the 16th International World Wide Web Conference (WWW '07)*, pp. 181–190, Banff, Canada, May 2007.
- [17] K. Liu and E. Terzi, "Towards identity anonymization on graphs," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '08)*, pp. 93–106, Vancouver, Canada, June 2008.
- [18] N. Medforth and K. Wang, "Privacy risk in graph stream publishing for social network data," in *Proceedings of the 11th IEEE International Conference on Data Mining (ICDM '11)*, pp. 437–446, Vancouver, Canada, December 2011.
- [19] B. Zhou and J. Pei, "Preserving privacy in social networks against neighborhood attacks," in *Proceedings of the IEEE 24th International Conference on Data Engineering (ICDE '08)*, pp. 506–515, Cancun, Mexico, April 2008.
- [20] L. Zou, L. Chen, and M. T. Å. Ozsu, "K-automorphism: a general framework for privacy preserving network publication," in *Proceedings of the VLDB Endowment*, pp. 946–957, 2009.
- [21] J. Cheng, A. W.-C. Fu, and J. Liu, " k -isomorphism: privacy preserving network publication against structural attacks," in *Proceedings of the International Conference on Management of Data (SIGMOD '10)*, pp. 459–470, Indianapolis, Ind, USA, June 2010.
- [22] W. Wu, Y. Xiao, W. Wang, Z. He, and Z. Wang, "K-symmetry model for identity anonymization in social networks," in *Proceedings of the 13th International Conference on Extending Database Technology: Advances in Database Technology (EDBT '10)*, pp. 111–122, Lausanne, Switzerland, March 2010.
- [23] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *Proceedings of the 30th IEEE Symposium on Security and Privacy*, pp. 173–187, Washington, DC, USA, May 2009.
- [24] M. Rattigan, "Reidentification of artists and genres in KDD Cup 2011," Technical Report UM-CS-2011-021, University of Massachusetts Amherst, 2011.
- [25] A. Narayanan, E. Shi, and B. I. P. Rubinstein, "Link prediction by de-anonymization: how we won the Kaggle Social Network Challenge," in *Proceedings of the International Joint Conference on Neural Network (IJCNN '11)*, pp. 1825–1834, San Jose, Calif, USA, August 2011.
- [26] K. Henderson, B. Gallagher, L. Li et al., "It's who you know: Graph mining using recursive structural features," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*, pp. 663–671, San Diego, Calif, USA, August 2011.
- [27] Z. Junping, H. Ping, Y. Minghao, and Z. Chunguang, "Phase transitions of EXPSPACE-complete problems," *International Journal of Foundations of Computer Science*, vol. 21, no. 6, pp. 1073–1088, 2010.
- [28] J. Zhou, M. Yin, X. Li, and J. Wang, "Phase transitions of EXPSPACE-complete problems: a further step," *International Journal of Foundations of Computer Science*, vol. 23, no. 1, pp. 173–184, 2012.
- [29] X. Li and M. Yin, "An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure," *Advances in Engineering Software*, vol. 55, pp. 10–31, 2013.
- [30] X. Li and M. Yin, "Multi-operator based biogeography based optimization with mutation for global numerical optimization," *Computers & Mathematics with Applications*, vol. 64, no. 9, pp. 2833–2844, 2012.

- [31] X. Li, J. Wang, J. Zhou, and M. Yin, "A perturb biogeography based optimization with mutation for global numerical optimization," *Applied Mathematics and Computation*, vol. 218, no. 2, pp. 598–609, 2011.
- [32] B. Hajek, "A tutorial survey of theory and applications of simulated annealing," in *Proceedings of the 24th IEEE Conference on Decision and Control*, pp. 755–760, 1985.
- [33] Y. Xiaowei and W. Xintao, "Randomizing social networks: a spectrum preserving approach," in *Proceedings of the 8th SIAM International Conference on Data Mining, Applied Mathematics 130*, pp. 739–750, Atlanta, Ga, USA, April 2008.

Research Article

A Hyperbolic Tangent Adaptive PID + LQR Control Applied to a Step-Down Converter Using Poles Placement Design Implemented in FPGA

**Marcelo Dias Pedroso, Claudinor Bitencourt Nascimento,
Angelo Marcelo Tusset, and Maurício dos Santos Kaster**

Federal University of Technology-Paraná, Avenida Monteiro Lobato km 4, s/n, 84016-210 Ponta Grossa, PR, Brazil

Correspondence should be addressed to Angelo Marcelo Tusset; a.m.tusset@gmail.com

Received 28 June 2013; Revised 24 September 2013; Accepted 4 October 2013

Academic Editor: Zhiqiang Ma

Copyright © 2013 Marcelo Dias Pedroso 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 work presents an adaptive control that integrates two linear control strategies applied to a step-down converter: Proportional Integral Derivative (PID) and Linear Quadratic Regulator (LQR) controls. Considering the converter open loop transfer function and using the poles placement technique, the designs of the two controllers are set so that the operating point of the closed loop system presents the same natural frequency. With poles placement design, the overshoot problems of the LQR controller are avoided. To achieve the best performance of each controller, a hyperbolic tangent weight function is applied. The limits of the hyperbolic tangent function are defined based on the system error range. Simulation results using the Altera DSP Builder software in a MATLAB/SIMULINK environment of the proposed control schemes are presented.

1. Introduction

The technological evolution of electronic devices has been very significant in recent years. With the increasing performance of microcontrollers and Digital Signal Processors (DSP), as well as the ascension of Field-Programmable Gate Array (FPGA), associated with high speeds of current A/D converters, some concepts related to digital signal processing have been reevaluated, and new forms of mathematical processing and algorithms developed [1, 2].

More powerful digital devices are needed to enable the implementation of complex control systems in several applications to help improve their performance, stability, and robustness. In the power electronics field, better control strategies can enhance power quality and efficiency [3], by reducing losses, which represents important goals in forthcoming appliances that must comply with government environmental policies for electric power generation.

FPGAs are devices with flexible logic programming. They allow several product features and functions, adapt to new standards, and reconfigure hardware for specific applications,

even after the device has been already installed in the end product. One can use a FPGA to implement any logical function that an application-specific integrated circuit (ASIC) could perform, but the ability to update the functionality after shipping provides advantages for many application areas such as automotive, broadcast, computer and storage, display, medical, military, test and measurement, wireless, wire line, and in a lot of other areas. The main advantage of FPGA is the ability to have its logical structures arranged to work in parallel as opposed to the inherent sequential execution in microcontrollers. This can drastically boost the overall performance of the device. Also, when compared to microcontrollers and DSPs, FPGAs offer many design advantages including rapid prototyping, shorter time to market, the ability to reprogram in the field, lower Nonrecurring Engineering (NRE) costs, and long product life cycle to mitigate obsolescence risk [4–6]. In the last years, FPGAs manufacturing costs decreased significantly, making possible its use in most common applications like power electronics applications [7–11]. In these applications, the use of FPGAs allows more complex control techniques that can be used to

improve the system performance such as response time along with reduced overshoot.

FPGAs can be programmed using Hardware Development Language (HDL) which describes the connections of the logic gates together to form adders, multipliers, registers, and so on. The HDL places a dedicated resource for the task and allows for parallel operation. However, the complexity of HDL coding can be a barrier for many electrical engineering applications [12].

As an alternative to low-level HDL programming, there are some optimized frameworks with ready-to-use high-level blocks such as the Altera's DSP Builder software, which provides graphical design blocks to run simulations into a MATLAB/SIMULINK environment. The same blocks are used by DSP Builder as the basis for autogenerating an optimized HDL code to be loaded into FPGA hardware. The use of DSP builder tool can reduce the implementation time of a project resulting in lower costs associated to human-related design efforts.

In electrical power conditioning systems that use switched static converters, the integration of high performance hardware with linear and nonlinear control techniques implemented digitally has provided the improvement of system performance, resulting in increased efficiency as well as enhanced quality of power supply [3]. Several types of voltage and current controllers can be implemented, either in the analog or digital way.

Regarding the control techniques employed in these applications, traditionally the Proportional-Integral (PI) controller is one of the mostly used techniques to carry out control of some aspects of the converters, like regulating the output voltage or correcting the power factor. In these applications, the derivative action is usually not considered because it can amplify high frequency noise caused by switches commutation. Due to the integral component, PID has been widely used because it presents a favorable characteristic of eliminating the steady-state error, but its response time is high when compared to other kinds of controllers when the system is subjected to external disturbances. Other linear control techniques are also widely used like, for example, the LQR control. As it is an optimal control, it is common to appear as significant overshoots. It is desired to improve the system response time without worrying about the overshoot, commonly resulted from load disturbance or reference change, which can be achieved by employing poles placement design. However, LQR control does not have integral action, and its use does not guarantee null steady-state error.

In order to improve system performance in relation to the response time without causing overshoots, which could damage the load or circuit elements, and to obtain a steady-state null error, adaptive and nonlinear controllers have been used. The Feedback Linearization, Adaptive Control, Gain Scheduling, Sliding Mode control, and State Dependent Riccati Equation (SDRE), can be highlighted each with its own characteristics regarding their application or parameters such as stability, robustness, and computational cost [13–24]. Another possibility of enhancing the system performance is the integration of two or more controllers. In this sense, the system error is compensated according to the best

independent performance of each controller. The combined action of the two controls is considered as an adaptive control [24, 25]. A common approach is the use of a decision function that establishes a limit value that supports the decision of which control will be used. Another approach is the use of both control actions in a weighted fashion, determined by a weight function.

Along with the advances in digital control, the use of more complex theories, mainly related to nonlinear and adaptive controllers design, has been excelling ever more. Adaptive control techniques have been highlighted in various applications. Alternative forms of adaptive control techniques applied to power electronic circuits are presented in [18–24].

In this context, this paper presents an adaptive control that uses the best responses of two linear controllers, that is, the PID and LQR controls. To reduce overshoot problems of the LQR controller, poles placement design is applied. As a result, a composite control law is obtained for two different controllers used at the same time, where the control actions are mathematically weighted according to system error by means of a hyperbolic tangent function. This function represents the decision function that establishes the weights of PID and LQR control actions into the resulting control action, referred to as Hyperbolic Tangent Adaptive Control (HTAC). The main objective of this control technique is to perform the output voltage control of a step-down converter operating in continuous conduction mode (CCM). This strategy was tested by means of simulations using the DSP Builder software in a MATLAB/SIMULINK environment applied to the output voltage control on a classic Buck converter. Load steps were applied to assess the performance of the Hyperbolic Tangent Adaptive Control (HTAC). The results show that the proposed technique achieves better responses than the controllers alone, with a fast transient response, a small overshoot, due to the poles placement design, and null steady-state error.

2. Control Strategies

2.1. Linear PID Control. Among the control structures used in the industrial segment, the classic parallel PID controller, shown in Figure 1, stands out as one of the most widely used controllers due to well established practical implementation and tuning. There are various consolidated techniques which uses the PID transfer function and the system transfer function in order to obtain the proportional (K_{plin}), integral (K_{iilin}), and derivative (k_{dlin}) gains of the controller [26]. The design criteria, like overshoot and settling time for the closed loop system, are generally satisfactory when using this linear PID controller structure.

The transfer function and the control law in the time domain of a PID controller with fixed gains are expressed by

$$C(s) = \frac{K_{\text{dlin}}s^2 + K_{\text{plin}}s + K_{\text{iilin}}}{s}, \quad (1)$$

where k_{dlin} represents the derivative linear gain, k_{plin} the proportional linear gain, and k_{iilin} the integral linear gain.

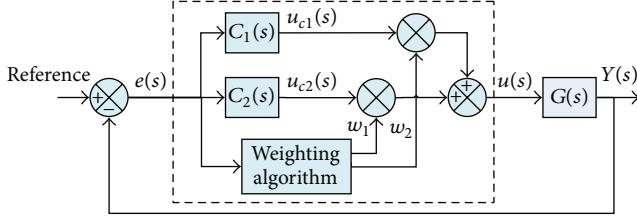


FIGURE 1: Hyperbolic tangent adaptive control.

2.2. LQR Control. The LQR controller is based on the minimization of a quadratic criterion associated with the state variables energy and the control signals. An optimal control provides a systematic way to calculate the gain matrix of the state feedback control [27, 28].

As LQR is a consolidated control technique, the controller design will not be presented in this work.

The problem formulation of optimal control (LQR) for the Buck converter can be ordered as follows: find a control function $u(t)$ ensuring that the output voltage of the converter is independent from the initial state to the reference value. The objective is to minimize the functional

$$J = \int_0^{\infty} (\delta^T Q \delta + u^T R u) dt, \quad (2)$$

where Q and R are symmetric and positive definite matrices, δ is a vector of the input errors, and u is the control signal. In addition, R can be chosen unitary without loss of generality [28].

From the control law presented in (3), for $R = 1$,

$$u(t) = -Kx(t), \quad (3)$$

where

$$K = -B^T P. \quad (4)$$

For the defined values of the matrix Q (where Q is a positive definite matrix), the matrix P (where P is a symmetric matrix) can be found, which is a solution for the Riccati equation:

$$A^T P + PA - PBB^T P + Q = 0. \quad (5)$$

The limitation is that the LQR addresses a regulation problem and cannot originally be applied to a tracking problem, which is desired in practice [27].

It is important to note that different values for the weight coefficient matrix Q are obtained for different trajectories, which implies that the range of values of Q matrix components influence the quality of the transient process.

The technique of pole placement is proposed in this paper to find an optimal matrix Q (LQR formulation) which ensures the desired characteristics of transient response, ensuring achieving optimal regulation.

To design the LQR controller feedback vector, the technique of pole placement is initially used [27, 28], since this control law works in the same way as the control law for the

LQR controller. Thus, the state feedback K is found satisfying all conditions imposed by LQR controller, and the operating point of the closed loop system can be assured.

Therefore, the vector K can be defined, by the pole placement technique, by equation

$$K_p = [\alpha_n - \beta_n \ \alpha_{n-1} - \beta_{n-1} \cdots \alpha_2 - \beta_2 \ \alpha_1 - \beta_1] T^{-1}, \quad (6)$$

where α_i is obtained from the characteristic polynomial:

$$\begin{aligned} & (s - \mu_1)(s - \mu_2) \cdots (s - \mu_n) \\ &= s^n + \alpha_1 s^{n-1} + \cdots + \alpha_{n-1} s + \alpha_n, \end{aligned} \quad (7)$$

where μ_i are the desired poles and the elements β_i are the characteristic polynomial coefficients given by

$$|sI - A| = s^n + \beta_1 s^{n-1} + \cdots + \beta_{n-1} s + \beta_n. \quad (8)$$

The transformation matrix T is given by

$$T = MW, \quad (9)$$

where M is the controllability matrix given by (10) and W is given by (11).

Consider

$$M = [B \ AB \ \cdots \ A^{n-1}B], \quad (10)$$

$$W = \begin{bmatrix} a_{n-1} & a_{n-2} & \cdots & a_1 & 1 \\ a_{n-2} & a_{n-3} & \cdots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_1 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}. \quad (11)$$

Replacing (6) in (4), the Riccati matrix (P) for gain vector (K_p) can be obtained by

$$P_p = (B^T)^+ K_p, \quad (12)$$

where $(B^T)^+$ is a pseudoinverse matrix.

Replacing (12) in (5) yields

$$Q = -A^T P_p - P_p A + P_p B B^T P_p. \quad (13)$$

Through the solution of (13), the control that causes the system to the desired orbits minimizing the functional (2) can be defined, with the predefined transient behavior, where, by means of mathematical manipulations, one can find a matrix Q that satisfies the conditions of the LQR controller. The matrix Q must be positive. In this sense, a single test can be carried through an algorithm to prove that the obtained matrix Q is equal to Q^T . Thus, it can be said that the (K_p) state feedback vector corresponds to an optimal controller offered by LQR algorithm.

2.3. Proposed Hyperbolic Tangent Adaptive Control (HTAC).

After some analyses realized by numeric simulations, to be presented afterwards, and relating to the system response during a load disturbance and to the steady-state error, it is possible to observe that either the proposed LQR and PID controllers are effective to maintain the system over the dominant poles in closed loop defined in the control design. Also, it can be observed that the LQR control is more effective for the transient response and the PID control for the steady state.

With the objective of obtaining a control that presents the combined efficiency of both controllers, enhancing the performance of the system in closed loop as well as reducing the overshoot and the settling time, a parallel combination of LQR and PID controls is proposed. The simplest approach is to switch which controller will actuate over the system given a specific rule, such as a predefined error value. The disadvantage of this approach is the occurrence of an abrupt change in the control structure. In order to avoid this abrupt commutation between the controllers, a weighted combination is proposed where their control actions are regulated by weights (w_i) defined by a hyperbolic tangent function.

The control law for the HTAC controller is defined by

$$u(t) = u_{C1(s)}(t) * w_1 + u_{C2(s)}(t) * w_2, \quad (14)$$

where $u_{C1(s)}(t)$ and w_1 represent the LQR control and its weight, respectively, and $u_{C2(s)}(t)$ and w_2 represent the PID control and its weight, respectively, with $w_1 + w_2 = 1$.

The value w_1 is determined as a function of the system error, given by

$$w_1 = \tanh |e|, \quad (15)$$

where e is the normalized error.

In this sense, it is possible to define the higher weight w_1 for the controller with faster responses in situations of larger errors, this will be the controller with higher predominance in the control action.

The value w_2 is defined by

$$w_2 = 1 - w_1. \quad (16)$$

So, the weight w_2 will be inversely proportional to the error and will present higher values in situations of small errors, bringing into spot the controller that ensures a null error in steady state. Considering this objective, a controller with an integrative term is a good choice.

In Figure 1, one can observe how the HTAC controller can be implemented.

3. Control Strategy Applied to a Step-Down Converter

Figure 2 presents the control strategy scheme applied to the step-down converter.

The design parameters for the converter are presented in Table 1.

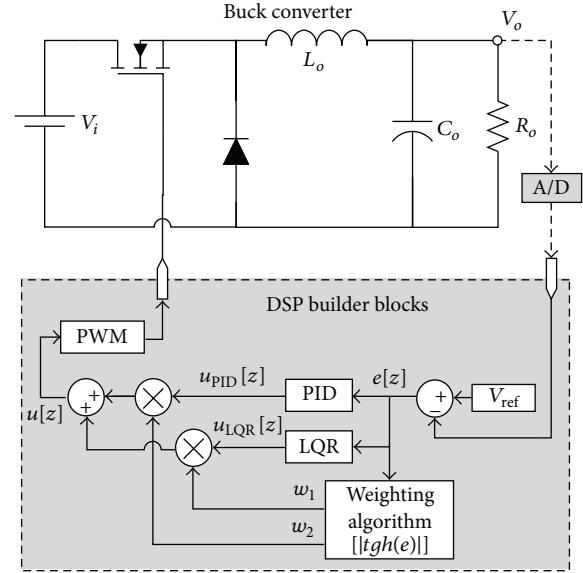


FIGURE 2: Control strategy scheme applied to the step-down converter.

TABLE 1: Parameters for the buck converter.

Parameters	Value
Input voltage (V_i)	48 V
Capacitance (C_o)	$3.33 \mu\text{F}$
Inductance (L_o)	1.1 mH
Resistance (R_o)	30 Ω
Reference voltage (V_{ref})	30 V

The matrices that determine the state space system of the buck converter are presented in the following equation:

$$A_1 = A_2 = \begin{bmatrix} 0 & \frac{1}{C_o} \\ -\frac{1}{L_o} & \frac{1}{C_o R_o} \end{bmatrix}; \quad (17)$$

$$B_1 = \begin{bmatrix} 0 \\ \frac{1}{L_o} \end{bmatrix}; \quad B_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix};$$

$$C_1 = C_2 = [0 \ 1], \quad D_1 = D_2 = 0.$$

As the duty cycle (D) is the ratio between the output reference voltage and the input voltage, the state matrices are rewritten as

$$A_1 = A_2 = \begin{bmatrix} 0 & 300300 \\ 909.1 & 10010 \end{bmatrix}; \quad (18)$$

$$B_1 = \begin{bmatrix} 0 \\ 909.1 \end{bmatrix}; \quad B_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix};$$

$$C_1 = C_2 = [0 \ 1], \quad D_1 = D_2 = 0.$$

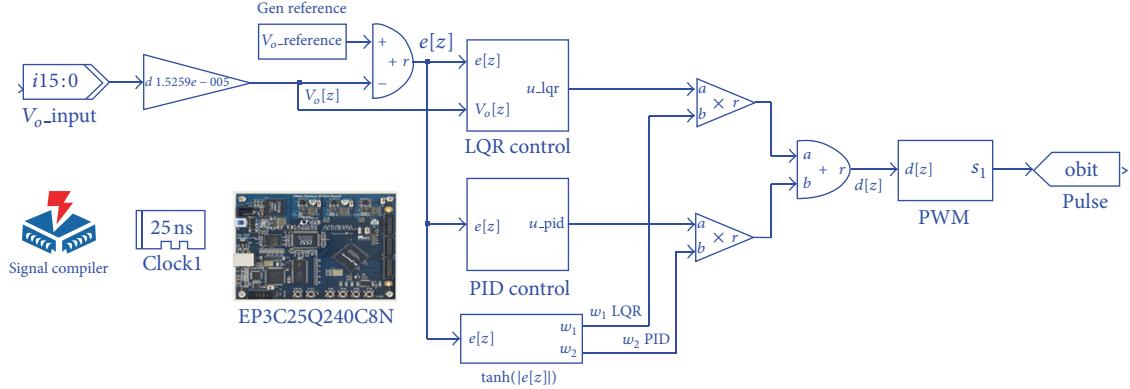


FIGURE 3: Block diagram implemented in DSP Builder.

Replacing the data in Table 1 into (17), one can define the duty cycle to the output voltage transfer function of the Buck converter, as expressed in

$$G_{v_o,d}(s) = \frac{\hat{v}_o(s)}{\hat{d}(s)} = \frac{1,28e^{10}}{s^2 + 10000s + 2.667e^7}. \quad (19)$$

Having the transfer function of the converter, the design of the controllers is set so that the operating point of the closed loop system has a natural frequency according to:

$$w_n = \frac{1}{\sqrt{L_o C_o}} = 16333 \text{ rad/s}. \quad (20)$$

The damping ratio (ξ) in open loop, determined by (20), is approximately 0.3 and is defined as a function of the settling time (t_s) at 2% and the natural frequency of the system. The value of t_s can be found from the step response of the open loop system:

$$\xi = \frac{4}{w_n t_s}. \quad (21)$$

From these data, the operating point of the closed loop system can be set to a new damping ratio near 0.8, which is the maximum acceptable value for second order systems. Thus, the set point is

$$s_1 = -13064 \pm 9798j. \quad (22)$$

With the control design parameters of the converter defined, one can design the PID and LQR, so that the operation of the closed loop system occurs at the same operating point. So, having the controllers following the same operating sequence, a comparative analysis of the performance for each control scheme can be carried out with more accuracy.

Following this analysis, the hyperbolic tangent function is applied in order to use the best responses of the applied controllers according to the error generated by possible disturbances.

From the operating point of the closed loop system, defined in (22), it is possible to design the PID controller. The designed controller gains are $K_p = 0.0723$, $K_i = 367.16$, and $K_d = 0.000004037$.

For the LQR controller using the operation point of the closed loop system defined in (22), the state feedback gain found from the proposed algorithm is

$$K_p = [4094 \ 16078]. \quad (23)$$

Replacing (23) into (13) results in $Q = 10^8 [0.0591 \ 0.0283]$, achieving the optimal control given in

$$u(t) = -4094v_{co} - 16078i_{co}. \quad (24)$$

Taking the previous designed results, the HTAC was implemented and presented in this paper using the combination of these two controllers. The limits of the hyperbolic tangent function are defined from the system error range.

4. Simulation Results

The hardware chosen for implementation of the proposed controllers in this project is a EP3C25Q240C8N Cyclone III FPGA, manufactured by Altera. The simulations are performed in the MATLAB/SIMULINK environment where the DSP Builder framework, provided by Altera, is installed as a toolbox, making, possible to simulate the model of the power converter itself and to export it to be compiled by the Altera Quartus II software without leaving Simulink.

To obtain more accurate results, various factors of a real prototype are taken into consideration. The simulation step size period was defined in the Simulink as 25 nanoseconds. The signal obtained from the output voltage is adjusted by a block that simulates the conditioning circuit in order to satisfy the inputs of the analog-to-digital converter (A/D). The external A/D converter is necessary because the FPGA hardware does not have internal A/D converters.

The chosen A/D converter is the AD7655, manufactured by Analog Devices, with a 16-bit resolution and sampling rate of 500 KSPS (samples per second). In the simulation, a system of blocks is used to represent the inherent delay of the digital conversion and quantization according to the sampling rate.

The complete implementation of the HTAC controller using DSP Builder is shown in Figure 3. It can be observed that the block diagram has a 16-bit input and a single output

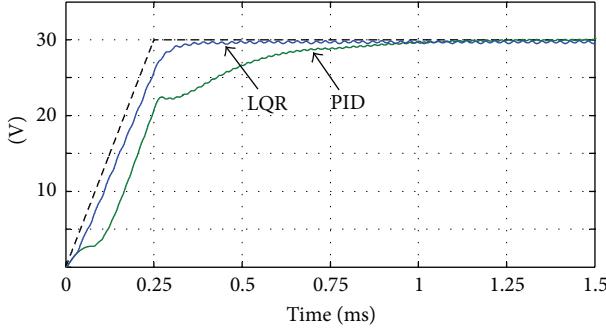


FIGURE 4: Startup process converter.

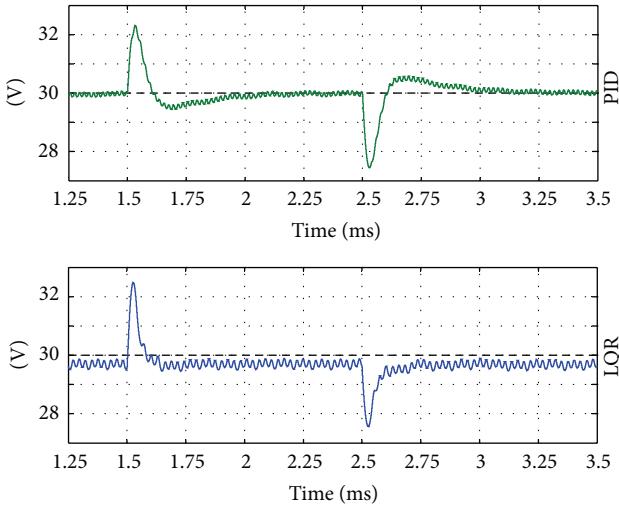


FIGURE 5: System response for a 50% load step.

(single bit) which is connected to the pulse driver that actuates the switch.

Figure 4 presents the transient response for the Buck converter in closed loop for the two control techniques employed, PID and LQR. For the converter startup process, a $250\ \mu\text{s}$ length ramp is used to achieve the rated output voltage.

Figure 5 presents the responses of each system alone for a load disturbance. Figure 6 presents the steady-state output voltage of the converter for all controllers. Unlike LQR, PID presents a zero steady-state error.

Figure 7 shows the current across the converter inductor L_o for the PID and LQR. It can be observed that the inductor L_o currents have the same ripple value defined in the converter's design. In relation to the load step response, it can be observed that for the initial overshoot the currents of PID and LQR controllers are similar.

Figure 8 presents the transient for the Buck converter in closed loop, enabling comparison between the responses of the controllers implemented separately: PID, LQR, and HTAC. Figure 9 presents in detail the responses for a load disturbance. Thus, one can observe that the HTAC presents a response as fast as that for the LQR controller and a null error in steady state as for the PID controller. Figure 10 presents the responses of HTAC for a load disturbance of 50%.

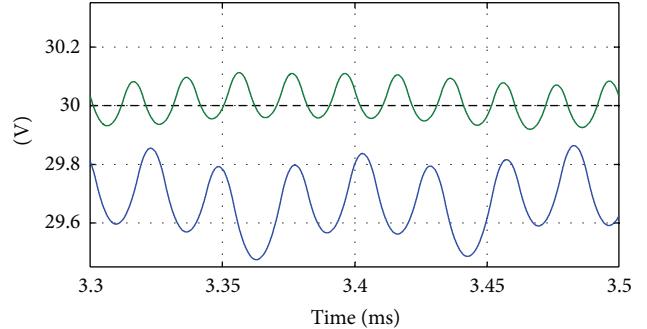


FIGURE 6: Steady-state output voltage of the converter for both controllers.

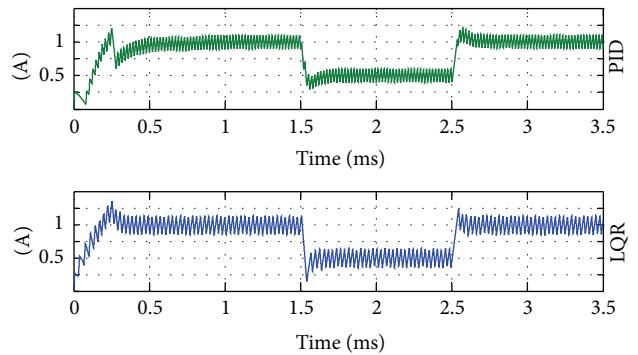
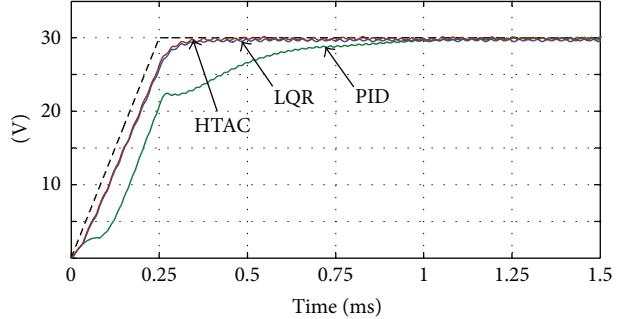
FIGURE 7: Current i_L with response for a 50% load step.

FIGURE 8: Startup process converter with HTAC control.

Figure 11 shows the converter inductor current L_o for the hyperbolic tangent adaptive control.

The variation on the weights w_1 and w_2 applied to the HTAC controller, multiplied by the control action of the PID and LQR controllers, is presented in Figure 12, where it is possible to observe that in the moments when disturbances occur, the weights exhibit variations that goes up to the moment of stabilization.

5. Conclusions

This work presented the design and simulations of an adaptive PID + LQR control technique applied to a step-down converter. The pole placement technique was used to guarantee that the two controllers work in the same operation

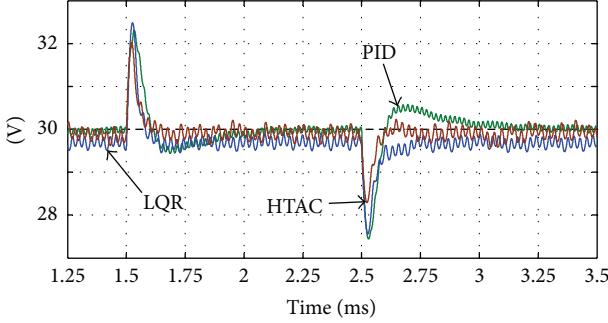


FIGURE 9: System response for a 50% load step for PID, LQR and HTAC.

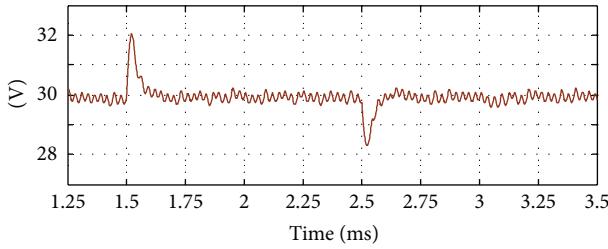


FIGURE 10: System response for a 50% load step with HTAC.

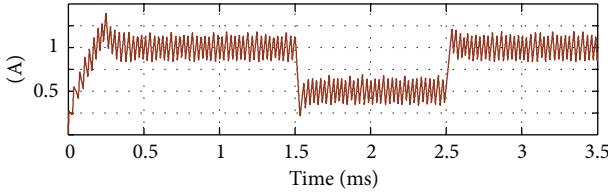


FIGURE 11: Current i_L for the hyperbolic tangent adaptive control for a 50% load step.

point and the system does not present excessive voltage and current overshoot. Knowing that the steady-state error of the converter output voltage for the PID is smaller than for the LQR control and the response time for LQR controller is smaller than for the PID, a parallel combination of the designed controllers is proposed, yielding an adaptive controller which improves the performance of the system, both in response to time and the reduction of overshoots of the controlled magnitudes.

A hyperbolic tangent weight function is used to gather the best performance of each controller according to the system error. Thus, the best responses in settling time and overshoot and annulling the steady-state error are achieved as compared to independent implementation of each controller. The Altera DSP Builder framework was used in a MATLAB/SIMULINK environment for the implementation of the Hyperbolic Tangent Adaptive Control (HTAC) and to obtain real-time simulation results.

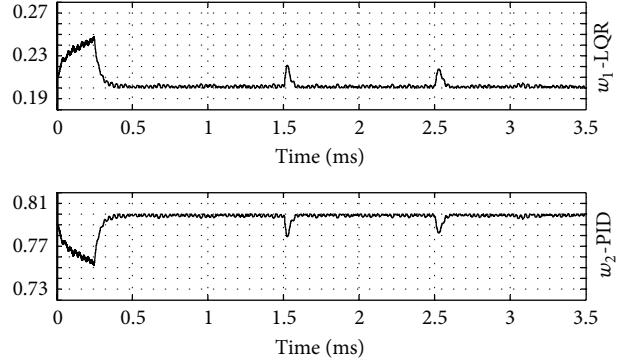


FIGURE 12: Variation of the weights w_1 and w_2 of the HTAC controller.

Acknowledgments

The authors thank FAADCTP, CNPq, and CAPES, Brazilian financing agencies.

References

- [1] P. T. Krein, “Digital control generations—digital controls for power electronics through the third generation,” in *Proceedings of the 7th International Conference on Power Electronics and Drive Systems (PEDS '07)*, pp. 1–5, Bangkok, Thailand, November 2007.
- [2] O. Jiménez, I. Urriza, L. A. Barragán, D. Navarro, J. I. Artigas, and O. Lucia, “Hardware-in-the-loop simulation of FPGA embedded processor based controls for power electronics,” in *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE '11)*, pp. 1517–1522, Gdansk, Poland, June 2011.
- [3] Y. F. Chan, M. Moallem, and W. Wang, “Design and implementation of modular FPGA-based PID controllers,” *IEEE Transactions on Industrial Electronics*, vol. 54, no. 4, pp. 1898–1906, 2007.
- [4] E. Monmasson, L. Idkhajine, M. N. Cirstea, I. Bahri, A. Tisan, and M. W. Naouar, “FPGAs in industrial control applications,” *IEEE Transactions on Industrial Informatics*, vol. 7, no. 2, pp. 224–243, 2011.
- [5] S. K. Sahoo, G. T. R. Das, and V. Subrahmanyam, “Contributions of FPGAs to industrial drives: a review,” in *Proceedings of the UK International Conference on Information and Communication Technology in Electrical Sciences (ICTES '07)*, pp. 343–348, Chennai, India, December 2007.
- [6] E. Monmasson and M. N. Cirstea, “FPGA design methodology for industrial control systems—a review,” *IEEE Transactions on Industrial Electronics*, vol. 54, no. 4, pp. 1824–1842, 2007.
- [7] S. Karimi, P. Poure, Y. Berville, and S. Saadate, “A design methodology for power electronics digital control based on an FPGA in the loop prototyping,” in *Proceedings of the 14th IEEE International Conference on Electronics, Circuits and Systems (ICECS '07)*, pp. 701–704, Marrakech, Morocco, December 2007.
- [8] E. T. Mekonnen, J. Katcha, and M. Parker, “An FPGA-based digital control development method for power electronics,” in *Proceedings of the 38th Annual Conference on IEEE Industrial Electronics Society*, pp. 222–2226, Montreal, Canada, 2012.
- [9] E. Monmasson, L. Idkhajine, I. Bahri, M.-W. Naouar, and L. Charabi, “Design methodology and FPGA-based controllers

- for power electronics and drive applications," in *Proceedings of the 5th IEEE Conference on Industrial Electronics and Applications (ICIEA '10)*, pp. 2328–2338, Bari, Italy, June 2010.
- [10] M. Matar and R. Iravani, "FPGA implementation of the power electronic converter model for real-time simulation of electromagnetic transients," *IEEE Transactions on Power Delivery*, vol. 25, no. 2, pp. 852–860, 2010.
- [11] M. Dagbagi, L. Idkhajine, E. Monmasson, and I. Slama-Belkhodja, "FPGA implementation of Power Electronic Converter real-time model," in *Proceedings of the International Symposium on Power Electronics, Electrical Drives, Automation and Motion*, pp. 658–663, Sorrento, Italy, 2012.
- [12] E. T. Mekonnen, J. Katcha, and M. Parker, "An FPGA-based digital control development method for power electronics," in *Proceedings of the 38th Annual Conference on IEEE Industrial Electronics Society*, pp. 222–226, Montreal, Canada, 2012.
- [13] V. Q. Leu, H. H. Choi, and J.-W. Jung, "Fuzzy sliding mode speed controller for PM synchronous motors with a load torque observer," *IEEE Transactions on Power Electronics*, vol. 27, no. 3, pp. 1530–1539, 2012.
- [14] Y. Zhao and W. Qiao, "A third-order sliding-mode controller for DC/DC converters with constant power loads," in *Proceedings of the 46th IEEE Industry Applications Society Annual Meeting (IAS '11)*, pp. 1–8, Orlando, Fla, USA, October 2011.
- [15] S. Qiang, Z. Chunming, H. Xiping, and L. Shouzhi, "Application of sliding mode control for half-bridge DC/DC converter," in *Proceedings of the International Conference on E-Product E-Service and E-Entertainment (ICEEE '10)*, pp. 1–4, Henan, China, November 2010.
- [16] T. Fujimoto, F. Tabuchi, and T. Yokoyama, "A design of FPGA based hardware controller for DC-DC converter using SDRE approach," in *Proceedings of the International Power Electronics Conference*, pp. 1001–1005, Sapporo, Japan, June 2010.
- [17] H. Uchida, T. Fujimoto, and T. Yokoyama, "SDRE control of single phase PWM inverter using FPGA based hardware controller," in *Proceedings of the 3rd Annual IEEE Energy Conversion Congress and Exposition (ECCE '11)*, pp. 3708–3713, Phoenix, Ariz, USA, September 2011.
- [18] V. P. Arikatla and J. A. Abu Qahouq, "Adaptive digital proportional-integral-derivative controller for power converters," *IET Power Electronics*, vol. 5, no. 3, pp. 341–348, 2012.
- [19] J. Nan, T. Hou-jun, and C. Guang-zhao, "Adaptive neuron PID control of buck type AC chopper voltage regulator," in *Proceedings of the 4th International Conference on Bio-Inspired Computing: Theories and Applications*, pp. 1–4, October 2009.
- [20] W. Huang, J. A. A. Qahouq, and S. Ahmed, "Linearized sensorless adaptive voltage positioning controller for DC-DC boost power converter," in *Proceedings of the Energy Conversion Congress and Exposition*, pp. 351–357, Raleigh, NC, USA, 2012.
- [21] H. V. Luu, A. Punzet, V. Müller, and N. L. Phung, "Control of front-end Converter with Shunt active Filter using adaptive gain," in *Proceedings of the European Conference on Power Electronics and Applications*, pp. 1–10, Dresden, Germany, September 2005.
- [22] S.-C. Lin and C.-C. Tsai, "Adaptive voltage regulation of PWM buck DC-DC converters using backstepping sliding mode control," in *Proceedings of the IEEE International Conference on Control Applications*, pp. 1382–1387, Taipei, Taiwan, September 2004.
- [23] H. Unbehauen, "Adaptive dual control systems: a survey," in *Proceedings of the Adaptive Systems for Signal Processing*, Communications, and Control Symposium, pp. 171–180, Lake Louise, Canada, 2000.
- [24] M. Algreer, M. Armstrong, and D. Giaouris, "Adaptive PD+I control of a switch-mode DC-DC power converter using a recursive FIR predictor," *IEEE Transactions on Industry Applications*, vol. 47, no. 5, pp. 2135–2144, 2011.
- [25] A. Diordiev, O. Ursaru, M. Lucanu, and L. Tigaeru, "A hybrid PID-fuzzy controller for dc/dc converters," in *Proceedings of the International Symposium on Signals, Circuits and Systems*, pp. 97–100, Iasi, Romania, 2003.
- [26] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 559–576, 2005.
- [27] A. Hasanzadeh, C. S. Edrington, and H. Mokhtari, "Optimal tuning of linear controllers for power electronics/power systems applications," *Electric Power Systems Research*, vol. 81, no. 12, pp. 2188–2197, 2011.
- [28] M. B. Poodeh, S. Eshtehardiha, A. Kiyomarsi, and M. Ataei, "Optimizing LQR and pole placement to control buck converter by genetic algorithm," in *Proceedings of the International Conference on Control, Automation and Systems (ICCAS '07)*, pp. 2195–2200, Seoul, South Korea, October 2007.

Research Article

New Algorithms for Bidirectional Singleton Arc Consistency

**Yonggang Zhang,^{1,2} Qian Yin,³ Xingjun Zhu,^{1,2} Zhanshan Li,^{1,2}
Sibo Zhang,^{1,2} and Quan Liu^{2,4}**

¹College of Computer Science and Technology, Jilin University, Changchun 130012, China

²Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education,
Jilin University, Changchun 130012, China

³College of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, China

⁴School of Computer Science and Technology, Soochow University, Jiangsu 215006, China

Correspondence should be addressed to Yonggang Zhang; zhangyg@jlu.edu.cn

Received 23 June 2013; Accepted 4 September 2013

Academic Editor: William Guo

Copyright © 2013 Yonggang Zhang 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.

Bidirectional singleton arc consistency (BiSAC) which is an extended singleton arc consistency (SAC) has been proposed recently. The first contribution of this paper is to propose and prove two theorems of BiSAC theoretically (one is a property of BiSAC and the other is the property of allowing the deletion of some BiSAC-inconsistent values). Secondly, based on these properties we present two algorithms, denoted by BiSAC-DF and BiSAC-DP, to enforce BiSAC. Also, we prove their correctness and analyze the space and time complexity of them in detail. Besides, for special circumstances, we show that BiSAC-DF admits a worst-case time complexity in $O(en^2d^4)$ and a best one in $O(en^2d^3)$ when the problem is an already BiSAC, while BiSAC-DP also has the same best one when the tightness is small. Finally, experiments on a wide range of CSP instances show BiSAC-DF and BiSAC-DP are usually around one order of magnitude faster than the existing BiSAC-1. For some special instances, BiSAC-DP is about two orders of magnitude efficient.

1. Introduction

Constraint satisfaction problem (CSP) is an important researching area in artificial intelligence. Many combinatorial problems can be solved by modeling as CSP. However, it is the NP-complete task of determining whether a given CSP instance has a solution. Fortunately, consistency techniques can reduce the search space and speed up solving process by removing the inconsistent values [1]. Many notions and algorithms of consistency have been proposed to achieve these goals so far. Among these techniques, arc consistency (AC) [2] is the oldest and most well-known way of filtering domain. This is indeed very simple and natural. More importantly, it is an effective technique to solve CSP instances. Now there are many algorithms to enforce it. For example AC-3 [3], AC-4 [4], and AC-2001 [5] are all very efficient methods and they are very easy to implement. Besides, nowadays singleton arc consistency (SAC) [6] has been focused on widely. It can remove redundant values like arc consistency

and path consistency [7], but the power of pruning the inconsistent values is stronger than that of AC. In fact, it ensures that the problem is also arc consistent after assigning any value to any variable. Debruyne and Bessière proposed a straightforward algorithm SAC-1 [6], and Prosser studied SAC again [8]. Consequently, the second algorithm SAC-2 [9] has been proposed by Bartok. It is based on AC-4 and uses external data structures to avoid invoking the AC procedure to test SAC condition more times than SAC-1. Bessière and Debruyne presented the algorithms SAC-OPT [10] and SAC-SDS [11]. SAC-OPT has the optimal time complexity with a high space complexity. SAC-SDS has a better space complexity though it is not optimal in time. However, it tries to avoid redundant work. Lecoutre proposed the first depth-first greedy algorithm SAC-3 [12]. Bessière and Debruyne (2008) studied the comprehensive theoretical of SAC and proposed a new local consistency—bidirectional singleton arc consistency (BiSAC) [13] by extending the SAC. They have

also proved BiSAC is stronger than SAC and presented the first algorithm BiSAC-1 to enforce it. Zhang et al. proposed the algorithms BiSAC-DF and BiSAC-DP (Algorithms 4 and 5), both of them were the enforced algorithms of BiSAC.

In this paper, we firstly propose two properties of BiSAC and then prove their correctness. Based on them, we present two algorithms to enforce BiSAC, denoted by BiSAC-DF and BiSAC-DP. BiSAC-DF is inspired by algorithm SAC-3 proposed by Lecoutre, and it is also a depth-first greedy algorithm. The BiSAC-DP also uses the previous properties as a theoretical foundation and combines domain partition and divide-and-conquer strategy. Besides, we propose operator ρ on subdomain to improve the efficiency of it. Also we prove their correctness and analyze their complexity. Finally, in our experiments on random CSPs, classical problems, and benchmarks, the results show that our algorithms perform better than the BiSAC-1.

In terms of complexity, BiSAC-DF and BiSAC-DP have a space complexity in $O(nd)$ and a worst-case time complexity in $O(en^3d^5)$, where n is the number of variables, d is the greatest domain size, and e is the number of constraints. In particular, when applied to an already bidirectional singleton arc consistent problem, BiSAC-DF admits the worst-case time complexity in $O(en^2d^4)$ and a best one in $O(en^2d^3)$. BiSAC-DP has a best-case time $O(en^2d^3)$ if the tightness of the problem is relatively small. Notice that it does not mean the problem is BiSAC if the tightness of it is small. Actually, the problem which is an already BiSAC may have a high tightness.

2. Preliminaries

A constraint satisfaction problem $P = (V, D, C)$, where V is a finite set of n variables, D is a finite set of domains for every variable, and C is a finite set of e constraints. For each variable $X \in V$, its domain is denoted by $D(X) \in D$. m will denote the size of the greatest domain. We can use the label D^P to represent the domain set of the problem P . Each $c_i \in C$ has an associated relation denoted by $\text{rel}^P(c_i)$ which represents the set of tuples allowed for the variables involved in c_i . We can say the tightness of P is small if $|\text{rel}^P(c_i)|$ is large enough for $\forall c_i \in C$, respectively. Problem Q is a subproblem of P if $\forall X \in V$, $D^Q \subseteq D^P$ and $\forall c_i \in C$, $\text{rel}^Q(c_i) \subseteq \text{rel}^P(c_i)$. A solution of a constraint satisfaction problem is an assignment of value from its domain to each variable such that every constraint is satisfied. A problem is said to be satisfiable if it admits at least one solution. We denote $P = \perp$ if there is a variable X of P and $D^P(X)$ wipes out.

A constraint satisfaction problem can be characterized by properties called partial consistency. It always removes some inconsistent values and some inconsistent pairs of values. There are many methods to do this. Now, we introduce several important notions. In this paper, we focused on binary CSP.

Definition 1. The value a of variable X is arc consistent (AC) if and only if, for each variable Y connected by the constraint c , there exists a value b of Y such that $(a, b) \in \text{rel}^P(c)$. The CSP

is arc consistent if and only if every value of every variable is arc consistent.

We use $\text{AC}(P)$ to represent the problem which is obtained after enforcing AC on a given problem P . If there is a variable with an empty domain in $\text{AC}(P)$, denote $\text{AC}(P) = \perp$.

Definition 2. The value a of variable X is singleton arc consistent (SAC) if and only if the problem restricted to $X = a$ is arc consistent ($\text{AC}(P|_{X=a}) \neq \perp$). The CSP is singleton arc consistent if and only if every value of every variable is singleton arc consistent.

From the definition of AC and SAC, we can clearly see SAC is stronger than AC. In 2008, Bessière extended SAC to a stronger level of local consistency and proposed a new technique—bidirectional SAC (BiSAC) [13]. Now we introduce BiSAC below.

Definition 3. The value a of variable X is bidirectional singleton arc consistent (BiSAC) if and only if $\text{AC}(T_{Xa}) \neq \perp$, $T_{Xa} = (V, D_{Xa}, C)$, where $D_{Xa}(Y) = \{b \in D^P(Y) \mid (X, a) \in \text{AC}(P|_{Y=b})\}$. The CSP is bidirectional singleton arc consistent if and only if every value of every variable is bidirectional singleton arc consistent.

In [13] Bessière proposed a straightforward algorithm BiSAC-1 to enforce bidirectional singleton arc consistency. Its correctness and its space and time complexity are $O(1)$ and $O(en^3d^5)$, respectively (see Algorithm 1).

3. New Algorithms to Enforce BiSAC

In this paper, we propose two algorithms to enforce BiSAC. Before describing them, we need to present two important theorems of BiSAC. They not only provide a theoretical basis for correctness of our algorithms, but also can improve efficiency for our algorithms.

3.1. Theoretical Analysis of BiSAC

Theorem 4. Let $P = (V, D^P, C)$ be a CSP and let Q be a subproblem of P . Given $(X, a) \in Q$,

- (1) (X, a) is bidirectional singleton arc consistent in P , if
 (X, a) is bidirectional singleton arc consistent in Q ;
- (2) (X, a) is bidirectional singleton arc consistent in P , if
 $\text{AC}(T_{Xa}) \neq \perp$, where $T'_{Xa} = (V, D'_{Xa}, C)$, $D'_{Xa}(Y) = \{b \in D^Q(Y) \mid (X, a) \in \text{AC}(P|_{Y=b})\}$.

Proof. (1) Let $T''_{Xa} = (V, D''_{Xa}, C)$, where $D''_{Xa}(Y) = \{b \in D^Q(Y) \mid (X, a) \in \text{AC}(Q|_{Y=b})\}$. Because (X, a) is BiSAC in Q , thus $\text{AC}(T''_{Xa}) \neq \perp$ is clearly concluded by the definition of BiSAC. For every $Y \in V$, the fact that Q is a subproblem of P can produce $D^Q(Y) \subseteq D^P(Y)$. Moreover, according to the correctness of AC procedure, for $(Y, b) \in Q$, if $(X, a) \in \text{AC}(Q|_{Y=b})$, then $(X, a) \in \text{AC}(P|_{Y=b})$. Above all, it implies $D''_{Xa}(Y) \subseteq D_{Xa}(Y)$, where $D_{Xa}(Y) = \{b \in D^P(Y) \mid (X, a) \in \text{AC}(P|_{Y=b})\}$. Let $T_{Xa} = (V, D_{Xa}, C)$; hence, T''_{Xa} is a

```

(1)  repeat
(2)    CHANGE  $\leftarrow$  false;
(3)    foreach  $(X, a) \in D^P$  do
(4)       $D^T \leftarrow \emptyset$ ;
(5)      foreach  $(Y, b) \in D^P$  do
(6)        if  $(X, a) \in AC(P|_{Y=b})$  then  $D^T \leftarrow D^T \cup \{(Y, b)\}$ ;
(7)        if  $AC(V, D^T, C) = \perp$  then
(8)           $D(X) \leftarrow D(X) \setminus \{a\}$ ;
(9)          if  $D(X) = \emptyset$  then return false;
(10)         CHANGE  $\leftarrow$  true;
(11) until not CHANGE;
(12) return true;

```

ALGORITHM 1: BiSAC-1.

subproblem of the problem T'_{Xa} . According to the correctness of AC procedure again, $AC(T'_{Xa}) \neq \perp$ implies $AC(T_{Xa}) \neq \perp$. So (X, a) is also bidirectional singleton arc consistent in P by the definition of BiSAC.

(2) Let $T_{Xa} = (V, D_{Xa}, C)$, where $D_{Xa}(Y) = \{b \in D^P(Y) \mid (X, a) \in AC(P|_{Y=b})\}$. Thus, T'_{Xa} is a subproblem of T_{Xa} . Now, using the correctness of AC and the fact that $AC(T'_{Xa}) \neq \perp$, $AC(T_{Xa}) \neq \perp$ is deduced correspondingly. Hence conclusion (2) is obtained by the definition of BiSAC. \square

Actually conclusion (1) exploits BiSAC of the subproblem Q to ensure that (X, a) is BiSAC in P . So it can avoid revoking the redundant AC procedure (see Definition 3). Conclusion (2) can be a similar result through a combination of definition of BiSAC and subproblem Q . It is not enough although Theorem 4 as a property of BiSAC can conclude that (X, a) is BiSAC in P . We also need another property to ensure (X, a) is bidirectional singleton arc inconsistent in P .

Theorem 5. Let $P = (V, D^P, C)$ be a CSP and $(X, a) \in P$.

- (1) (X, a) is bidirectional singleton arc inconsistent in P , if $AC(P|_{X=a}) = \perp$.
- (2) (X, a) is bidirectional singleton arc inconsistent in P , if $Q = AC(P|_{X=a}) \neq \perp$ and $AC(T'_{Xa}) = \perp$, where $T'_{Xa} = (V, D'_{Xa}, C)$, $D'_{Xa}(Y) = \{b \in D^Q(Y) \mid (X, a) \in AC(P|_{Y=b})\}$.

Proof. (1) Let $T_{Xa} = (V, D_{Xa}, C)$, where $D_{Xa}(Y) = \{b \in D^P(Y) \mid (X, a) \in AC(P|_{Y=b})\}$ and $R = P|_{X=a}$. It means that $D_{Xa}(X) = D^R(X) = \{a\}$ and $D_{Xa}(Y) \subseteq D^P(Y) = D^R(Y)$ for every $Y \in V$ s.t. $Y \neq X$. Thus, according to the correctness of AC, $AC(R) = AC(P|_{X=a}) = \perp$ can lead to $AC(T_{Xa}) = \perp$. Therefore, (X, a) is bidirectional singleton arc inconsistent in P by the definition of BiSAC.

(2) Let $S = \{(Z, c) \mid (Z, c) \in R = P|_{X=a} \wedge (Z, c) \notin Q = AC(P|_{X=a})\}$; it means that every value in S is arc inconsistent when the problem P is restricted to $X = a$. Besides, let $T_{Xa} = (V, D_{Xa}, C)$ where, $D_{Xa}(Y) = \{b \in D^P(Y) \mid (X, a) \in AC(P|_{Y=b})\}$. It implies $D_{Xa}(X) = D'_{Xa}(X) = \{a\}$ and $D'_{Xa}(Y) \subseteq D_{Xa}(Y)$ for every Y in V and $Y \neq X$. Thus every element in $D_{Xa} \setminus D'_{Xa}$ must be in $D^R \setminus D^Q$, so $D_{Xa} \setminus D'_{Xa} \subseteq S$ and

T'_{Xa} is a subproblem of T_{Xa} . However, every value in $D_{Xa} \setminus D'_{Xa}$ is arc inconsistent in T_{Xa} by the correctness of AC. Again, this correctness and $AC(T'_{Xa}) = \perp$ will lead to $AC(T_{Xa}) = \perp$. Thus (X, a) is bidirectional singleton arc inconsistent in P . So, conclusion (2) is obtained. \square

Theorem 5 is a property of bidirectional singleton arc inconsistency. Conclusion (1) of it, using the fact that problem P restricted to $X = a$ is arc inconsistent, can infer (X, a) is bidirectional singleton arc inconsistent in P . Another conclusion also can tell us (X, a) is bidirectional singleton arc inconsistent in P through arc inconsistency of a special subproblem. Actually, these theorems together provide a theoretical basis for soundness and completeness of our algorithms. More importantly, they are also approaches to speed up the efficiency of consistency technique.

Now, we propose two algorithms to enforce BiSAC, denoted by BiSAC-DF and BiSAC-DP, which rely on Theorems 4 and 5.

3.2. BiSAC-DF Algorithm. The first algorithm BiSAC-DF is inspired by SAC-3, which is also depth-first greedy approach. Moreover the above theorems can improve efficiency and ensure correctness of BiSAC-DF. The function BiSAC-DF firstly puts all variable-value pairs in the structure *Queue* and then calls the function Judge-DF to establish BiSAC. If the Judge-DF returns \perp , it means there exists a domain that wipes out. So the BiSAC returns false. Otherwise, this processing continues until a fix-point is reached.

The function Judge-DF performs a depth-first search in order to instantiate variables, but it does not have a backtrack mechanism when dead ends occur. As long as, the current branch, no inconsistency is found, we try to extend it. In other words, the current problem BP is restricted to $X = a$ and then arc consistency is maintained. So a subproblem $AC(BP|_{X=a})$ is generated correspondingly. The other value pairs will be judged whether they are BiSAC on this current subproblem. Theorem 4 concludes that (X, a) is BiSAC in P if it is BiSAC in sub-problem or $AC(BP|_{X=a}) \neq \perp$ (line 15 and line 25). In particular, it maybe finds a lucky solution (line 18). For bidirectional singleton arc inconsistency, a dead end must occur. In that case, Theorem 5 can tell us that (X, a) is bidirectional singleton arc inconsistent in P if the structure

occur is null (line 6 and line 14). Therefore, the value a should be removed from the domain of the variable X . If this domain wipes out, the algorithm terminates and returns \perp (line 23). Otherwise, we cannot judge whether (X, a) is BiSAC or not. So, it needs further propagations.

Theorem 6. *BiSAC-DF is a correct algorithm to enforce bidirectional singleton arc consistency.*

Proof. Consider the following.

Soundness. Suppose (X, a) is the first BiSAC value removed by BiSAC-DF. It is necessarily removed in line 22 of the function Judge-DF. It means that the structure *occur* is \emptyset , and at this time it goes to line 22 only from line 6 or line 14. Hence, if it jumps from line 6, it implies $AC(BP|_{X=a}) = \perp$. The main loop in Judge-DF was executed only once because the structure *occur* is \emptyset , so $BP = P$. Therefore, (X, a) is bidirectional singleton arc inconsistent in P according to conclusion (1) of Theorem 5. Jumping from line 14, it is implied that $Q = AC(BP|_{X=a}) \neq \perp$ and $AC(T^Q) = \perp$; conclusion (2) of Theorem 5 admits that (X, a) is also bidirectional singleton arc inconsistent in P . Therefore, this contradicts the assumption that (X, a) is the first BiSAC value removed by BiSAC-DF. So, BiSAC-DF is sound.

Completeness. When the BiSAC-DF terminated, for every $(X, a) \in D$, (X, a) is removed from *Queue* if the clause repeat (in Judge-DF) is executed for the first time. It means $AC(BP) \neq \perp$, where $BP = (V, D^{BP}, C)$, $D^{BP}(Y) = \{b \in D^P(Y) \mid (X, a) \in AC(P|_{Y=b})\}$. According to the definition of BiSAC, (X, a) is bidirectional singleton arc consistent in P . Otherwise, it is concluded that $AC(BP) \neq \perp$, where $BP = (X, D^{BP}, C)$, $D^{BP}(Y) = \{b \in D^Q(Y) \mid (X, a) \in AC(P|_{Y=b})\}$. Theorem 4 can ensure that (X, a) is also bidirectional singleton arc consistent in P . Thus, BiSAC-DF is complete. \square

Theorem 7. *BiSAC-DF admits a space complexity in $O(nd)$ and a worst-case time complexity in $O(en^3d^5)$.*

Proof. The additional data structure in BiSAC-DF is *Queue* and *BP*, while the space of them is also $O(nd)$, so the overall space complexity of BiSAC-DF is $O(nd)$. The worst-case of running BiSAC-DF is that there is exactly one value removed from its domain when the loop (see line 5 in Algorithm 2) terminates. In that case, the number of elements in *Queue* is $(nd - i + 1)$ when the loop repeat is executed i ($1 \leq i \leq nd$) times. Notice that the number of variable-value pairs in P is also $(nd - i + 1)$. Let k be the total times of calling Judge-DF at i iterative times in the loop repeat (in BiSAC-DF), and let del_j be the number of elements removed from *Queue* at j times of calling Judge-DF. From the function Judge-DF, it is easily concluded that the times of calling AC are at most $(nd - i + 1)$ when a value is removed from *Queue*. Thus, at i iterative times in the loop repeat (in BiSAC-DF), the total number of calling AC is

$$\sum_{j=1}^{j=k} del_j \times (nd - i + 1) = (nd - i + 1)^2. \quad (1)$$

```

(1) repeat
(2)   CHANGE  $\leftarrow$  false;
(3)   foreach  $(X, a) \in P$  do
(4)     Queue  $\leftarrow$  Queue  $\cup \{(X, a)\}$ ;
(5)   while Queue  $\neq \emptyset$  do
(6)     if Judge-DF()  $= \perp$  then return false;
(7)   until not CHANGE
(8) return true;
```

ALGORITHM 2: BiSAC-DF.

Therefore, the times of AC called by BiSAC-DF are $(nd)^2 + (nd - 1)^2 + \dots + (nd - i + 1)^2 + \dots + 1^2 = 1/6 \times (nd) \times (nd + 1) \times (2nd + 1)$. If an optimal AC is used, then the time complexity is $O(ed^2)$, such as AC-2001. So, the worst-case time complexity of BiSAC-DF is $1/6 \times (nd) \times (nd + 1) \times (2nd + 1) \times O(ed^2)$, namely, $O(en^3d^5)$. \square

Theorem 8. *Applied to a constraint network which is BiSAC, the worst-case time complexity of BiSAC-DF is $O(en^2d^4)$ and the best-case time complexity is $O(en^2d^3)$.*

Proof. When it is applied to a BiSAC network, there will be no value removed. It implies that the loop repeat (in Algorithm 2) is executed only once. During this procedure, the number of elements in *Queue* is nd . For every element in *Queue*, in the worst-case, it needs call AC $2 \times nd$ times and the total is $2 \times (nd)^2$. Thus, worst-case time complexity is $2 \times (nd)^2 \times O(ed^2) = O(en^2d^4)$. For the best case, it only executes Judge-DF exactly once (finding a lucky solution). So the number of calling AC is $d(n - i) + i$ at i ($1 \leq i \leq n$) iterative times in the loop repeat (in Algorithm 3). Note that the number of elements in *BP* is $d(n - i) + i$. Thus, the total of calling AC is

$$\sum_{i=1}^{i=n} (d(n - i) + i) = n/2 \times (n(d + 1) + (1 - d)). \quad (2)$$

So the best-case time complexity is $n/2 \times (n(d + 1) + (1 - d)) \times O(ed^2) = O(en^2d^3)$. \square

3.3. BiSAC-DP Algorithm. Another algorithm BiSAC-DP is also based on the previous propositions, but it is not depth-first greedy approach. So it cannot find a lucky solution. However, it combines domain partition technique and uses the operator ρ presented by us to find a special subproblem so that it saves lots of redundant work. Now we propose operator ρ : let $P = (V, D, C)$; operator $\rho(P, X)$ must remove every value b from the domain of every variable Y connected with X by the constraint c , where the value b s.t. $\exists a \in D(X)$ such that $(a, b) \notin \text{rel}^P(c)$. During removing values, if a domain wipes out, operator ρ returns \perp ; otherwise, it returns the new problem. From the definition, we know operator ρ can run in $O(nd^2)$ at worst without additional data structure. Actually, if $|D(X)| = 1$, the function of operator ρ is the

```

(1) occur  $\leftarrow \emptyset$ ;
(2) BP  $\leftarrow P$ ;
(3) repeat
(4)     get  $(X, a)$  from Queue s.t.  $X \notin \text{vars}(\textit{occur})$  and  $(X, a) \in \textit{BP}$ ;
(5)     BP  $\leftarrow \text{AC}(\textit{BP}|_{X=a})$ ;
(6)     if BP =  $\perp$  then goto L;
(7)     flag  $\leftarrow \text{false}$ ;
(8)     foreach  $(Y, b) \in \textit{BP}$  do
(9)         if  $(X, a) \notin \text{AC}(P|_{Y=b})$  then
(10)            BP  $\leftarrow \textit{BP} \setminus \{(Y, b)\}$ 
(11)            flag  $\leftarrow \text{true}$ ;
(12)        if flag =  $\text{true}$  then
(13)            BP  $\leftarrow \text{AC}(\textit{BP}|_{X=a})$ ;
(14)        if BP =  $\perp$  then goto L;
(15)        else occur  $\leftarrow \textit{occur} \cup \{(X, a)\}$ ;
(16)    else occur  $\leftarrow \textit{occur} \cup \{(X, a)\}$ ;
(17) until  $\text{vars}(\textit{occur}) = \text{vars}(P)$ ;
(18) return Solution;
(19) L:
(20) if occur =  $\emptyset$  then
(21)     CHANGE  $\leftarrow \text{true}$ ;
(22)     P  $\leftarrow P \setminus \{(X, a)\}$ ;
(23)     if P =  $\perp$  then return  $\perp$ ;
(24) else
(25)     Queue  $\leftarrow \textit{Queue} \setminus \textit{occur}$ ;
(26)     Queue  $\leftarrow \textit{Queue} \cup \{(X, a)\}$ ;

```

ALGORITHM 3: Judge-DF.

```

(1) repeat
(2)     CHANGE  $\leftarrow \text{false}$ ;
(3)     foreach  $Y \in V$  do
(4)          $D_1 :: D_2 \leftarrow D(Y)$ ;
(5)         if  $\text{Judge-DP}(Y, D_1) = \perp \parallel \text{Judge-DP}(Y, D_2) = \perp$  then
(6)             return  $\text{false}$ ;
(7) until  $\text{not } \textit{CHANGE}$ ;
(8) return  $\text{true}$ ;

```

ALGORITHM 4: BiSAC-DP.

same as forward checking (FC). For $|D(X)| > 1$, it can be considered as the extended forward checking. Note that the operator ρ is only used to improve efficiency through finding an appropriate special subproblem. However $\rho(P, X) = \perp$ when $|D(X)| > 1$; we cannot say that the current problem has no solutions. Actually, in that case all values in $D(X)$ are also necessary to be propagated for BiSAC. So, a “good” domain $D(X)$ for $\rho(P, X) \neq \perp$ is very important for efficiency of our algorithm. Therefore, we introduce divide-conquer strategy to automatically find an appropriate subdomain.

The function BiSAC-DP partitions the domain of every variable into two subdomains firstly and then calls the function Judge-DP to establish the BiSAC. This process terminates until the fix-point is reached. It means that every value in the current subdomain D_X is bidirectional singleton arc consistent when function Judge-DP returns true. Otherwise, it means that a domain has wiped out when it returns \perp .

The main function of Judge-DP is to determine whether all values in subdomain D_X are BiSAC in the special subproblem. If they are all BiSAC, then they return true; otherwise it needs to partition the current subdomain. Firstly, Judge-DP uses operator ρ to find the special subproblem. If $\rho(Q, X) = \perp$ (line 4), the function would go to label L. Otherwise, Judge-DP calls the AC. It would still go to L if AC returns \perp too. Otherwise, it enforces BiSAC on this current problem (line 6-line 10). Theorem 4 can ensure that it is BiSAC in P if it is BiSAC in the current subproblem for every value in subdomain D_X (line 10). However, if not, it also needs to go to L for further processing. In label L, according to the definition of BiSAC, if $|D_X| = 1$, the single value in D_X is bidirectional singleton arc inconsistent, so it must be removed (line 13). However, if $|D_X| \neq 1$, the current subdomain D_X needs to be partitioned further (line 18).

```

(1)    $Q \leftarrow P;$ 
(2)    $D^Q(X) \leftarrow D_X;$ 
(3)   if  $|D^Q(X)| = 0$  then return true;
(4)   if  $\rho(Q, X) = \perp$  then goto L;
(5)   if  $AC(Q) = \perp$  then goto L;
(6)   foreach  $(Y, b) \in Q$  s.t  $Y \neq X$  do
(7)     if  $AC(P|_{Y \leftarrow b}) \neq \perp$  then
(8)       if exist  $a \in D_X$  s.t.  $(X, a) \notin AC(P|_{Y=b})$  then
(9)          $Q \leftarrow Q \setminus \{(Y, b)\};$ 
(10)    if  $AC(Q) \neq \perp$  then return true;
(11) L:
(12)   if  $|D_X| = 1$  then
(13)      $P \leftarrow P \setminus \{(X, D_X)\};$ 
(14)      $CHANGE \leftarrow \text{true};$ 
(15)     if  $P = \perp$  then return  $\perp;$ 
(16)     else return true;
(17)    $D_1 :: D_2 \leftarrow D_X;$ 
(18)   if Judge-DP( $X, D_1$ ) =  $\perp$   $\parallel$  Judge-DP( $X, D_2$ ) =  $\perp$  then return  $\perp;$ 
(19)   return true;

```

ALGORITHM 5: Judge-DP (X, D_X).

Theorem 9. BiSAC-DP is a correct algorithm to enforce bidirectional singleton arc consistency.

Proof. Consider the following.

Soundness. Suppose (X, a) is the first BiSAC value removed by BiSAC-DP. It is necessarily removed in line 13 of the function Judge-DP. It implies that the subdomain D_X only contains one value $a (D_X = \{a\})$. From the Judge-DP, it goes to line 13 only through line 4, line 5, or line 10 (in this case, $AC(Q) = \perp$). If it goes through line 5, then $AC(Q) = \perp$ and $Q = P|_{X=a}$. So conclusion (1) of Theorem 5 admits (X, a) is bidirectional singleton arc inconsistent in P . Through line 4, $AC(Q) = \perp$ and $Q = P|_{X=a}$ can also be concluded because $D_X = \{a\}$ and $\rho(Q, X) = \perp$. So the same result can be obtained. Otherwise, from line 10 and $AC(Q) = \perp$, in this case, according to conclusion (2) in Theorem 5, (X, a) is also bidirectional singleton arc inconsistent in P . Above all, this contradicts the assumption. So, BiSAC-DP is sound.

Completeness. When the BiSAC-DP terminated, for every $(X, a) \in D$, (X, a) is checked or D_X is checked for BiSAC, where D_X is subdomain of D and $(X, a) \in D_X$. If (X, a) is checked, then (X, a) is BiSAC in P by Theorem 4. Otherwise, the same result is also obtained by the definition of operator ρ and Theorem 4. Thus, BiSAC-DP is complete. \square

Theorem 10. BiSAC-DP admits a space complexity in $O(nd)$ and a time complexity in $O(\lambda end^3)$, where λ denotes the number of partitions built by the BiSAC-DP.

Proof. The BiSAC-DP has only the additional data structure Q and the space complexity of it is $O(nd)$. So the space complexity of BiSAC-DP is $O(nd)$. From the function partition, we can see that the number of calling AC is nd when the Judge-DP (from lines 1 to 10) is executed only once. So, the

time complexity of BiSAC-DP is $2 \times \lambda \times nd \times O(ed^2) = O(\lambda end^3)$. \square

Corollary 11. The worst-case time complexity of BiSAC-DP is $O(en^3d^5)$, and the best-case time complexity is $O(en^2d^3)$.

Proof. The worst-case of running BiSAC-DP is that there is exactly one value removed from its domain when the loop (see line 3 in BiSAC-DP) terminates. So it calls the function Judge-DP $2 \times n^2d$ times. The number of partitions produced by calling the function Judge-DP once is at most $2 + 2^2 + \dots + 2^k = 2^{k+1} - 2$, where $2^k = d$. Consequently, $\lambda = 2 \times n^2d \times (2^{k+1} - 2)$. Thus the worst-case time complexity is $O(en^3d^5)$. However the best case is $\lambda = n$. The best-case time complexity $O(en^2d^3)$ is obtained. \square

From the above proofs, the time complexity of the algorithm BiSAC-DP is $O(en^2d^3)$ when $\lambda = n$. In that case, the tightness of problem must be small enough. For another algorithm BiSAC-DF, it has the same time complexity if it can find a solution in an already BiSAC constraint network. Note that a problem whose tightness is small may not be a bidirectional singleton arc consistent. In other words a bidirectional singleton arc consistent problem may have a big tightness.

4. Experiments

To compare the different algorithms mentioned in this paper, we have performed experimentation with respect to random CSPs, classical problems, and benchmarks. Performances have been measured in terms of the CPU time in seconds and the number of checks performed by BiSAC. All algorithms have been implemented in C++ and embedded into the constraint solving platform “Ming Yue” [14–16] designed by

TABLE 1: Results obtained on classical problems.

Problems	BiSAC-1		BiSAC-DF		BiSAC-DP	
	Time (ms)	#chks (M)	Times (ms)	#chks (M)	Times (ms)	#chks (M)
15-queens	3813	229	1719	104	485	29
20-queens	26140	1763	11297	771	2672	174
25-queens	129031	8543	55781	3636	10750	687
15-pigeons	2672	182	1187	82	218	14
20-pigeons	20250	1491	9547	716	1109	82
25-pigeons	99828	7482	43860	3304	4141	299
$\langle 60, 6, 177 \rangle$	6234	329	2484	155	1718	102
$\langle 80, 6, 316 \rangle$	17093	1043	8938	544	5500	328
$\langle 100, 5, 495 \rangle$	26156	1514	15171	914	9717	576

TABLE 2: Results obtained on benchmarks.

Problems	BiSAC-1		BiSAC-DF		BiSAC-DP	
	Time (ms)	#chks (M)	Times (ms)	#chks (M)	Times (ms)	#chks (M)
frb30-15-1	40688	2106	25141	1294	15766	796
frb30-15-2	42235	2195	25297	1312	15782	803
Lsq_dg_7	19094	1298	5868	392	4016	273
Lsq_dg_8	64235	4712	37218	2769	12563	912
qk_8_0_5	1828	69	16	0.107	0	0.1
qk_10_0_5	9844	389	16	0.296	16	0.277
qk_12_0_5	38187	1621	47	0.664	31	0.624
composed-25_1_25_0	406	23	1516	89	375	21
composed-25_1_80_9	484	26	1766	104	422	24

us. In this platform, we use dom/deg variable ordering and choose the AC with heuristic information [17] because it is easy to implement and very efficient. The experiments were carried out on a DELL Intel PIV 3.0 GHz CPU/512 MB RAM with Windows XP Professional SP2/Visual C++6.0.

Random constraint satisfaction problems have become a standard for testing constraint satisfaction algorithms and heuristic algorithms because of their randomness [18]. A binary random constraint satisfaction problem (<http://www.lirmm.fr/~bessiere/generator.html>) is specified by four parameters $\langle n, m, p_1, p_2 \rangle$, where n defines a number of variables, m defines the size of domains, p_1 (*density*) defines how many constraints appear in the problem, and p_2 (*tightness*) defines how many pairs of values are inconsistent in a constraint. This paper chooses $m = n = 20$, while p_1 and p_2 range from 0.05 to 0.95 (interval is 0.05). Figure 1 shows the difference of running times for BiSAC-1 and BiSAC-DF and a ratio BiSAC-1/BiSAC-DF. From Figure 1(a) we can see that the average running time required by BiSAC-DF is much less than that of BiSAC-1. Actually, from the right graph of Figure 1, we can know that BiSAC-DF is around 2–5 times faster the BiSAC-1 for small *tightness*. When the *tightness* is relatively big, BiSAC-DF is about 10–25 times than BiSAC-1. For some special combination of the *density* and *tightness*, BiSAC-DF can reach about 30 times the BiSAC-1. All of them can be explained by the fact that the times of revoking AC procedure of BiSAC-DF are much less than BiSAC-1, and BiSAC-DF can determine whether (X, a) is BiSAC earlier

than BiSAC-1. Figure 2 tells us that the number of checks performed by BiSAC-DF is also much smaller than BiSAC-1. For the other algorithm BiSAC-DP, the running time is also less than BiSAC-1. In fact, it is less than BiSAC-DF too (see Figure 3). For small *tightness* BiSAC-DP is about 2–20 times the BiSAC-1 because the BiSAC-DP can find the appropriate subproblem earlier and more easily so that lots of values can be inferred BiSAC without more BiSAC checks. For the other *tightness*, BiSAC-DP is on average 40–60 times the BiSAC. BiSAC-DP is around 100 times faster than BiSAC-1 when *density* ranges from 0.5 to 0.6, and *tightness* is set to about 0.7, which is not surprising since BiSAC-DP not only uses the operator ρ to find a good subproblem successfully, but also avoids much redundant work at testing BiSAC checks. Figure 4 also shows that BiSAC-DP performs a visibly smaller number of checks than BiSAC-1.

Besides random CSPs, we have made our algorithms apply to classical problems like Queens, Pigeons, and Color problems. From Table 1, we can see BiSAC-DF and BiSAC-DP also outperform BiSAC-1 on range of CSP instances. On average, the efficiency of BiSAC-DF is about 3 times that of BiSAC-1, and BiSAC-DP is around 8–20 times faster than BiSAC-1.

Table 2, built from other instances (<http://cpai.ucc.ie/05/Benchmarks.html>), shows that our algorithms also have a better performance than BiSAC-1. Especially, BiSAC-DP and BiSAC-DP are about 1000 times more efficient than BiSAC-1 for the instances—“qk.” For frb and Lsq_dg, our

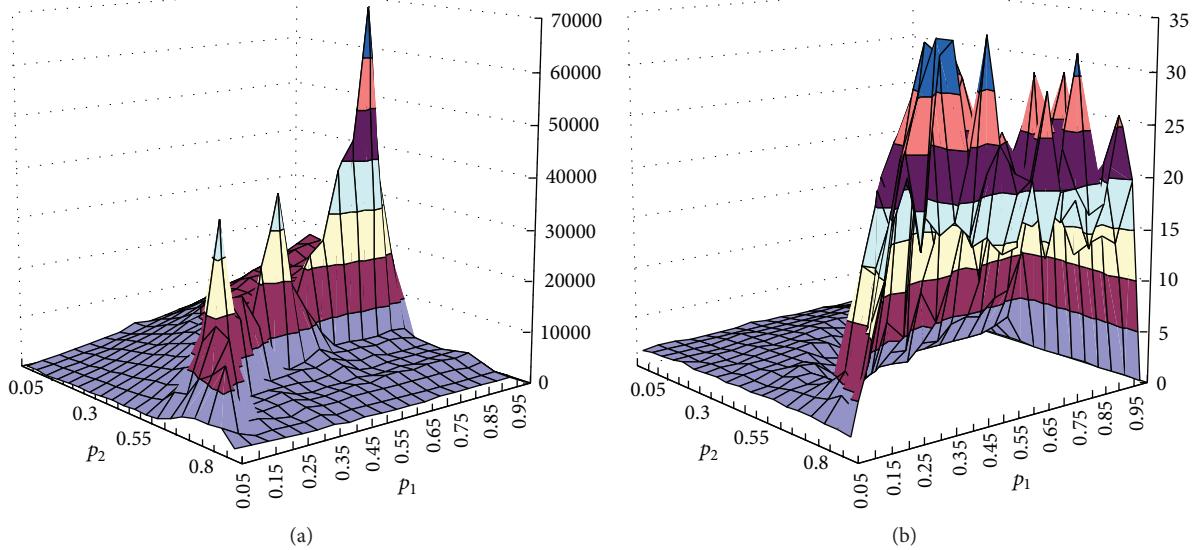


FIGURE 1: A comparison of running times for BiSAC-1 and BiSAC-DF. The left shows time difference (BiSAC-1 – BiSAC-DF), while the right shows time ratio (BiSAC-1/BiSAC-DF). Time is measured in milliseconds.

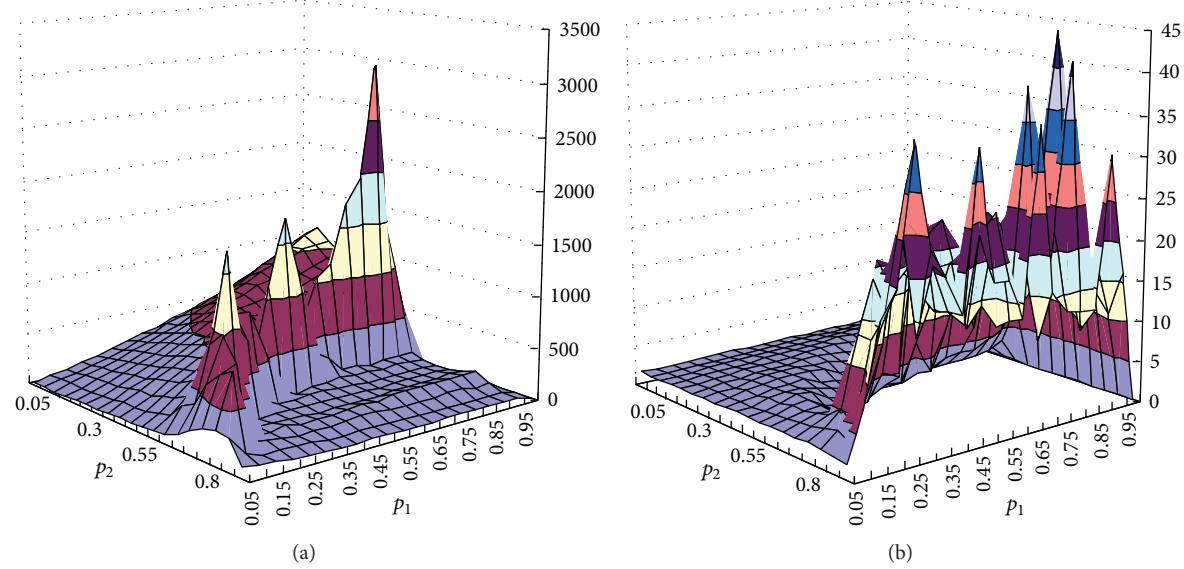


FIGURE 2: A comparison of number of checks performed by BiSAC-1 and BiSAC-DF. The left shows a relative comparison ($\text{BiSAC-1} - \text{BiSAC-DF}$), while the right shows number of checks ratio ($\text{BiSAC-1}/\text{BiSAC-DF}$). Check is measured in million.

algorithms are 2-3 times more efficient than BiSAC-1. However, BiSAC-DF is slower than BiSAC-1. Maybe this is because that the procedure of depth-first instantiating variables delays the time of finding the domain which has been wiped out.

5. Conclusion

In this paper, we made a study of bidirectional singleton arc consistency and then proposed two properties. Based on these properties, two algorithms (BiSAC-DF and BiSAC-DP) were proposed. Besides, we theoretically prove that these

algorithms have the same space complexity $O(nd)$ and worst-case time complexity $O(en^3d^5)$. For different cases, they also have the same best-case time complexity $O(en^2d^3)$. Because our algorithms do not invoke the AC procedure to test the BiSAC condition more times than the original BiSAC-1, so they avoid lots of redundant works and have a better performance. Our experimental results also show that our approaches significantly outperform BiSAC-1 on range of CSP instances, and it will be applied to solve large scale real life problems coming from the scheduling and configuration areas.

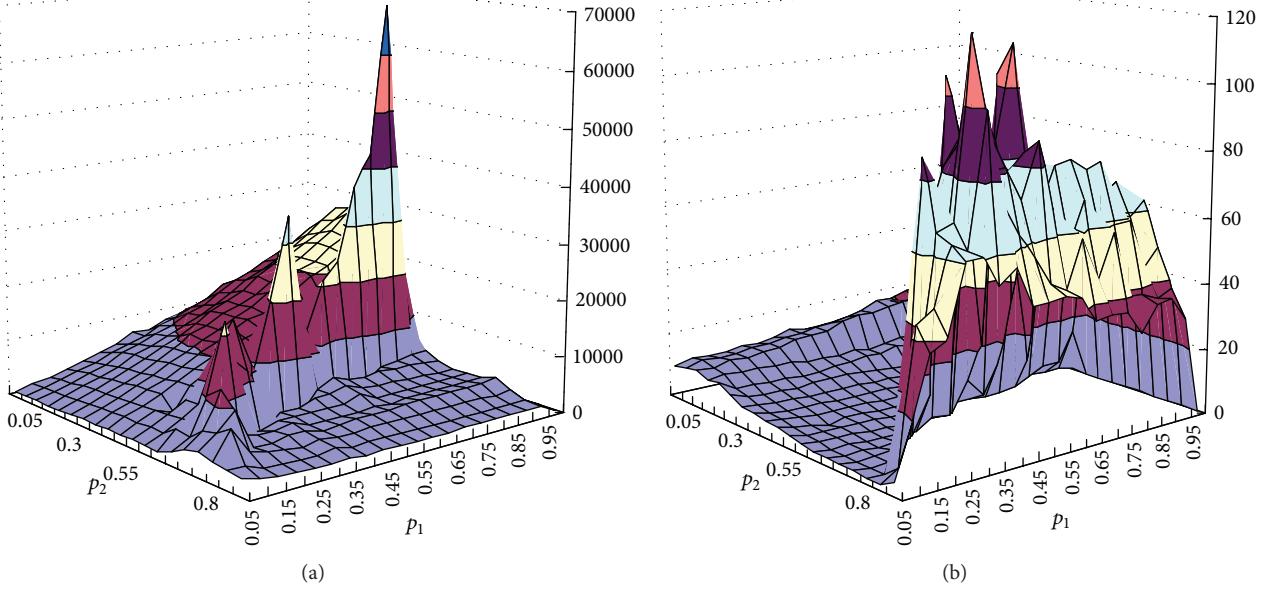


FIGURE 3: A comparison of running times for BiSAC-1 and BiSAC-DP. The left shows time difference (BiSAC-1 – BiSAC-DP), while the right shows time ratio (BiSAC-1/BiSAC-DP). Time is measured in milliseconds.

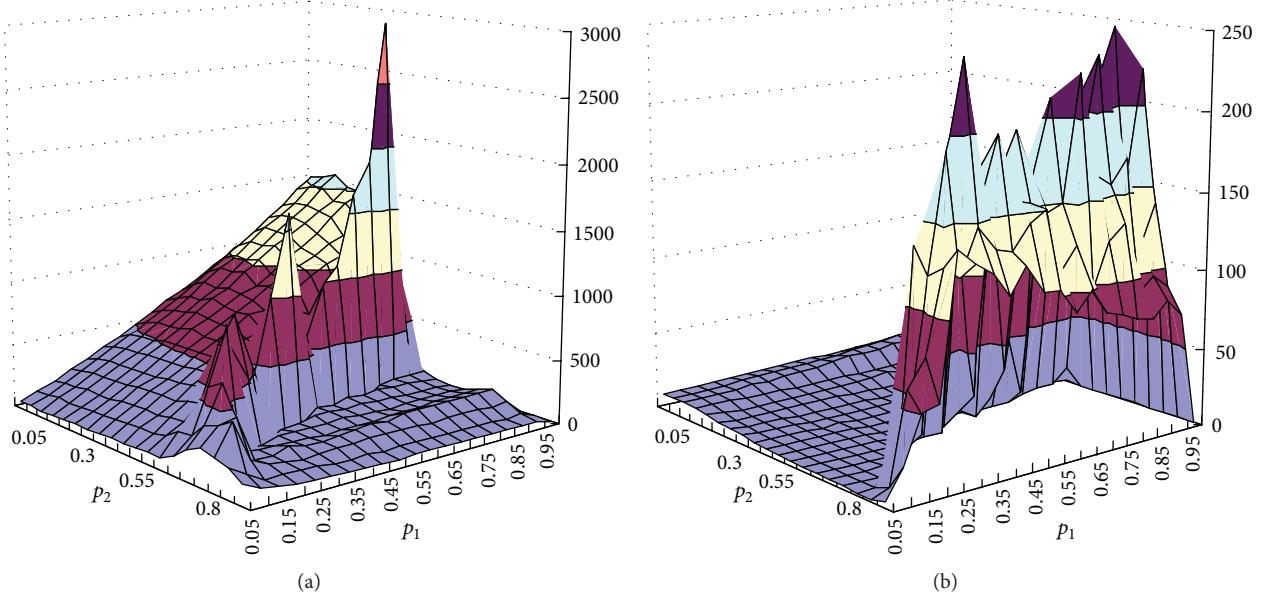


FIGURE 4: A comparison of number of checks performed by BiSAC-1 and BiSAC-DP. The left shows a relative comparison (BiSAC-1 – BiSAC-DP), while the right shows number of checks ratio (BiSAC-1/BiSAC-DP). Check is measured in million.

Acknowledgments

The authors of this paper express sincere gratitude to all the anonymous reviewers for their hard work. This work was supported in part by NSFC under Grant nos. 61170314, 61272208, and 61373052.

References

- [1] R. Barták, "Theory and practice of constraint propagation," in *Proceedings of the 3rd Workshop on Constraint Programming in*

Decision and Control, J. Figwer, Ed., pp. 7–14, Gliwice, Poland, 2001.

- [2] C. Bessière, J.-C. Régin, R. H. C. Yap, and Y. Zhang, “An optimal coarse-grained arc consistency algorithm,” *Artificial Intelligence*, vol. 165, no. 2, pp. 165–185, 2005.
 - [3] A. K. Mackworth, “Consistency in networks of relations,” *Artificial Intelligence*, vol. 8, no. 1, pp. 99–118, 1977.
 - [4] R. Mohr and T. C. Henderson, “Arc and path consistency revisited,” *Artificial Intelligence*, vol. 28, no. 2, pp. 225–233, 1986.
 - [5] C. Bessière and R. Régin, “Refining the basic constraint propagation algorithm,” in *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, 1997, pp. 319–323.

- Conference on Artificial Intelligence (IJCAI '01)*, pp. 309–315, 2001.
- [6] R. Debruyne and C. Bessière, “Some practical filtering techniques for the constraint satisfaction problem,” in *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI '97)*, pp. 412–417, 1997.
 - [7] A. Chmeiss and P. Jégou, “Efficient path-consistency propagation,” *International Journal on Artificial Intelligence Tools*, vol. 7, no. 2, pp. 121–142, 1998.
 - [8] P. Prosser, K. Stergiou, and T. Walsh, “Singleton consistencies,” in *Proceedings of the International Conference on Principles and Practice of Constraint Programming (CP '00)*, pp. 353–368, 2000.
 - [9] R. Barták, “A new algorithm for singleton arc consistency,” in *Proceedings of the Forging Links and Integrating Resources Conference (FLAIRS '04)*, pp. 257–262, 2004.
 - [10] C. Bessière and R. Debruyne, “Theoretical analysis of singleton arc consistency,” in *Proceedings of the ECAI Workshop on Modelling and Solving Problems with Constraints*, pp. 20–29, 2004.
 - [11] C. Bessière and R. Debruyne, “Optimal and suboptimal singleton arc consistency algorithms,” in *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI '05)*, pp. 54–59, 2005.
 - [12] C. Lecoutre and S. Cardon, “A greedy approach to establish singleton arc consistency,” in *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI '05)*, pp. 199–204, 2005.
 - [13] C. Bessière and R. Debruyne, “Theoretical analysis of singleton arc consistency and its extensions,” *Artificial Intelligence*, vol. 172, no. 1, pp. 29–41, 2008.
 - [14] J. G. Sun and S. Y. Jing, “Solving non-binary constraint satisfaction problem,” *Chinese Journal of Computers*, vol. 26, no. 12, pp. 1746–1752, 2003.
 - [15] J.-G. Sun, X.-J. Zhu, Y.-G. Zhang, and Y. Li, “An approach of solving constraint satisfaction problem based on preprocessing,” *Chinese Journal of Computers*, vol. 31, no. 6, pp. 919–926, 2008.
 - [16] X.-J. Zhu, J.-G. Sun, Y.-G. Zhang, and Y. Li, “A new preprocessing technique based on entirety singleton consistency,” *Acta Automatica Sinica*, vol. 35, no. 1, pp. 71–76, 2009.
 - [17] J. G. Sun, X. J. Zhu, Y. G. Zhang, and J. Gao, “Fail first principle for constraint propagation,” *Chinese Journal of Mini-Micro Systems*, vol. 29, no. 4, pp. 678–681, 2008.
 - [18] I. P. Gent, E. Macintyre, P. Prosser, B. M. Smith, and T. Walsh, “Random constraint satisfaction: flaws and structure,” *Constraints*, vol. 6, no. 4, pp. 345–372, 2001.

Research Article

Machine Directional Register System Modeling for Shaft-Less Drive Gravure Printing Machines

Shanhui Liu,¹ Xuesong Mei,¹ Jian Li,² and Li'e Ma¹

¹ School of Mechanical Engineering, Xi'an Jiaotong University, Xianning West Road 28, Xi'an, Shaanxi 710049, China

² School of Mechatronics Engineering, Henan University of Science and Technology, Xiyuan Road 48, Luoyang, Henan 471003, China

Correspondence should be addressed to Shanhui Liu; shanhui.liu820404@163.com

Received 22 June 2013; Accepted 8 September 2013

Academic Editor: William Guo

Copyright © 2013 Shanhui Liu 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 the latest type of gravure printing machines referred to as the shaft-less drive system, each gravure printing roller is driven by an individual servo motor, and all motors are electrically synchronized. The register error is regulated by a speed difference between the adjacent printing rollers. In order to improve the control accuracy of register system, an accurate mathematical model of the register system should be investigated for the latest machines. Therefore, the mathematical model of the machine directional register (MDR) system is studied for the multicolor gravure printing machines in this paper. According to the definition of the MDR error, the model is derived, and then it is validated by the numerical simulation and experiments carried out in the experimental setup of the four-color gravure printing machines. The results show that the established MDR system model is accurate and reliable.

1. Introduction

Recently, with the development of high-speed and high-precision printing, shaft-less drive technology has been widely used in the latest gravure printing machines. In this technology, the printing rollers are driven by independent servo motors, and the register error is controlled through directly adjusting the speed of the pertinent servo motor instead of changing the position of the compensation roller, so the traditional mechanical shaft is eliminated to make printing faster and more accurate. This register mode has the advantages such as higher response frequency, better control accuracy, and lower additional cost. However, the register error in the shaft-less drive mode has a cumulative effect; namely, the adjustment of register error in upstream will affect all register errors in downstream. Hence, the mathematical model of the register system which can accurately reflect the variation of the register errors is needed to be studied to control the register errors accurately.

It is difficult to get an exact register system model because the register errors are affected by various factors, such as the web tension and speed, the precision of parts and assembling, and the web material and production environment. Register

errors are defined as two-dimensional errors: machine directional register (MDR) errors and cross-directional register (CDR) errors. Brandenburg [1] derived a linear mathematical model of the MDR errors of a moving web in a first-order differential equation by using an equilibrium equation of mass which is transported by printing rollers, but this model was based on the mechanical shaft drive mode and could not adapt to the shaft-less drive mode. Yoshida et al. [2–4] developed the MDR system model for the mechanical shaft and shaft-less drive mode, but the model still ignores many influencing factors and is imperfect. Hua et al. [5] got the MDR errors model of gravure printing machines via the system identification method, but this method has poor universality and bad practicability. Although the MDR errors model of the two-color system was established in [6], the MDR system model of the multicolor system was not derived, some key deducing steps were ignored, and some wrong symbols appeared in some formulae, which makes some processes of derivation difficult to be understood for readers. The modeling and control of CDR system were studied in [7–9]. The CDR errors are controlled by the lateral motion of printing rollers, which is greatly different from the control of MDR errors. In addition, initial works of modeling and

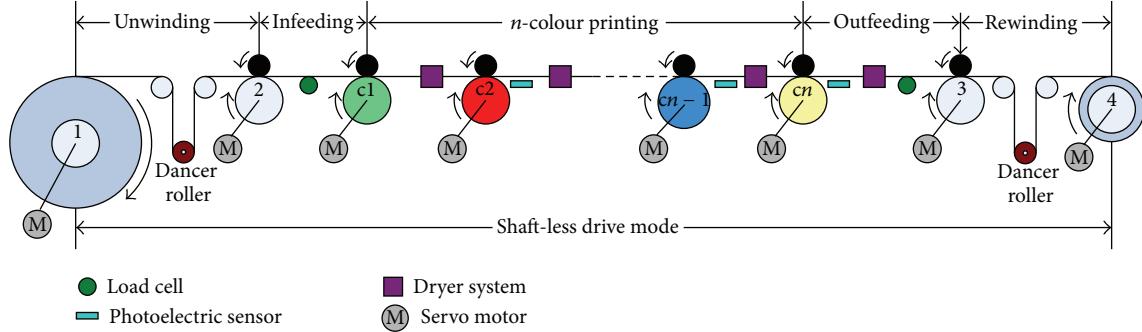


FIGURE 1: Schematic diagram of the n -color gravure printing machines.

control in correlation between MDR and CDR errors were researched in [10, 11]. However, the in-depth research on the MDR system model based on shaft-less drive mode is still not enough in gravure printing machines.

Therefore, in this study, a mathematical model of the MDR system in shaft-less drive mode is derived for gravure printing machines by using an equilibrium equation of mass and considering the fluctuation of tension, speed, and other factors. This paper is structured as follows. Section 2 introduces the schematic diagram of shaft-less drive gravure printing machines. Section 3 establishes the MDR errors model of the two-color system and then obtains the MDR system model of the multicolor gravure printing machines. Section 4 verifies the derived model by simulation and experiments. The results show that the model is accurate enough to describe the MDR errors in shaft-less drive gravure printing machines.

2. Shaft-Less Drive Gravure Printing Machines

The schematic diagram of the n -color gravure printing machines is presented in Figure 1, and n stands for the total number of printing units. This system is composed of an unwinding unit, an infeeding unit, an n -color printing system, an outfeeding unit, and a rewinding unit. Shaft-less drive mode has been used in the machines; that is, all of the driving shafts are driven by independent servo motors. The n -color printing system consists primarily of n printing rollers, and c_1 is the first printing roller whose main task is to maintain the system speed constant. In the printing subsystem, tension is same with that in infeeding zone at steady state. Load cells are installed at idle rollers in the middle of continuous process in both the infeeding and outfeeding units for tension pickup. Two passive dancer rollers are used to reduce tension fluctuations and measure the tension signals simultaneously in the unwinding and rewinding units. Photoelectric sensors are installed in the machine for registering errors pickup.

Control of the gravure printing machines is basically composed of tension control and register control. Register control is required for high printing resolution in the multistage printing systems. Hence, an accurate MDR system

model is essential to ensure adequate printing performance for the gravure printing machines.

3. Mathematical Modeling

3.1. Basic Assumption. In the gravure printing machines, the MDR error of a moving web is defined in two adjacent printing rollers as the relative difference of the distance between the previous printed pattern by the upstream printing roller and the later printed one in the downstream printing roller. To remove unnecessary interference and simplify the mathematical model, the following assumptions are made in the derivation of model.

- (1) There is no slippage between web material and rollers and no mass transfer between the web material and the environment.
- (2) The web material is isotropic, and its properties do not change with temperature and humidity.
- (3) The web cross-section in the unstretched state does not vary along machine direction in a web span.
- (4) The density and modulus of elasticity of the web in the unstretched state are constant in the cross-section.
- (5) The strain in web is small (much less than unity) and uniform within the web span.
- (6) The printing system is in a stable state, in which there are no disturbances from tension and other factors in time $t \leq 0$.

3.2. MDR Error Model of the Two-Color System. The mathematical MDR error model of the two-color system in shaft-less drive gravure printing machines is illustrated in Figure 2 by the coupling between the i th and $(i+1)$ th units, and the i th unit is register errors reference. As shown in Figure 2, a coordinate axis along the machine direction is set up and the position of the i th unit is set as the origin of the coordinate axis. Figures 2(b) and 2(c) show that the color marks on printing roller c_i and c_{i+1} are printed on the web at the printing instants t_1 and t_2 , respectively.

The MDR error in the moving coordinate system is to calculate the difference between the printing point on the roller and the printed point on the web representing

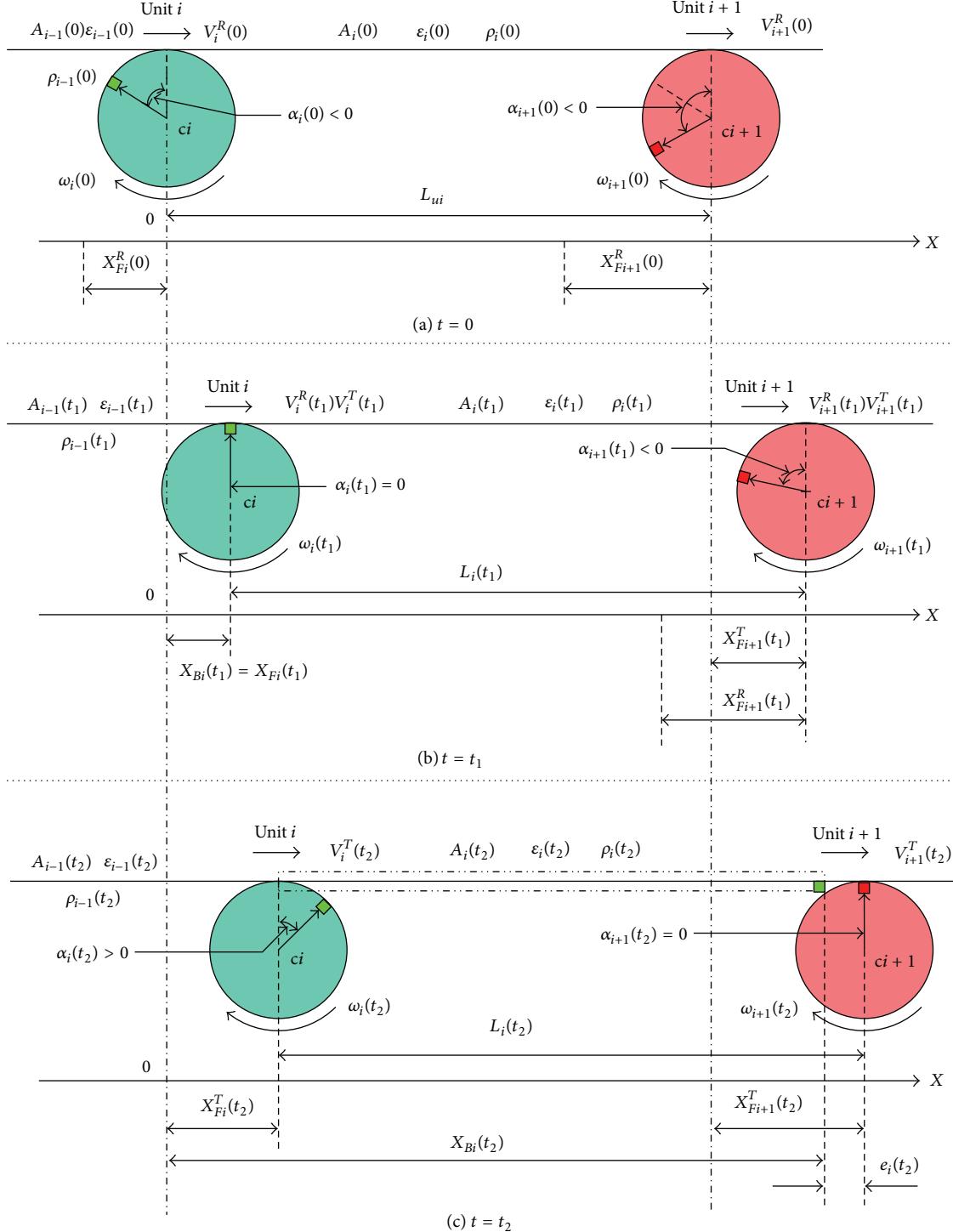


FIGURE 2: MDR error model of the two-color system.

the same point of the pattern at the printing instants. Therefore, according to the moving coordinate system, the MDR error is just the deviation from the coordinate of color mark on the printing roller X_{Fi+1} and that on the web X_{Bi} . At instant t_2 , the MDR error is defined by

$$e_i(t_2) = X_{Bi}(t_2) - X_{Fi+1}(t_2). \quad (1)$$

It is to be noted that the movement of the printing roller can be viewed as a combination of two movements: one is the rotation of the roller and the other is the virtual translational movement of the roller along the machine direction which is caused by the variation of the web length. This variation can be affected by many factors such as dryer system, tension sensors, or some other disturbances.

By using assumptions (1) and (2), the following conclusion can be obtained:

$$\begin{aligned} \frac{d}{dt} \int_0^{L_i} \rho_i(x, t) A_i(x, t) dx \\ = \rho_{i-1}(x, t) A_{i-1}(x, t) V_i^R(t) - \rho_i(x, t) A_i(x, t) V_{i+1}^R(t). \end{aligned} \quad (2)$$

Hence, the mass of the web material entering the printing unit i in the time interval from t_1 to t_2 can be derived as follows:

$$\frac{d}{dt} \int_{X_{Bi}(t_1)}^{X_{Bi}(t_2)} \rho_i(x, t) A_i(x, t) dx = \rho_{i-1}(x, t) A_{i-1}(x, t) V_i^R(t). \quad (3)$$

According to assumptions (3) and (4) and the definition of strain, the relationship between the stretched and unstretched webs can be written as follows:

$$\frac{\rho_i(x, t) A_i(x, t)}{\rho_u A_u} = \frac{1}{1 + \varepsilon_i(x, t)}, \quad (4)$$

where u means the unstretched state of the web.

Combining (3) with (4), we can obtain

$$\int_{X_{Bi}(t_1)}^{X_{Bi}(t_2)} \frac{1}{1 + \varepsilon_i(x, t)} dx = \int_{t_1}^{t_2} \frac{V_i^R(t)}{1 + \varepsilon_{i-1}(x, t)} dt. \quad (5)$$

Under assumption (5), we can obtain

$$\frac{1}{1 + \varepsilon_i(x, t)} = \frac{1}{1 + \varepsilon_i(t)} \approx 1 - \varepsilon_i(t). \quad (6)$$

Combining (5) with (6), we can get

$$(X_{Bi}(t_2) - X_{Bi}(t_1))(1 - \varepsilon_i(t)) = \int_{t_1}^{t_2} (1 - \varepsilon_{i-1}(t)) V_i^R(t) dt. \quad (7)$$

As shown in Figure 2(a), the initial phase angle of the color mark on the printing roller ci is negative and can be calculated by

$$\alpha_i(t) = \alpha_i(0) + \int_0^t \omega_i(t) dt. \quad (8)$$

The coordinate of color mark on the printing roller ci X_{Fi} can be calculated by (9) including two parts, X_{Fi}^R and X_{Fi}^T , which are caused by rotation and virtual translational movement of the printing roller ci , respectively:

$$\begin{aligned} X_{Fi}(t) &= X_{Fi}^R(t) + X_{Fi}^T(t) \\ &= X_{Fi}^R(0) + X_{Fi}^T(0) + \int_0^t [V_i^R(t) + V_i^T(t)] dt. \end{aligned} \quad (9)$$

Under assumption (6), the following equation could be obtained

$$\begin{aligned} X_{Fi}^R(0) &= R_i \alpha_i(0) \leq 0, \\ X_{Fi}^T(0) &= 0. \end{aligned} \quad (10)$$

So, (9) can be rewritten as follows:

$$X_{Fi}(t) = R_i \alpha_i(0) + \int_0^t [V_i^R(t) + V_i^T(t)] dt. \quad (11)$$

At $t = t_1$, as shown in Figure 2(b), the color mark on the printing roller ci is printed on the web, the following equation can be gotten

$$\alpha_i(t_1) = \alpha_i(0) + \int_0^{t_1} \omega_i(t) dt = 0. \quad (12)$$

We can rewrite (12) as follows:

$$\int_0^{t_1} \omega_i(t) dt = -\alpha_i(0) \implies \int_0^{t_1} V_i^R(t) dt = -R_i \alpha_i(0). \quad (13)$$

The coordinate of the color mark on the printing roller ci coincides with that of the color mark on the web at $t = t_1$, so combining (11) with (13), we can get the following:

$$X_{Fi}(t_1) = X_{Bi}(t_1) = \int_0^{t_1} V_i^T(t) dt. \quad (14)$$

Because the i th unit is MDR error reference, the printing roller ci has no virtual translational movement, so $V_i^T(t) = 0$, and we can obtain

$$X_{Bi}(t_1) = \int_0^{t_1} V_i^T(t) dt = 0. \quad (15)$$

Hence, (7) can be rewritten as follows:

$$X_{Bi}(t_2) = \frac{1}{1 - \varepsilon_i(t)} \int_{t_1}^{t_2} (1 - \varepsilon_{i-1}(t)) V_i^R(t) dt. \quad (16)$$

Similarly, as shown in Figure 2, the coordinate of the color mark on the printing roller $ci+1$ can be derived by

$$\begin{aligned} X_{Fi+1}(t) &= X_{Fi+1}^R(t) + X_{Fi+1}^T(t) \\ &= X_{Fi+1}^R(0) + X_{Fi+1}^T(0) + \int_0^t [V_{i+1}^R(t) + V_{i+1}^T(t)] dt. \end{aligned} \quad (17)$$

At $t = 0$, (18) can be obtained as follows:

$$X_{Fi+1}^T(0) = L_{ui}. \quad (18)$$

Derived with the same procedure as X_{Fi} , the following conclusion can be obtained:

$$\int_0^{t_2} V_{i+1}^R(t) dt = -X_{Fi+1}^R(0). \quad (19)$$

Combining (17), (18) with (19), we can get

$$X_{Fi+1}(t_2) = X_{Bi+1}(t_2) = L_{ui} + \int_0^{t_2} V_{i+1}^T(t) dt. \quad (20)$$

From the literature [12], we can obtain

$$\begin{aligned} [1 - \varepsilon_i(t)] \frac{dL_i(t)}{dt} - L_i(t) \frac{d\varepsilon_i(t)}{dt} \\ = V_i^R(t) - V_{i+1}^R(t) - \varepsilon_{i-1}(t) V_i^R(t) + \varepsilon_i(t) V_{i+1}^R(t). \end{aligned} \quad (21)$$

Since $d\varepsilon_i(t)/dt \approx 0$, (21) can be rewritten as follows:

$$\frac{dL_i(t)}{dt} = \frac{1 - \varepsilon_{i-1}(t)}{1 - \varepsilon_i(t)} V_i^R(t) - V_{i+1}^R(t). \quad (22)$$

Because the change of the web length mapped to the coordinate is equivalent to the speed difference between the virtual translational movements of the two printing rollers, it can be expressed as follows:

$$\frac{dL_i(t)}{dt} = V_{i+1}^T(t) - V_i^T(t). \quad (23)$$

Due to $V_i^T(t) = 0$, combining (22) with (23), (24) can be obtained as follows:

$$V_{i+1}^T(t) = \frac{1 - \varepsilon_{i-1}(t)}{1 - \varepsilon_i(t)} V_i^R(t) - V_{i+1}^R(t). \quad (24)$$

Combining (20) with (24), (20) can be rewritten as follows:

$$X_{Fi+1}(t_2) = L_{ui} + \int_0^{t_2} \left[\frac{1 - \varepsilon_{i-1}(t)}{1 - \varepsilon_i(t)} V_i^R(t) - V_{i+1}^R(t) \right] dt. \quad (25)$$

By using assumptions (5), we can get the following:

$$\frac{1}{1 - \varepsilon_i(t)} \approx 1 + \varepsilon_i(t), \quad (26)$$

$$\varepsilon_{i-1}(t) \varepsilon_i(t) \approx 0.$$

Therefore, (25) can be simplified as follows:

$$X_{Fi+1}(t_2) = L_{ui} + \int_0^{t_2} \left[V_i^R(t) - V_{i+1}^R(t) - (\varepsilon_{i-1}(t) - \varepsilon_i(t)) V_i^R(t) \right] dt. \quad (27)$$

According to the definition of MRD error, and combining (1), (16), and (27), the MDR error between the i th and unit $(i+1)$ th printing units at the instant t_2 can be calculated as follows:

$$e_i(t_2) = \frac{1}{1 - \varepsilon_i(t)} \int_{t_1}^{t_2} (1 - \varepsilon_{i-1}(t)) V_i^R(t) dt - L_{ui}$$

$$- \int_0^{t_2} \left[V_i^R(t) - V_{i+1}^R(t) - (\varepsilon_{i-1}(t) - \varepsilon_i(t)) V_i^R(t) \right] dt. \quad (28)$$

The running time $t_{Ti} = t_2 - t_1$ represents the time that a color mark on the web takes to move from printing roller ci to printing roller $ci+1$, so $t_1 = t_2 - t_{Ti}$, (28) can be simplified as follows:

$$e_i(t_2) = \int_0^{t_2} \left(V_{i+1}^R(t) - \varepsilon_i(t) V_i^R(t) \right) dt$$

$$- \int_0^{t_2 - t_{Ti}} (1 - \varepsilon_{i-1}(t)) V_i^R(t) dt - L_{ui}. \quad (29)$$

Extending (28) to any time interval, then we can get

$$e_i(t) = \int_0^t \left[V_{i+1}^R(t) - \varepsilon_i(t) V_i^R(t) \right] dt$$

$$- \int_0^{t-t_{Ti}} (1 - \varepsilon_{i-1}(t)) V_i^R(t) dt - L_{ui}. \quad (30)$$

The differential form of MDR error between the i th and $(i+1)$ th units can be derived as follows:

$$\frac{de_i(t)}{dt} = V_{i+1}^R(t) - V_i^R(t - t_{Ti})$$

$$+ \varepsilon_{i-1}(t - t_{Ti}) V_i^R(t - t_{Ti}) - \varepsilon_i(t) V_i^R(t). \quad (31)$$

In the shaft-less drive printing system, there is no change of the web length caused by compensation rollers, so $\Delta L_i(t)$ is mainly caused by load cells, dryer systems, and other disturbances. In practice, $\Delta L_i(t) \ll L_{ui}$, so we can obtain

$$L_i(t) = L_{ui} + \Delta L_i(t) \approx L_{ui}. \quad (32)$$

In (21), we substitute L_{ui} for $L_i(t)$, and then (21) can be rewritten as follows:

$$L_{ui} \frac{d\varepsilon_i(t)}{dt} = [1 - \varepsilon_i(t)] V_{i+1}^R(t) - [1 - \varepsilon_{i-1}(t)] V_i^R(t). \quad (33)$$

Combining (31) with (33), the MDR error model of the two-color system can be obtained as follows:

$$\frac{de_i(t)}{dt} = V_{i+1}^R(t) - V_i^R(t - t_{Ti})$$

$$+ \varepsilon_{i-1}(t - t_{Ti}) V_i^R(t - t_{Ti}) - \varepsilon_i(t) V_i^R(t), \quad (34)$$

$$L_{ui} \frac{d\varepsilon_i(t)}{dt} = [1 - \varepsilon_i(t)] V_{i+1}^R(t) - [1 - \varepsilon_{i-1}(t)] V_i^R(t).$$

Considering $V_i^R(t) = R_i \omega_i(t)$ and $\varepsilon_i(t) = T_i(t)/AE$, we can obtain

$$AE \frac{de_i(t)}{dt} = AE (R_{i+1} \omega_{i+1}(t) - R_i \omega_i(t - t_{Ti}))$$

$$+ R_i (T_{i-1}(t - t_{Ti}) \omega_i(t - t_{Ti}) - T_i(t) \omega_i(t)),$$

$$L_{ui} \frac{dT_i(t)}{dt} = AE (R_{i+1} \omega_{i+1}(t) - R_i \omega_i(t))$$

$$+ T_{i-1}(t) R_i \omega_i(t) - T_i(t) R_{i+1} \omega_{i+1}(t). \quad (35)$$

In the practical printing process of gravure printing machines, the change of the web cross-section in different web spans is very small and negligible, so we substitute average A for $A_i(t)$ in (35). Equation (35) shows that the MDR mathematical model of the two-color gravure printing machines is expressed by two equations representing the dynamics of tension fluctuation between units and the dynamics of the register error fluctuation, respectively.

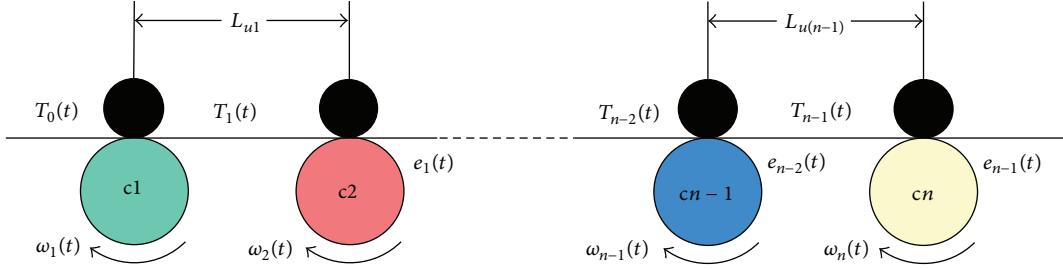
FIGURE 3: Schematic diagram of the n -color gravure printing machine.

FIGURE 4: Experimental setup of the four-color gravure printing machines.

3.3. MDR System Model of the n -Color Gravure Printing Machines. The schematic diagram of the n -color MDR system is presented in Figure 3. $T_0(t)$ is the tension of the infeeding zone at steady state and can be measured by load cells.

According to (35) and Figure 3, the MDR system model of the n -color gravure printing machines can be expressed as follows:

$$\begin{aligned} AE \frac{de_1(t)}{dt} &= AE(R_2\omega_2(t) - R_1\omega_1(t - t_{T1})) \\ &\quad + R_1(T_0(t - t_{T1})\omega_1(t - t_{T1}) - T_1(t)\omega_1(t)), \\ L_{u1} \frac{dT_1(t)}{dt} &= AE(R_2\omega_2(t) - R_1\omega_1(t)) \\ &\quad + T_0(t)R_1\omega_1(t) - T_1(t)R_2\omega_2(t), \\ &\vdots \\ AE \frac{de_{n-1}(t)}{dt} &= AE(R_n\omega_n(t) - R_{n-1}\omega_{n-1}(t - t_{T(n-1)})) \\ &\quad + R_{n-1}(T_{n-2}(t - t_{T(n-1)})\omega_{n-1}(t - t_{T(n-1)}) \\ &\quad - T_{n-1}(t)\omega_{n-1}(t)), \\ L_{u(n-1)} \frac{dT_{n-1}(t)}{dt} &= AE(R_n\omega_n(t) - R_{n-1}\omega_{n-1}(t)) \\ &\quad + T_{n-2}(t)R_{n-1}\omega_{n-1}(t) - T_{n-1}(t)R_n\omega_n(t). \end{aligned} \quad (36)$$

The $\omega_i(t)$ is composed of a stationary component ω^* which corresponds to web feed speed and a variable component $\Delta\omega_i(t)$ determined by the MDR control input. In particular, $\omega_1(t)$ is used as a speed reference, so we can get (37) as follows:

$$\begin{aligned} \omega_1(t) &= \omega^*, \\ \omega_2(t) &= \omega^* + \Delta\omega_2(t), \\ &\vdots \\ \omega_n(t) &= \omega^* + \Delta\omega_n(t). \end{aligned} \quad (37)$$

In (36), t_{Ti} is also known as time delay and is defined as follows:

$$t_{Ti} = \frac{L_{ui}}{R_i\omega_i} \approx \frac{L_{ui}}{R_i\omega^*}. \quad (38)$$

Equations (36) and (37) show that the n -color MDR system in the gravure printing machine is a multi-input multioutput, nonlinear, strong-coupling, and time-delay system. In the system, $\omega_2(t) \sim \omega_n(t)$ are the input signals, and $e_1(t) \sim e_{n-1}(t)$ are the output signals.

4. Simulation and Experiments

4.1. Experimental Setup and MDR Simulation Model of the Four-Color System. To investigate the reliability of the derived MDR mathematical model, numerical simulation and experiments under open-loop control are carried out by using MATLAB and experimental setup for the four-color gravure printing machines. Figures 4 and 5 show the setup configuration of the shaft-less drive gravure printing machine.

The system includes an unwinding unit, an infeeding unit, an outfeeding unit, a rewinding unit, and a four-color printing subsystem, and all of the driving shafts are driven by independent servo motors. The printing rollers are driven by servo motors (YASKAWA SGMGH-44), and printing speed is obtained by the encoder on the servo motor. The register controller (KESAI ST3000) works in shaft-less drive mode, and the MDR errors are measured using photoelectric sensors (XINMAO XM08). Tension signals are tested by load cells (MITSUBISHI LX-030TD) and tension measuring amplifiers (MITSUBISHI LM-10TA) mounted on the shaft of idle rollers. The shaft-less drive system is controlled by multiaxis

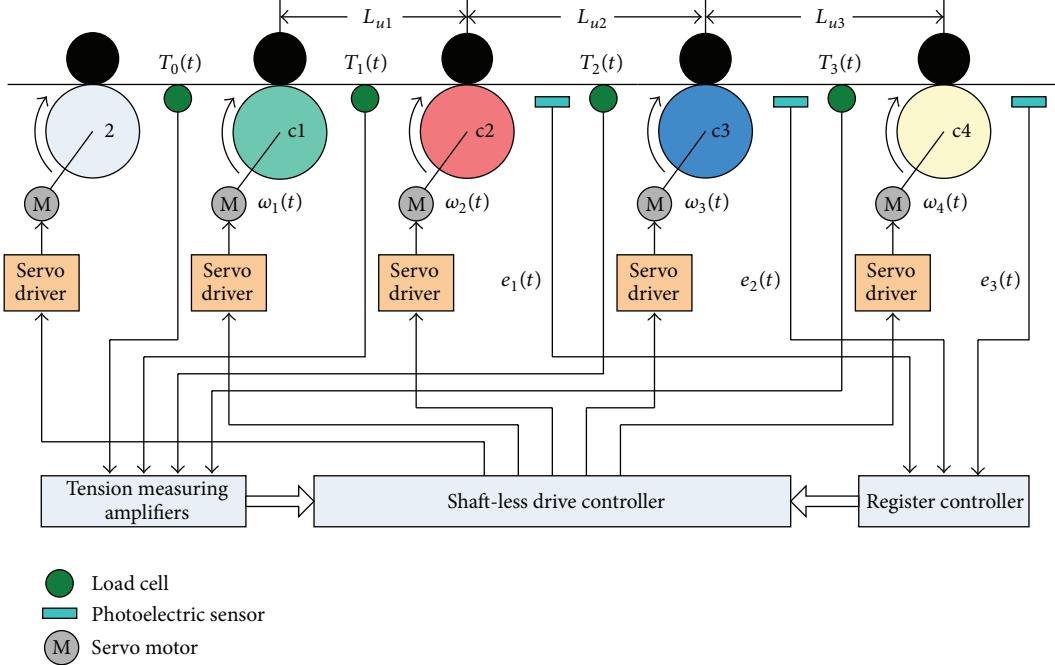


FIGURE 5: Control configuration of the experimental setup.

TABLE 1: System parameters.

Parameters	Values	Units
Entry tension to 1th unit (T_0)	100	N
Web length (L_{ui})	9.1	m
Radius of the printing roller (R_i)	0.2	m
Web cross-section area (A)	2.0×10^{-5}	m^2
Modulus of elasticity of the web (E)	160	MPa

controller (GOOGOLTECH T8VME). The experimental web material is biaxially oriented polypropylene (BOPP). As shown in Figure 5, tension of the printing subsystem is same with that in the infeeding zone at steady state and is not controlled; in other words, $T_1(t)$, $T_2(t)$, and $T_3(t)$ are not controlled. In the infeeding unit, $T_0(t)$ is adjusted by changing the speed of the roller 2. After preregistration, $e_1(t)$, $e_2(t)$, and $e_3(t)$ work in the open-loop control mode and are sampled once per revolution of the printing rollers. The parameters of the experimental setup are summarized in Table 1.

According to Table 1, we can get

$$\begin{aligned} R_1 &= R_2 = R_3 = R_4 = R, \\ L_{u1} &= L_{u2} = L_{u3} = L_u, \\ t_{T1} &= t_{T2} = t_{T3} = t_T. \end{aligned} \quad (39)$$

Hence, combining (36) with (39), the MDR system model of the four-color gravure printing machines in the simulation can be expressed as follows:

$$\begin{aligned} AE \frac{de_1(t)}{dt} &= AER(\omega_2(t) - \omega_1(t - t_T)) \\ &+ R(T_0(t - t_T)\omega_1(t - t_T) - T_1(t)\omega_1(t)), \end{aligned}$$

$$\begin{aligned} AE \frac{de_2(t)}{dt} &= AER(\omega_3(t) - \omega_2(t - t_T)) \\ &+ R(T_1(t - t_T)\omega_2(t - t_T) - T_2(t)\omega_2(t)), \\ AE \frac{de_3(t)}{dt} &= AER(\omega_4(t) - \omega_3(t - t_T)) \\ &+ R(T_2(t - t_T)\omega_3(t - t_T) - T_3(t)\omega_3(t)), \\ L_u \frac{dT_1(t)}{dt} &= AER(\omega_2(t) - \omega_1(t)) \\ &+ R(T_0(t)\omega_1(t) - T_1(t)\omega_2(t)), \\ L_u \frac{dT_2(t)}{dt} &= AER(\omega_3(t) - \omega_2(t)) \\ &+ R(T_1(t)\omega_2(t) - T_2(t)\omega_3(t)), \\ L_u \frac{dT_3(t)}{dt} &= AER(\omega_4(t) - \omega_3(t)) \\ &+ R(T_2(t)\omega_3(t) - T_3(t)\omega_4(t)). \end{aligned} \quad (40)$$

4.2. Results and Analysis of the Simulation and Experiments. In order to ensure the consistency of the simulation and experiments, the parameters used in the numerical simulation are taken from Table 1. Because the simulation and experiments under open-loop control are carried out, the speed of all servo motors is the same in the four-color printing system. In order to verify the developed MDR system model of the n -color gravure printing machines, $T_0(t)$ has a step change from 100 N to 110 N at $t = 5$ s when ω_1 is equal to 100 r/min and 200 r/min. The responses of the MDR errors

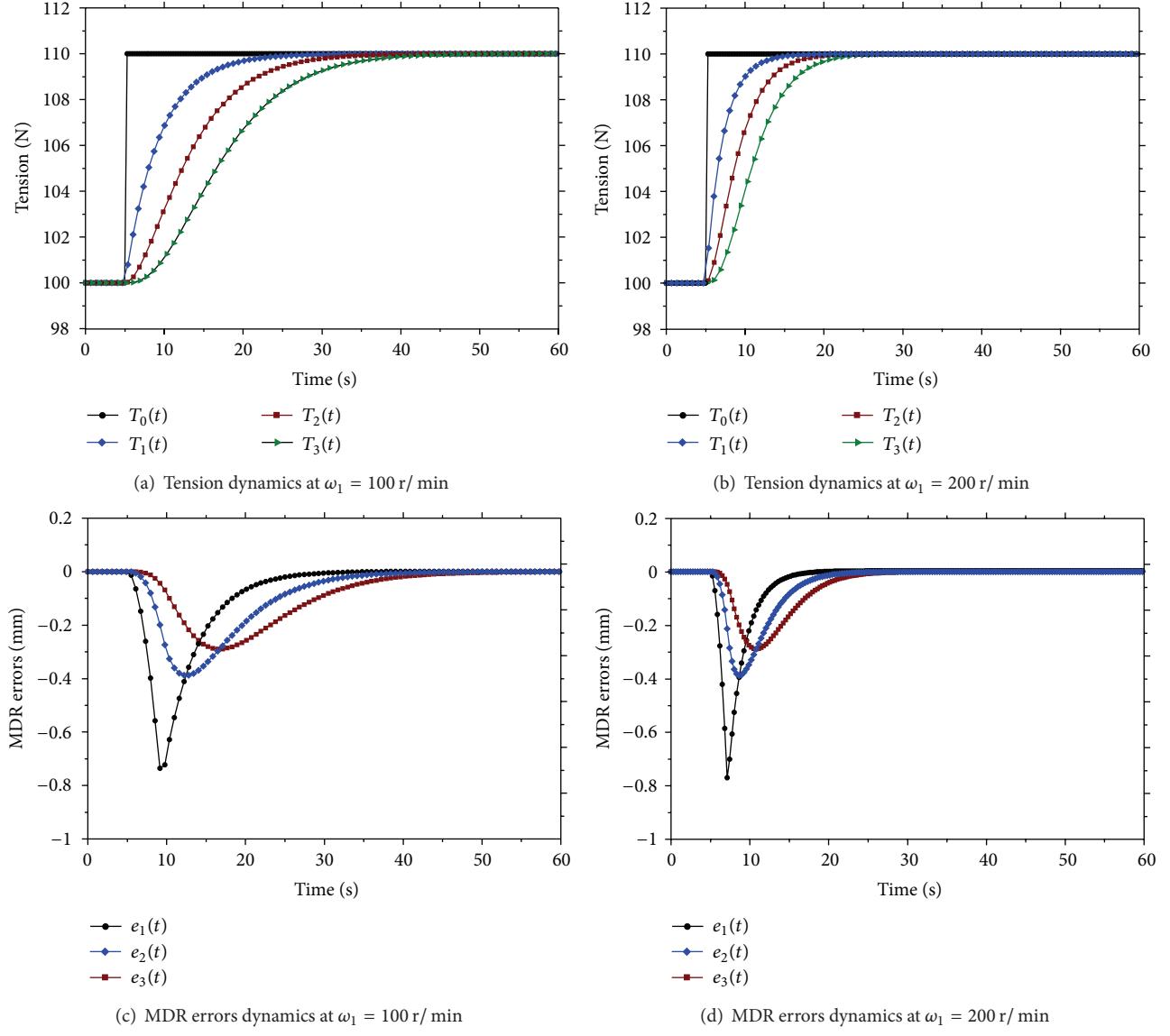


FIGURE 6: Open-loop dynamics of the MDR errors in simulation.

and tension signals in simulation and experiments are shown in Figures 6 and 7, respectively.

As shown in Figure 6, the step change of $T_0(t)$ causes the step change of $T_1(t)$, $T_2(t)$, and $T_3(t)$, and with an increase in $\omega_1(t)$, the time that $T_1(t)$, $T_2(t)$, and $T_3(t)$ take to reach the steady state value is gradually shortened. For example, when $\omega_1(t)$ is equal to 100 r/min and 200 r/min, $T_1(t)$ reaches 110 N at 35 s and 20 s, respectively. The step change of $T_0(t)$ also directly causes the change of $e_1(t)$ and then the changes of $e_2(t)$ and $e_3(t)$ by coupling characteristics, and as the speed increases, the interval from the peak of $e_i(t)$ to the peak of $e_{i+1}(t)$ is gradually shortened; for example, when $\omega_1(t)$ is equal to 100 r/min and 200 r/min, the interval from the peak of $e_1(t)$ to the peak of $e_2(t)$ is 3.11 s and 1.56 s, respectively.

Figures 6 and 7 illustrate that although there are some disturbances in the experiments, the experimental results are

almost coincident with the simulation curves. For example, with an increase in $\omega_1(t)$, the time that tension signals take to reach the steady state value is gradually shortened and so on. Comparing with Figures 6(c), 6(d), 7(c), and 7(d), respectively, the peak of $e_i(t)$ in the experiments is slightly more than that in the simulation, such as when $\omega_1(t)$ is equal to 200 r/min, the peaks of $e_1(t)$ in the experiments and simulation are -0.82 mm and -0.78 mm, respectively. There are two primary reasons: one is that there are external disturbances from environment in the experiments, and the other is that there are errors between the simulation parameters and real experimental parameters.

The results of the simulation and experiments indicate that the established MDR system model is accurate and reliable and can be used to design a MDR controller. At the same time, the results also offer a powerful proof that the

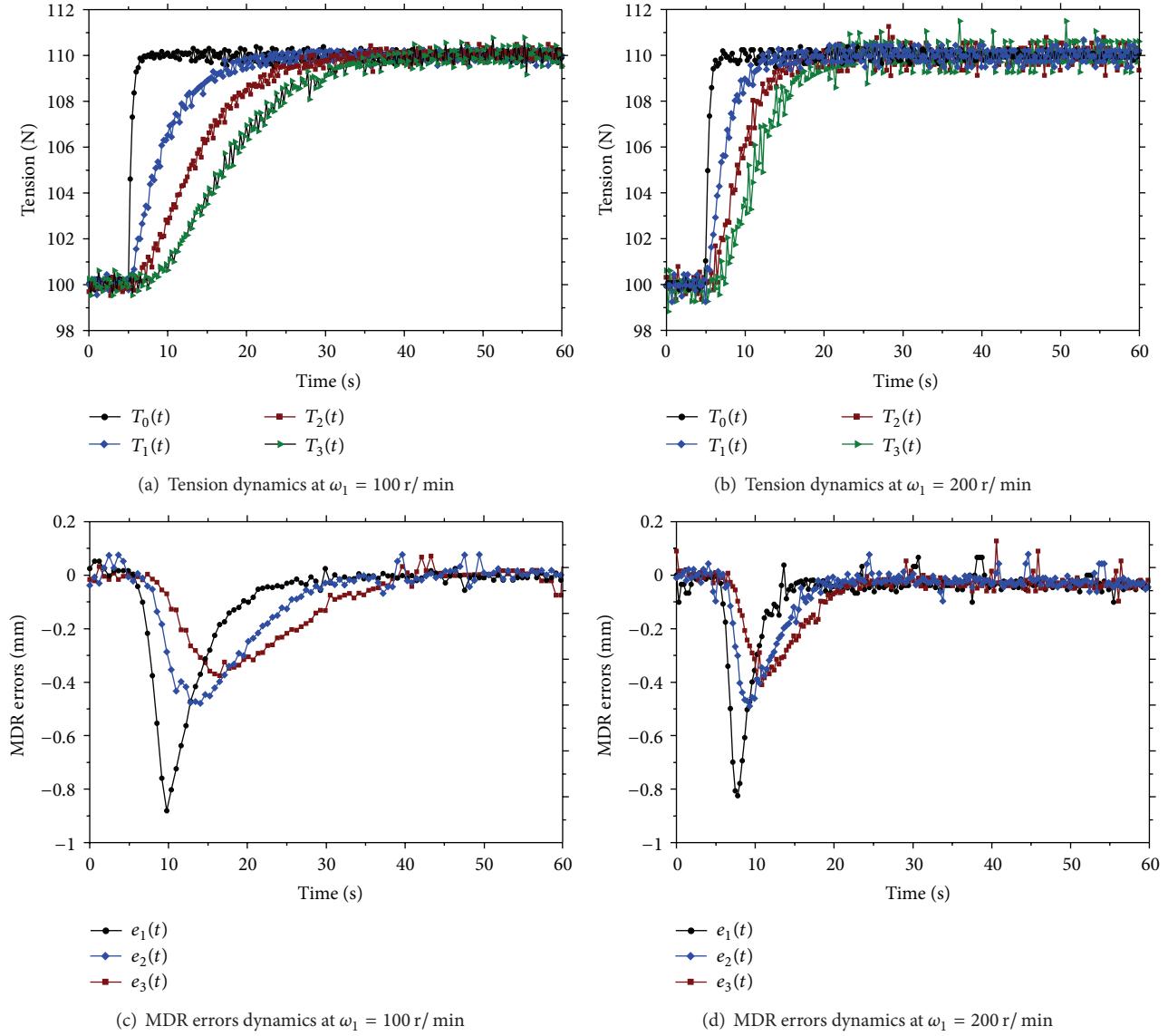


FIGURE 7: Open-loop dynamics of the MDR errors in experiments.

MDR system of the multicolor gravure printing machines is a multi-input multioutput, nonlinear, strong-coupling, and time-delay system.

5. Conclusions

In the gravure printing machines, the register accuracy is the most important in performance. Therefore, in order to improve the control accuracy of MDR system, a nonlinear mathematical model of the MDR system in shaft-less drive mode is presented and derived in detail for the multicolor gravure printing machines in this paper. First, we introduce the schematic diagram of the shaft-less drive gravure printing machines. Secondly, we make a deep analysis of the causes and influencing factors of the MDR errors, then establish the MDR error of the two-color system, and further obtain the MDR system model of the n -color gravure printing machines.

At last, the proposed model is verified by the numerical simulations and experimental studies. Therefore, it can be concluded that the MDR system model is accurate enough to describe dynamics of the MDR errors in multicolor printing system and is worthwhile to be developed as an effective MDR controller in future study.

Nomenclature

- α_i : Phase angle of the color mark on the roller c_i
- t_i : Printing moment of the color mark on the roller c_i
- t_{Ti} : Transmission time of the color mark from i th to $(i+1)$ th printing units
- ω_i : Angular velocity of rotation of the roller c_i

- X_{Fi} : Coordinate on x -axis of the color mark on the roller ci
- X_{Bi} : Coordinate on x -axis of the color mark on the web
- e_i : Register error between i th and $(i+1)$ th printing units
- R_i : The radius of the roller ci
- L_{ui} : Length of the free web between i th and $(i+1)$ th printing units
- L_i : Actual length of the web between i th and $(i+1)$ th printing units
- ΔL_i : The length variation of the web between i th and $(i+1)$ th printing units
- ε_i : Strain of the web between i th and $(i+1)$ th printing units
- T_i : Tension of the web between i th and $(i+1)$ th printing units
- V_i^T : Virtual velocity of the roller ci
- V_i^R : Web velocity of the roller ci
- E : The modulus of elasticity of the web material
- ρ_i : Density of the web between i th and $(i+1)$ th printing units
- A_i : Cross-section of the web between i th and $(i+1)$ th printing units.

Conflict of Interests

The authors declare that they have no financial and personal relationships with other people or organizations that can inappropriately influence our work; there is no professional or other personal interest of any nature or kind in any product or company that could be construed as influencing the position presented in, or the review of, the paper.

Acknowledgments

This research is supported by the Program for Changjiang Scholars (IRT1172) and the National Science & Technology Pillar Program of China, under Grant no. 2012BAF13B06.

References

- [1] G. Brandenburg, “New mathematical models for web tension and register error,” *Noordnederlands Boekbedrijf*, vol. 1, pp. 411–438, 1977.
- [2] T. Yoshida, S. Takagi, T. Shen, and Y. Muto, “Modeling and cooperative resister control of gravure printing press,” *Nihon Kikai Gakkai Ronbunshu C*, vol. 74, no. 2, pp. 339–345, 2008.
- [3] T. Yoshida, S. Takagi, Y. Muto, and T. Shen, “Register control of rotogravure printing press—Application of nonlinear control theory to sectional drive printing press,” *IEEJ Transactions on Industry Applications*, vol. 128, no. 9, pp. 1133–1139, 2008.
- [4] T. Yoshida, S. Takagi, Y. Muto, and T. Shen, “Register control of rotogravure printing press. Application of nonlinear control theory to sectional drive printing press,” *Electronics and Communications in Japan*, vol. 94, no. 1, pp. 17–24, 2011.
- [5] D. Z. Hua, L. Xi, and Y. Xiaomeng, “Model and control for gravure register printing system,” in *Proceedings of the 26th Chinese Control Conference (CCC '07)*, pp. 277–280, Zhangjiajie, China, July 2007.
- [6] J. Li, X. Mei, T. Tao, and S. Liu, “Research on the register system modelling and control of gravure printing press,” *Proceedings of the Institution of Mechanical Engineers C*, vol. 226, no. 3, pp. 626–635, 2012.
- [7] H. Kang, C. Lee, and K. Shin, “A novel cross directional register modeling and feedforward control in multi-layer roll-to-roll printing,” *Journal of Process Control*, vol. 20, no. 5, pp. 643–652, 2010.
- [8] H.-K. Kang, C.-W. Lee, J.-M. Lee, and K.-H. Shin, “Cross direction register modeling and control in a multi-layer gravure printing,” *Journal of Mechanical Science and Technology*, vol. 24, no. 1, pp. 391–397, 2010.
- [9] H. Wang, D. Logghe, and D. Miskin, “Physical modelling and control of lateral web position for wallpaper making processes,” *Control Engineering Practice*, vol. 13, no. 4, pp. 401–412, 2005.
- [10] H.-K. Kang, C.-W. Lee, and K.-H. Shin, “Novel modeling of correlation between two-dimensional registers in large-area multilayered roll-to-roll printed electronics,” *Japanese Journal of Applied Physics*, vol. 50, no. 1, Article ID 016701, 7 pages, 2011.
- [11] H. Kang, C. Lee, J. Lee, and K. Shin, “Two-dimensional register modeling and control in multi-layer roll-to-roll e-printing systems,” in *Proceedings of the 18th IFAC World Congress*, vol. 18, pp. 6763–6770, 2011.
- [12] S. Liu, X. Mei, F. Kong, and J. Shen, “Tension controller design for unwinding tension system based on active disturbance rejection control,” in *Proceedings of the IEEE International Conference on Mechatronics and Automation*, pp. 1798–1803, Chengdu, China, 2012.

Research Article

An Efficient Schema for Cloud Systems Based on SSD Cache Technology

Jinjiang Liu,¹ Yihua Lan,¹ Jingjing Liang,¹ Quanzhou Cheng,¹ Chih-Cheng Hung,^{2,3} Chao Yin,⁴ and Jiadong Sun⁴

¹ School of Computer and Information Technology, Nanyang Normal University, Nanyang 473061, China

² Sino-US Intelligent Information Processing Joint Lab, Anyang Normal University, Anyang 455000, China

³ Center for Biometrics Research, Southern Polytechnic State University, Atlanta, GA 30060-2896, USA

⁴ School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

Correspondence should be addressed to Yihua Lan; lanhua_2000@sina.com

Received 11 July 2013; Accepted 7 September 2013

Academic Editor: William Guo

Copyright © 2013 Jinjiang Liu 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.

Traditional caching strategy is mainly based on the memory cache, taking read-write speed as its ultimate goal. However, with the emergence of SSD, the design ideas of traditional cache are no longer applicable. Considering the read-write characteristics and times limit of erasing, the characteristics of SSD are taken into account as far as possible at the same time of designing caching strategy. In this paper, the flexible and adaptive cache strategy based on SSD is proposed, called FAC, which gives full consideration to the characteristics of SSD itself, combines traditional caching strategy design ideas, and then maximizes the role SSD has played. The core mechanism is based on the dynamic adjustment capabilities of access patterns and the efficient selection algorithm of hot data. We have developed dynamical adjust section hot data algorithm, DASH in short, to adjust the read-write area capacity to suit the current usage scenario dynamically. The experimental results show that both read and write performance of caching strategy based on SSD have improved a lot, especially for read performance. Compared with traditional caching strategy, the technique can be used in engineering to reduce write times to SSD and prolong its service life without lowering read-write performance.

1. Introduction

In recent years, CPU has been faster and faster, but storage system is always the performance's bottleneck of computer system. Meanwhile, in a computer system, I/O is still the biggest bottleneck. For some applications which have quite high requirements on response time, simply to improve CPU performance cannot improve the overall performance of a system; hence, it is necessary to improve the I/O performance and reduce the speed difference between storage system and CPU. However, caching technology is good enough to make up the gap between the performance of storage system and CPU. Through the analysis of locality of reference and accessing frequency, part of data will be stored in cache, greatly reducing the response time and improving the overall performance of system.

In order to solve the problem that the accessing speed of storage system cannot keep up with the processing speed of CPU, we brought in caching technology, whose mechanism is putting those frequently accessed data in fast accessing device, speeding up the access and reducing the latency time [1]. In the caching technology research, hot spot is still the policy to balance recency and frequency, while detection-based is the direction of the efforts, which combines those technologies in other fields of computer with cache replacement technology. Most of the current caching algorithms take into account locality of reference and accessing frequency; yet, it is still a big problem about how to combine the two factors to adapt to the current access features. Furthermore, each caching algorithm has its own emphasis and can only be applied to some sort of access pattern or service environment in using. But how to design a more versatile caching strategy

which can make dynamical adjustment according to different access patterns is an exciting research.

All traditional caching strategies take improving cache hit ratio as their main goal [2–4], an accurate method of evaluating for disks which have the same read-write cost. However, for the current SSD mainly based on flash media, read-write performance is greatly different, and, moreover, service time of SSD should be taken into account for the magnifying of write.

Thus, we should evaluate caching strategies for SSD by access cost [5, 6]. CFLRU is the first cache replacement algorithm, which applies LRU into flash medium storage and takes features of flash medium into consideration, prolonging write of dirty pages as far as possible and reducing write to flash medium. But there are some limitations because this strategy cannot make dynamical adjustment based on diverse access patterns and is greatly dependent on boundary definition and window size of working area and erasing area. So a new strategy is needed to include all the above [7–9].

With the emergence of SSD, it brings about traditional caching strategy with many new ideas because of its superior read-write performance. Nevertheless, SSD cannot spread universally for its fatal flaws: the high price and limit of erase-write times. Unfortunately, if we apply it to cache technologies in the future, these problems must be taken into consideration, and then SSD can play its best role [10–12]. Thus, there are still a lot of needs for research and exploration, such as how to apply SSD to cache technologies and how to combine SSD with features of other storage devices such as memory and disks. Different from disk-oriented caching strategy, caching strategy based on SSD does not take simply improving cache hit rate as its main target but needs to evaluate accessing overhead of system, reducing read-write response time as far as possible. In view of read-write characteristics of flash media and service life of SSD, we should reduce write times to SSD as much as possible without affecting the system performance. Besides, we need to design caching strategy that is suitable for different usage scenarios, and it can make dynamical adjustment according to different usage scenarios and read-write ratio, dropping accessing overhead to the minimum in the range that the system allows.

Flexible and adaptive cache strategy based on SSD-FAC strategy designed in this paper is a kind of caching algorithm based on SSD, the ultimate goal of which is to reduce response time of the system and the accessing overhead as far as possible and extend the service life of SSD when considering the limit of SSD erase-write times and without affecting the performance.

At the same time, it can make dynamical adjustment according to the current program access pattern and system response to adapt to the current usage scenario and achieve optimal performance.

The contributions of this paper are described as follows.

- (i) In view of the analysis of some existing caching strategies, we abandon the design schema taking read-write speed as ultimate goal and propose design thought whose ultimate goal is to reduce accessing

overhead, offering the basis for the design of FAC strategy, which can have ability of access patterns to make dynamical adjustment to adapt to the different scenes.

- (ii) Research and design of the selection algorithm of hot data—DASH, which selects comparable amount of hot data each time according to the capacity of SSD. Because of the impact on system services caused by data migration, data migration does not process separately but processes with requests processing.
- (iii) The result shows that the strategy can make dynamic adjustment according to the current access pattern and greatly reduce the write times to SSD and extend the service life of SSD while increasing the system performance of read and write. FAC cache is 13.5% higher than flash cache in sequential read request and 18.7% in sequential write, while it is 9.2% and 30.1% higher than flash cache in random read request and random write request averagely.

The rest of this paper is organized as follows. Section 2 shows related works. The design of FAC is introduced in Section 3. Section 4 gives the experimental evaluation and results. We make conclusions in Section 5.

2. Related Work

At present, many researchers have proposed flash-based buffer management algorithm for SSD usage scenarios, such as FAB [13], CLC [14], BPLRU [15], BPAC [16], BPCLC [17], CFLRU [18], and improved CRLRU [19]. Taking advantage of flash read-write performance asymmetry, CFLRU is a kind of buffer replacement strategy which first replaces read-only pages and assumes that write cost of flash is far greater than read cost. Its core idea is dividing LRU linked list into two parts: working area and replacement area. Once cache is full and some data needs to be replaced to outside, supposing there exists read-only data in replacement area, CFLRU will choose read-only pages for replacement according to LRU. When there are only dirty pages in replacement area, dirty pages at the tail of the linked list will be replaced. Some other researchers improved traditional LRU and LFU to accommodate diverse application requirements.

SieveStore [20] designed by Pritchett and Thottethodi and some others put forward two filter storage solutions based on high traffic, using a small SSD which can be shared to store hot data. In order to lower SSD write allocation selective buffer allocation strategy is adopted to make full use of cache. We can use two practical filtering methods to accurately find accessed frequently blocks. The first approach is an offline, discrete-allocation variant-SieveStore-D, which enforces selectivity using an access-count-based discrete batch-allocation (ADBA) mechanism. SieveStore-D maintains precise state of block, by logging each access, and periodically combines the states of block in an offline pass. Blocks are allocated into SSD only if their access count in an epoch exceeds a threshold value. The second approach is an online, continuous-allocation variant-SieveStore-C, which

sieves accesses using a hysteresis-based, lazy cache allocation mechanism wherein disk blocks are allocated into SSD on the n times (for some threshold n) miss over a recent time window. To maintain the state of blocks that are not cached, SieveStore-C uses a two-tier structure, a preliminary tier that sieves with imprecise (potentially aliased) state of block, followed by an accurate tier to maintain the quality of sieving.

Griffin [21] put forward by Soundararaja and some others uses disk as write cache of SSD, which can reduce write times to SSD in the case of high performance. Through tests, we found that Griffin can prolong the service life of SSD apparently and reduce the average I/O latency by 56%.

Within the industry, many companies put forward some solutions for SSD as cache and hierarchical storage medium.

SSD cache scheme of Huawei Symantec is called SmartCache. The scheme uses one or more SSDs as cache array, called SmartCache buffer pool. Acquire hot data statistics of other mechanical disks in the array and then every once in a while update to buffer pool; thus, when hot data is accessed, it can be directly read in SmartCache, which just takes enough advantage of high read performance of SSD, improving the read performance of the overall system. The scheme is mainly suitable for the case when there exist more read and less write hot data and mainly random small requests.

LSI launched LSI Mega RAID Cache Cade Pro2.0 read/write cache software used for some LSI Mega RAID 6 Gb/s SATA+SAS control cards, which uses SSD as a fast buffer to store hot data, improving read-write performance effectively, reducing I/O time delay, speeding up application response, and shortening RAID accessing and rebuilding time. This software can import hot data into SSD cache dynamically, with no need of artificially manual configuration.

Since 2009, EMC has successively launched two generations of FAST. The first generation of FAST is applicable to some storage series products, such as CLAR ii ON and Symmetrix. Function of FAST in different product lines is slightly different in details of setting and executing, however, the same on the basic modes of operation. In view of that, automatic migration of FAST is the whole LUN, and the granularity is not elaborate enough; thus, the layered effect is not optimal, and requirement of the source migration needs is too high. So far, FAST has been upgraded to the second generation, which is called FAST VP (short for virtual pool). As EMC announced, FAST VP adopts Sub-LUN level automatic migration technology, which means that, because of LUN, fidelity of data can go with Symmetrix VMAX and newly launched VNX to integrate storage products. In this paper, we have used the same idea to select the hot data and cold data.

3. FAC

3.1. The Architecture of FAC. During designing FAC caching algorithm, considering three relations among memory cache, SSD and disk, and their respective access features, FAC caching strategy can be divided into two modules: memory cache module and SSD cache module. The memory cache module is mainly applied to cache the hottest data, and according to the analysis of some accessing traces, we know

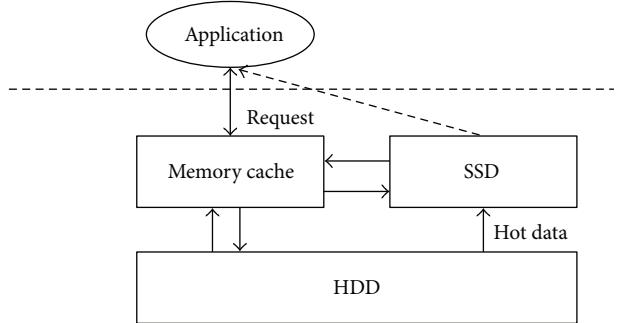


FIGURE 1: The architecture of FAC.

that, in particular access pattern, only 4% data in average will be accessed in a large number, whose traffic takes over more than 90% of the total traffic, and these data will be stored in memory cache and attained directly when application needs. Besides, 3%–7% data belongs to hotter data and will be stored in SSD according to filtering mechanism of memory cache. The rest of the data will be directly written to the underlying disk due to its extremely low access frequency, avoiding the pollution to SSD caused by cold data. The architecture of FAC caching strategy is shown in Figure 1.

3.2. Design of Memory Cache Module

3.2.1. Data Organization. Based on read-write performance difference of SSD, data in memory cache can be divided into read area and write area. Data application read request is stored in read area and data write request is stored in write area. Hot data that has been recently accessed is stored both in read and write areas. If one of them is full, according to hot data filtering mechanism of memory cache, filtered hot data will be imported into SSD and cold data will be directly written into disk. As shown in Figure 2, for those data blocks obsoleted because of using up of free blocks in read area or write area, their Meta data will be preserved in respective memory block, convenient to subsequent analysis of access pattern and capacity adjustment of read and write area. Data blocks obsoleted in read and write area will immediately be written into disk. When data is confirmed to be hot data, it can be written into SSD. Once hot data in disk is imported into SSD, Meta data zone which is mainly used to record Meta data of SSD needs to be updated, including the mapping relationship between data in SSD and data in disk. In addition, data consistency of disk and SSD needs to be recorded by Meta data zone.

In view of write speed of SSD and depletion to SSD caused by frequent writing, adopting hot data filtering mechanism can as far as possible avoid the case that cold data that has been stored in SSD directly is obsoleted into disk because of none access, causing pollution to SSD. Therefore, data that has been confirmed to be hot data through hot data filtering mechanism can be written into SSD.

3.2.2. Elimination Strategy. Through review of some currently mainstream cache replacement algorithm, class LRU

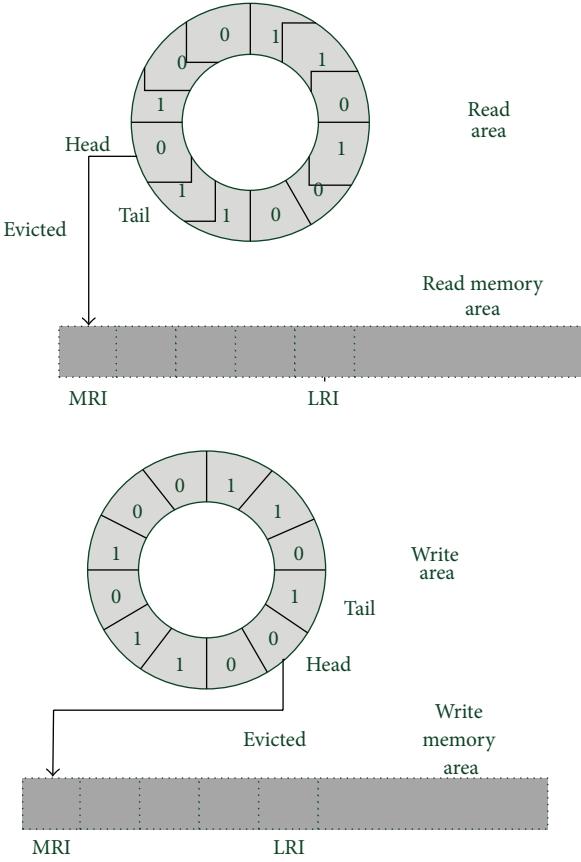


FIGURE 2: Data elimination in read-write area.

algorithm is suitable for most usage scenarios. However, LRU needs to record related access record terms, and each time these record terms are accessed, they need to be updated and large amount of extra space is also needed, and the cost of operation is great. But, by attentive analysis, according to the features of locality of reference, there is no need to record accurately the access situation of each data block but only to judge whether the data block in the window at the present has been accessed. In this way, cost of space and time can be significantly reduced. Therefore, we adopt CLOCK replacement algorithm.

The main idea of the algorithm is that an access bit is set in every page, and all pages in memory are linked into a circular queue through a link pointer. When one page is being accessed, its access bit will be set to 1.

When the replacement algorithm chooses one page to eliminate, it only needs to check the access bit of the page. If the bit is zero, then the page will be replaced; if the bit is one, then the bit will be set to zero again, and the page will not be replaced for the time being and will have the second chance to stay in memory. Then, the next page will be checked according to FIFO algorithm. Once it comes to the last page of the queue, if the access bit is still one, then it goes back to the head of queue to check the first page, as shown in Figure 2.

On the basis of hot and cold data selection algorithm, the cold data which were eliminated from read-write memory area would be directly written to disk instead of entering SSD.

Meta data of those are kept in cold data memory area which shares the feature with read-write memory area.

3.2.3. Dynamic Adjustment Mechanism of Read-Write Area. When an application write request is not required hitting from disk or SSD in the first reading is needed; there may be a large part because of the need to phase out dirty blocks to operate the disk writing, so that the missing of writing causes much more space. When reading misses, if the selection algorithm of thermal SSD data can select a large part of hot spot data, you can make most of the read requests completed in the SSD, apparently costing very little space. According to the current needs of read-write access mode and the characteristics of the system, the read-write in the memory buffer zone is dynamically adjusted to minimize the cost of accessing to the system, which makes the system applicable in the current scenario optimal.

Because large performance difference of flash media always causes cost difference of access, traditional method that evaluates replacement algorithm based on hit rate is not applicable anymore; instead, cost of access needs to be taken as the ultimate goal. FAC strategy divides the memory cache into read area and write area, and capacity of write area is 8 times larger than that of read area.

Shown as follows, the adjustment mechanism of read-write area based on access pattern is that, assuming that the total capacity of memory buffer is S , capacity of read area is $1/9 \times S$ and that of write area is $8/9 \times S$.

Considering that Meta data of blocks that have recently been eliminated is preserved in read-write memory area, when read request does not hits in read area but hits in read memory area, it shows that the data has been accessed not long ago, but the block was replaced by updated block because of the capacity of read area. Therefore, we take into account the increasing capacity of read area. Every time one unit length is added, capacity of write area will decrease by one unit length at the same time, keeping the total length invariant. It is in the same way for write request of application. Because capacity of write area is 8 times larger, there are two strategies for increasing write area. One is gradually changing, which means a minor increase every time. For example, every time two units length is added and the read area decreases the same length meanwhile. If the effect is not so obvious, increasing length will be 4 units length, 8 units length, and so forth. Different from gradual changing, leaping means that increasing length will be on the basis of the capacity rate between write area and read area, which is to say 8 units length will be added to write area each time. Leaping strategy can help achieve ideal effect rapidly; however, it also is easy to cause sharp decreasing of read area capacity and increasing of hit rate of read memory area, and the case cannot be improved immediately because of the restriction of expansion unit length of read area. We give some definitions in Table 1.

Expansion of read area:

$$\begin{aligned} RAC_{-}(t_i) &= RAC_{-}(t_{-}(i-1)) + P, \\ WAC_{-}(t_i) &= WAC_{-}(t_{-}(i-1)) - P. \end{aligned} \quad (1)$$

TABLE 1: Symbols and meaning.

Symbol	Meaning
P	Unit length
RAC	Read area capacity
WAC	Write area capacity
i_m	The minimum hot spot rank
N_b	The total access frequency
$HR(t)$	Hot spot ratio
HR_m	The minimum hot spot ratio
K_i	Time window
H_B	Heat counting
δ_i	Weight of K_i window
K	Access times of the current time window

Expansion of write area:

$$\begin{aligned} WAC_{-}(t_i) &= WAC_{-}(t_{-}(i-1)) + P', \\ RAC_{-}(t_i) &= RAC_{-}(t_{-}(i-1)) - P'. \end{aligned} \quad (2)$$

P' increases progressively from P to $2P$, $4P$, and $8P$:

$$\begin{aligned} WAC_{-}(t_i) &= WAC_{-}(t_{-}(i-1)) + 8P, \\ RAC_{-}(t_i) &= RAC_{-}(t_{-}(i-1)) - 8P. \end{aligned} \quad (3)$$

3.3. Design of SSD Cache Module

3.3.1. Selection Algorithm of Hot Data. First, let us see some existing research in this problem. Some researchers have put forward OHPM (short for object hot spot prediction model) to serve for hot spot data selection. In a storage system, N objects form a set S , in which objects are in order, and o_i means object ranking the i th place. The total access frequency of all objects is M and hot spot queue length is N_b . At a time T , assuming an object o_i in S is in hot spot queue, at t which equals k times of T , if o_i ranks the i' th place, then what is the value of i' ?

After some researches, the minimum hot spot rank can be counted by the following:

$$i_m = N_b k^{-1/\alpha}. \quad (4)$$

And hot spot ratio is confirmed by the following formula:

$$HR(t) = \frac{H(t)}{N_b}. \quad (5)$$

Further, the minimum hot spot ratio can be conducted by the previous formula and some other tests. Consider

$$HR_m = k^{-1/\alpha}. \quad (6)$$

Selecting cold and hot data is mainly judged by its access times; when it exceeds a specified threshold, the data will be judged as hot data, and the threshold can make dynamical adjustment. It is unnecessary to set a counter for every data block because of the huge cost. Through the observation on

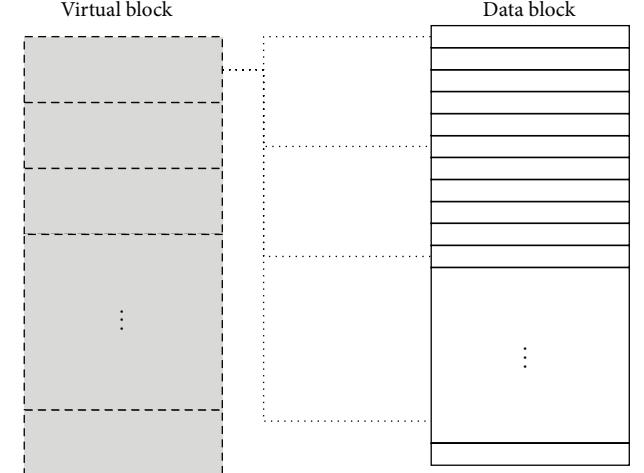


FIGURE 3: Multiple-one mapping.

the access trace of a large number of applications, only about 2% data can be judged as hot spot data in the absolute sense that will be accessed by a large number of applications. 4%–7% data is counted as hotter data because of relatively high access times. The rest of data has quite a little access time. On account of this point, we proposed selection algorithm of hot and cold data, Dynamical Adjust Section Hot data algorithm, called DASH, which is designed in two-layer structure. The first layer is filter layer, filtering blocks that do not achieve the threshold. Before filtering, multiple blocks are mapped into a large virtual data block, set with a counter as shown in Figure 3. For instance, 100 blocks are mapped into virtual data block, and those data blocks that have exceeded the filter value are qualified to enter the second layer of accurate statistics. For most filtered virtual data blocks, it is dispensable to count access times of each request page because the traffic of every virtual data block is less than the threshold. The second layer is statistics layer, counting accurately the access times of each small data block or request page and further judging whether it is hot data or not. In the procedure of implementation, the number of data blocks to make up a virtual data block depends on the characteristics of data stored in the current system. The less the amount of hot data is, the larger the request page selected should be; then, the space for counting hot data will be smaller. And many misjudgments caused by small request page in large data block, which will lead to entering the second layer of large amount of cold data, can be avoidable because of less hot data amount. On the contrary, in the case of plenty of hot data, selecting a hand of request pages to form large data block can avoid entering the second layer of large amount of cold data because of too many request pages in large data block and the counting cost of time.

The virtual blocks whose traffic exceeds filter value will generate an accurate statistical table, called Hash Table, which is used to record later access times of each request page in the virtual block as shown in Figure 4. The virtual block which has generated an accurate statistical table will enter the second layer to conduct accurate statistics, mainly, selecting a time window first and then counting the access times of

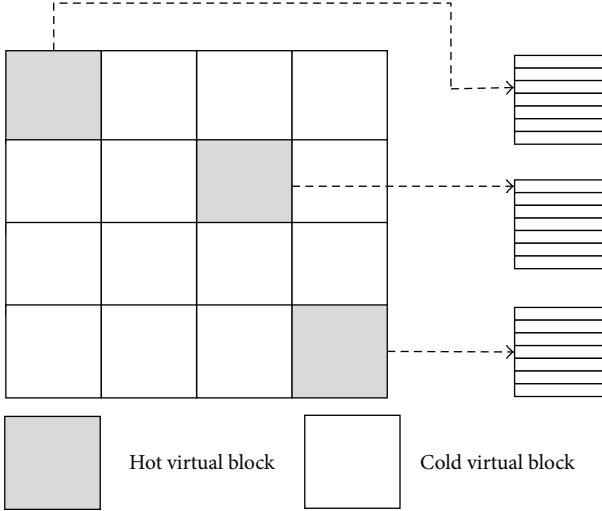


FIGURE 4: Multiple-one mapping.

each page in the time window. The time window will be divided into several smaller time windows such as K , for the characteristics of locality of reference. Access times weight of each time window is different; the closer to present time the time window is, the larger the weight is.

The heat of each request block in the whole time window can be counted according to the access times of request block and weight of each time window. Assuming that the weight of K_i window is δ_i , then heat counting formula of the current request page is

$$H_B = \sum_{i=1}^N K_i \delta_i. \quad (7)$$

For one data block, counting formula of its access times is

$$K_{T_i} = K + \delta K_{T_{i-1}}. \quad (8)$$

K refers to the access times of the current time window.

3.3.2. Hot Data Migration. In view of hot data updating with every data access, only when the data block accessed in disk is confirmed to be hot data before the next access and is marked as “ought to be in SSD,” it can be written into SSD. Thus, it is necessary to set a record table to record the mapping relation between data in SSD and that in disk and update the record table at the same time of updating data in SSD. Besides, because of write request of application not passing SSD and all data update completed in disk, which might lead to inconformity of data in disk and SSD, it is needful to set a flag bit to record whether the data in disk and SSD is consistent or not.

Before writing new hot data into SSD, we need to select one data block in SSD to eliminate. Considering that flag bit of data in SSD has been modified after hot data filtering, for those data blocks without being marked with “reserving,” they can be replaced. After modifying flag bit, the address of these data blocks can be recycled, similar to memory

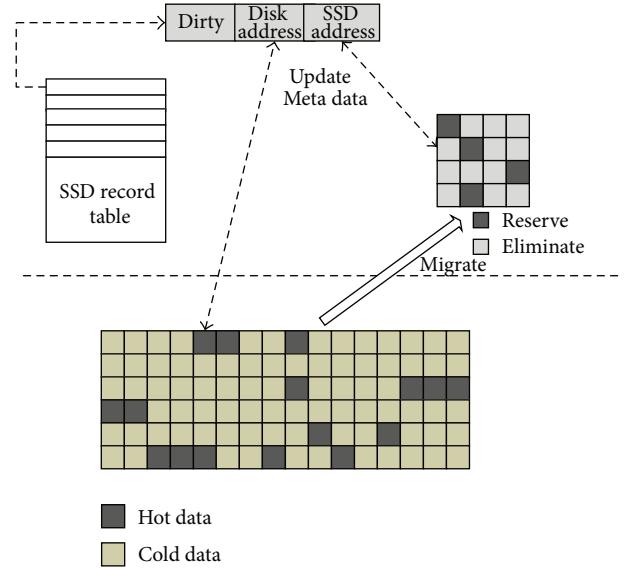


FIGURE 5: Hot data migration.

management mechanism. Forming a linked list with data blocks waiting to be eliminated, when elimination is needed, one data block in the list will be directly eliminated and SSD record table will be updated, as shown in Figure 5.

3.3.3. Dirty Data Migration. When it comes to the write request of application, those data not hit in memory cache will be directly read into memory from disk or SSD and then be modified. Before data page is read into memory, it will involve elimination in the case of that memory is full, and dirty block which is eliminated will be written into disk; thus, when writing data block including SSD copy, it leads to data inconformity of disk and SSD. So, dirty data migration mechanism is needed to ensure the consistency.

One policy is that, when modifying data including SSD copy, a flag bit is set to mark the data block, which can be only read from disk if it is dirty. That is to say, for those hot data that have been modified, SSD does not work and data in SSD has been invalid. Data marked as dirty block can be imported into SSD according to a particular policy, mainly guiding disk by time and according to the threshold. In the process of guiding disk by time, start guiding disk generator at set intervals, and update dirty data blocks in disk into SSD; meanwhile, update Meta data flag bit in SSD. In practical applications, frequency of guiding disk depends on the access features of the current system. If guiding disk is too frequent, it is likely to cause repetitive writing of SSD, reducing the service life of SSD. However, once the interval of guiding disk is too long, then maybe large amount of hot data needs to be accessed in disk, affecting the performance of the system. The other policy is guiding disk by threshold, meaning that the operation of guiding disk will be launched when dirty blocks in disk reach a certain quantity and write dirty data into disk without affecting disk. Magnitude of threshold is decided according to both the current access features and condition of loading.

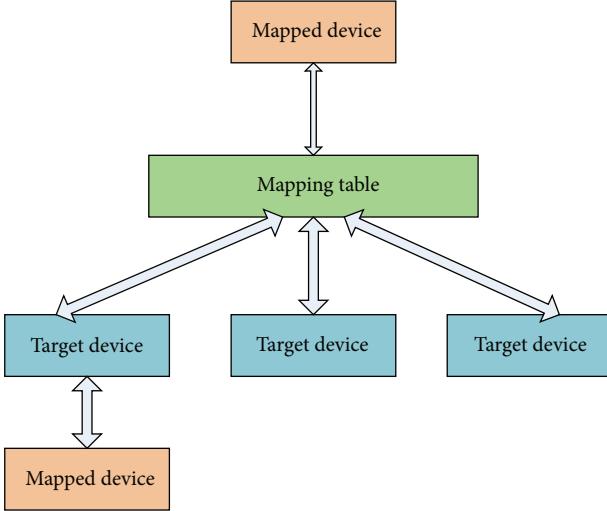


FIGURE 6: Device mapper mechanism.

Another strategy is that, if the coming write request hits in SSD, modify data in SSD directly. Compared with the previous strategy, the benefit of this strategy is to reduce system overhead and lower response time due to the faster write of SSD. Defects come with meeting frequent write, leading to multiple repetitive write and affecting the service life of SSD. Data modified in SSD, needs to be marked as “dirty” and guided into disk at regular time, enduring the consistency of data in disk and SSD.

4. Implement and Evaluation Methodology

4.1. Implements. FAC strategy designed in this paper based on SSD is implemented on the basis of flash cache. Flash cache is a new open source project of Facebook technical team, the fundamental purpose of which is using SSD to cache data to accelerate a kernel module of MySQL. It was originally used to accelerate the database and meanwhile was designed as general caching strategy, applicable to any applications built up on block device.

Based on Device Mapper mechanism, SSD and ordinary disk are mapped into a logical block device with cache, to be the interface of user action. Users perform read and write operations directly not on underlying SSD or ordinary disk but on this logical device. Which operating on these underlying block devices, caching function offered as an entirety will be lost.

Mapped device builds up mapping relations through mapping table and three target devices, and target device could be a single storage device or be evolved through mapped device and build up mapping relation through mapping table and the next layer target device as shown in Figure 6.

4.2. Experimental Setup. Cache system based on SSD works mainly on the existing storage server in laboratory, which is linked to server through client and tested with its read

TABLE 2: Server hardware environment.

CPU	Intel Xeon CPU E5606 @ 2.13 GHz 8 cores
RAM	DDR2 16 G
DISK	Seagate 7200 1 T
SSD	Intel SSD 320 Serials 120 G
OS	RHEL5.4 x86_64
KERNEL	Linux2.6.18-164

TABLE 3: Client machine environment.

CPU	Intel Pentium Dual-Core CPU E5300 @ 2.60 Hz
RAM	DDR2 2 G
DISK	Seagate 7200 500 G
OS	Windows 7 Ultimate 32 bit

and write performance in client-side. The environmental hardware configuration of server and client is provided in Tables 2 and 3.

The test mainly adopts Iometer to test the read-write performance of the system. Iometer is an I/O subsystem measurement and a characterization tool for single and clustered systems created by Intel Corporation, which can test performance of disk or network controller, capability of bandwidth transmission and response, network throughput of devices, and performance of hardware and network. Iometer primarily consists of Iometer and dynamo. Launch dynamo on the storage server first, link it to client machine, and then turn on Iometer application on client, and so we can see server-side host and disk to test in the case of different access patterns.

4.3. Performance Test Results

4.3.1. Read and Write Speed Comparison in Sequence. Performance test for caching strategy based on SSD on hybrid storage system is mainly to test read-write performance of system, capability of making dynamical adjustment based on access pattern, and depletion to SSD. We test depletion to SSD mainly by counting write times to SSD and judging and then comparing it with caching strategy of flash cache itself.

It can be seen from Figure 7, compared with included LRU caching algorithm of flash cache, in sequence, that FAC does not have performance advantage at the beginning, but its advantage of read-write performance has surfaced for making dynamical adjustment with the handing out of read-write requests. Furthermore, owing to SSD being mainly used as read cache in FAC caching strategy, it can be told from the figure that increasing read speed is larger and write speed do not decrease but suffer a certain growth.

While focusing on sequential read request, when block requests 8 KB, read speed of FAC has been obviously superior to LRU caching algorithm of flash cache, and with the processing of request handing, the advantage becomes more and more visible. When block requests 128 KB in sequential write request, read speed of FAC caching strategy starts to

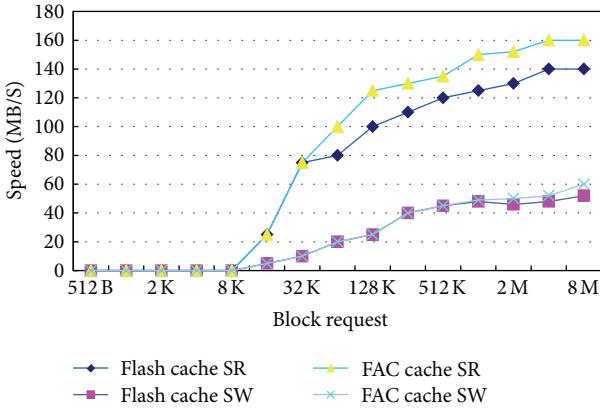


FIGURE 7: Read-write performance comparison between flash cache and FAC in sequence.

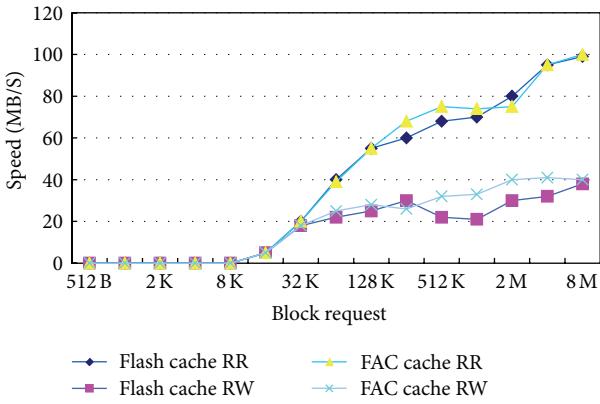


FIGURE 8: Read-write performance comparison between flash cache and FAC in random.

exceed that of flash cache and then superiority becomes larger and larger.

4.3.2. Read and Write Speed Comparison in Random. It can be seen from Figure 8, because of low cache hit rate in random, read-write performance of both flash cache LRU algorithm and FAC caching strategy is far lower than that in sequence. However, compared with flash cache LRU algorithm, FAC has a certain performance advantage in random even though the increasing is inferior to that in sequence.

It can be told from the figure that performance advantage of FAC caching strategy begins to appear from block size being 128 KB in case of random. Though its speed waves are slower than that of flash cache LRU algorithm when block size reaches 2 M, the overall trend of curve is still superior to flash cache LRU. In the case of random write, when block size is 32 KB, write speed of FAC caching strategy starts to exceed that of flash cache LRU algorithm and the overall trend is always superior to flash LRU algorithm.

The encoding efficiency with ICRS coding is 34.2% higher than using CRS and 56.5% more than using RS coding equally. The decoding rate by using ICRS is 18.1% higher than using CRS and 31.1% higher than using RS averagely.

5. Conclusions

FAC caching strategy based on SSD and cloud platform is designed in this paper. The strategy is combined with designing ideas of traditional caching strategy at the same time of giving full consideration to the characteristics of SSD itself, maximizing the role SSD plays. It can be seen from the test result that, benefit from the capability of making dynamical adjustment of FAC caching strategy and the advantage SSD has as read cache, the system can make dynamical adjustment according to the access pattern, greatly reduce write times to SSD, and prolong its service life without lowering read-write performance.

Caching strategy designing scheme designed in this paper is able to reduce the response time of system and the erase times to SSD. After that, considering that capability of making dynamical adjustment of memory cache would do little to make read-write of large disk qualitative change, it will be the next research target to combine memory cache and SSD better to improve accessing performance of disk.

So far, we have just implemented capability of making dynamical adjustment of memory cache module and hot data selection algorithm and data migration of SSD module, and the next target is to implement prefetching in system to improve performance.

References

- [1] B. S. Gill and L. A. D. Bathen, "Optimal multistream sequential prefetching in a shared cache," *ACM Transactions on Storage*, vol. 3, no. 3, article 10, 2007.
- [2] P. Cao and S. Irani, "Cost-aware WWW proxy caching algorithms," in *Proceedings of the USENIX Symposium on Internet Technology and Systems*, 1997.
- [3] L. A. Belady, "A study of replacement algorithms for a virtual-storage computer," *IBM Systems Journal*, vol. 5, no. 2, pp. 78–101, 1966.
- [4] C. Dirik and B. Jacob, "The performance of pc solid-state disks (SSDs) as a function of bandwidth, concurrency, device architecture, and system organization," in *Proceedings of the 36th Annual International Symposium on Computer Architecture (ISCA '09)*, pp. 279–289, June 2009.
- [5] L. Youyou, S. Jiwu, and Z. Weimin, "Extending the lifetime of flash-based storage through reducing write amplification from file systems," in *Proceedings of the 11th USENIX Conference on File and Storage Technologies (FAST '13)*, pp. 257–270, 2013.
- [6] T. Kgil and T. Mudge, "FlashCache: a NAND flash memory file cache for low power web servers," in *Proceedings of the International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES '06)*, pp. 103–112, October 2006.
- [7] G. Wu and X. He, "Reducing SSD read latency via NAND flash program and erase suspension," in *Proceedings of the 10th USENIX Conference on File and Storage Technologies*, 2012.
- [8] Y. Oh, E. Lee, D. Lee, J. Choi, and S. H. Noh, "Hybrid solid state drives for improved performance and enhanced lifetime," in *Proceedings of 29th IEEE Symposium on Massive Storage Systems*, 2013.
- [9] S. Lee, K. Ha, K. Zhang, J. Kim, J. Kim, and F. S. Flex, "A flexible flash file system for MLC NAND flash memory," in *Proceedings of the USENIX Annual Technical Conference*, San Diego, Calif, USA, June 2009.

- [10] N. Agrawal, V. Prabhakaran, T. Wobber, J. D. Davis, M. Manasse, and R. Panigrahy, "Design tradeoffs for SSD performance," in *Proceedings of the USENIX on Annual Technical Conference (ATC '08)*, 2008.
- [11] T. Kgil, D. Roberts, and T. Mudge, "Improving NAND flash based disk caches," in *Proceedings of the 35th International Symposium on Computer Architecture (ISCA '08)*, pp. 327–338, June 2008.
- [12] Y. Kim, A. Gupta, B. Urgaonkar, P. Berman, and A. Sivasubramaniam, "HybridStore: a cost-efficient, high-performance storage system combining SSDs and HDDs," in *Proceedings of the 19th Annual IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS '11)*, pp. 227–236, July 2011.
- [13] H. Jo, J.-U. Kang, S.-Y. Park, J.-S. Kim, and J. Lee, "FAB: flash-aware buffer management policy for portable media players," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 2, pp. 485–493, 2006.
- [14] S. Kang, S. Park, H. Jung, H. Shim, and J. Cha, "Performance trade-offs in using NVRAM write buffer for flash memory-based storage devices," *IEEE Transactions on Computers*, vol. 58, no. 6, pp. 744–758, 2009.
- [15] H. Kim and S. Ahn, "BPLRU: a buffer management scheme for improving random writes in flash storage abstract," in *Proceedings of 6th USENIX Conference on File and Storage Technologies*, pp. 158–171, ACM Press, San Jose, Calif, USA, 2008.
- [16] G. Wu, B. Eckart, and X. He, "BPAC: an adaptive write buffer management scheme for flash-based solid state drives," in *Proceedings of the IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST '10)*, IEEE Computer Society Press, Incline Village, Nevada, May 2010.
- [17] H. Zhao, P. Jin, P. Yang, and L. Yue, "BPCLC: an efficient write buffer management scheme for flash-based solid state disks," *International Journal of Digital Content Technology and its Applications*, vol. 4, no. 6, pp. 123–133, 2010.
- [18] S.-Y. Park, D. Jung, J.-U. Kang, J.-S. Kim, and J. Lee, "CFLRU: a replacement algorithm for flash memory," in *Proceedings of the International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES '06)*, pp. 234–241, ACM Press, October 2006.
- [19] H. Shim, B.-K. Seo, J.-S. Kim, and S. Maeng, "An adaptive partitioning scheme for DRAM-based cache in solid state drives," in *Proceedings of the IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST '10)*, May 2010.
- [20] T. Pritchett and M. Thottethodi, "SieveStore: a highly-selective, ensemble-level disk cache for cost-performance," in *Proceedings of the 37th International Symposium on Computer Architecture (ISCA '10)*, pp. 163–174, ACM Press, June 2010.
- [21] G. Soundararajan, V. Prabhakaran, M. Balakrishnan, and T. Wobber, "Extending SSD lifetimes with disk-based write caches," in *Proceedings of the 8th USENIX Conference on File and Storage Technologies (FAST '10)*, 2010.

Research Article

Opposition-Based Animal Migration Optimization

Yi Cao, Xiangtao Li, and Jianan Wang

School of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, China

Correspondence should be addressed to Yi Cao; caoy370@nenu.edu.cn

Received 25 July 2013; Accepted 8 September 2013

Academic Editor: William Guo

Copyright © 2013 Yi Cao 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.

AMO is a simple and efficient optimization algorithm which is inspired by animal migration behavior. However, as most optimization algorithms, it suffers from premature convergence and often falls into local optima. This paper presents an opposition-based AMO algorithm. It employs opposition-based learning for population initialization and evolution to enlarge the search space, accelerate convergence rate, and improve search ability. A set of well-known benchmark functions is employed for experimental verification, and the results show clearly that opposition-based learning can improve the performance of AMO.

1. Introduction

Many real world problems can be summarized as optimization problems. Optimization problems play an important role in both industrial application and scientific research. In the past decades, different optimization algorithms have been proposed. Among them, genetic algorithm may be the first and popular algorithm inspired by natural genetic variation and natural selection [1, 2]. Inspired by the social behavior of bird flocking or fish school, particle swarm algorithm was developed by Kennedy and Eberhart in 1995 [3, 4]. Artificial bee colony algorithm was proposed by Karaboga and Basturk in 2005, which simulates the foraging behavior of bee swarm [5, 6]. Ant colony optimization (ACO) which simulates the action of ants was first introduced by Dorigo [7, 8]. Animal Migration Optimization (AMO) as a new optimization algorithm inspired by animal migration behavior was first proposed by Li et al. in [9]. AMO simulates the widespread migration phenomenon in the animal kingdom, through the change of position, replacement of individual, and finding the optimal solution gradually. AMO has obtained good experimental results on many optimization problems.

Optimization algorithms often begin from an initial set of variables which are generated randomly and through iteration to obtain the global optimal solutions or the maximum of the objective function. It is known that the performance of these algorithms is highly related to diversity of particles; it may easily fall into local optima and has slow convergence rate and poor accuracy in the later stage of evolution. In

recent years, many efforts have been done to improve the performance of different algorithms.

The concept of opposition-based learning (OBL) was first introduced by Tizhoosh [10]. The main idea is to consider current candidate solution and its opposite candidate in order to enlarge the search scope, and it also uses elite selection mechanism to speed up the convergence speed and find the optimal solution. It has been proved in [11], an opposite learning mechanism has more chance to be closer to the global optimum solution than a random candidate solution. The opposition-based learning idea has been successfully applied on GA [11] that PSO [12–16], DE [17, 18], artificial neural networks (ANN) [19, 20], and ant colony optimization (ACO) [21]; experimental results show that opposition-based learning can improve the search capabilities of the algorithm to some extent.

This paper presents an algorithm to improve the performance of AMO. In the absence of priori information about the population, we employ opposition-based learning during population initialization and population evolution. Through the introduction of opposition-based learning mechanism, it can transform solutions from current search space to a new search space to enlarge the search space. By means of selecting the better solution between current solution and the opposite solution, it will improve search ability and accelerate convergence rate, and it has more chance to find the global optima.

The rest of this paper is organized as follows. Section 2 briefly introduces the animal migration optimization algorithm. Section 3 gives a simple description of

```

For i = 1 to NP
  For j = 1 to D
    If rand > Pa
       $x_{i,G+1} = x_{r1,G} + \text{rand} \cdot (x_{\text{best},G} - x_{i,G}) + \text{rand} \cdot (x_{r2,G} - x_{i,G})$ 
    End if
  End for
End for

```

ALGORITHM 1

opposition-based learning. Section 4 explains an implementation of the proposed algorithm, opposition-based AMO algorithm. Section 5 presents a comparative study among AMO, OPAMO, and other optimization algorithms on 23 benchmark problems. Finally, the work is concluded in Section 6.

2. Animal Migration Optimization

AMO is a new heuristic optimization algorithm inspired by the behavior of animal migration which is a ubiquitous phenomenon that can be found in all major animal groups, such as birds, mammals, fish, reptiles, amphibians, insects, and crustaceans [9].

In this algorithm, there are mainly two processes. In the first process the algorithm simulates how the groups of animals move from current position to a new position. During this process, each individual should obey three main rules: (1) move in the same direction as its neighbors; (2) remain close to its neighbors; (3) avoid collisions with its neighbors. We select one neighbor randomly and update the position of the individual according to this neighbor, as can be seen in the following formula:

$$x_{i,G+1} = x_{i,G} + \delta (x_{\text{neighborhood},G} - x_{i,G}), \quad (1)$$

where $x_{\text{neighborhood},G}$ is the current position of the neighborhood, δ is produced using a random number generator controlled by a Gaussian distribution, $x_{i,G}$ is the current position of i th individual, and $x_{i,G+1}$ is the new position of i th individual.

In the following process, the algorithm simulates how new animals are introduced to the group during the migration. During the population updating process, some animals will leave the group and some new animals will join the new population. We assume that the number of available animals is fixed. The animals will be replaced by some new individuals with a probability Pa . The probability is used according to the quality of the fitness. For the best fitness, the probability Pa is $1/NP$. For the worst fitness, the probability is 1. This process can be shown in Algorithm 1, where $r_1, r_2 \in [1, \dots, N_p]$ are randomly chosen integers and $r_1 \neq r_2 \neq i$. After producing a new solution, it will be evaluated and compared with $x_{i,G}$. If the objective fitness of $x_{i,G+1}$ is smaller than the fitness of $x_{i,G}$, $x_{i,G+1}$ is accepted as a new basic solution; otherwise, $x_{i,G}$ would be obtained.

To verify the performance of AMO, 23 benchmark functions chosen from the literature are employed. The results

show that the proposed algorithm clearly outperforms some evolution algorithms from the literature.

3. Opposition-Based Learning

In the evolutionary algorithm, algorithm often starts from random initial population until a satisfactory solution is found. If no population prior information is known, the speed of evolution is relation to the distance between the initial particles and the best particle. If we select some initial particles close to the best individual, we can accelerate the convergence of the algorithm to some extent.

The opposition learning algorithm is a new type of reinforcement learning algorithm, and it has been proven to be an effective concept to enhance various optimization approaches [22]. This algorithm utilizes opposition learning mechanism to generate opposite population and employs elite selection to choose the individual closer to the best individual as the member of initial population, thus facilitating the overall evolutionary convergence speed.

For ease of description, we give the definition of the opposition point first.

Definition 1. Let $x \in [a, b]$ be a real number; the opposite point x^* is defined as

$$x^* = a + b - x. \quad (2)$$

Similarly, the definition can be extended to high dimensional space.

Definition 2. Let $x = [x_1, x_2, \dots, x_D]$ as a point in D -dimension space, where $x_i \in [a_i, b_i]$, for all $i \in \{1, 2, 3, \dots, D\}$; the point $x^* = (x_1^*, x_2^*, \dots, x_D^*)$ is defined as the opposition value of x , in which

$$x_i^* = a_i + b_i - x_i. \quad (3)$$

Through the two previous definitions, we can conclude that the opposite learning algorithm is defined as follows.

Definition 3. $x = [x_1, x_2, \dots, x_D]$ is a candidate solution in D -dimension space; assume $f(x)$ is the fitness function which is used to evaluate the fitness of the candidate. According to the definition of the opposite point, $x^* = (x_1^*, x_2^*, \dots, x_D^*)$ is the opposite of x . If $f(x^*)$ is better than $f(x)$, then choose x^* as candidate instead of x ; otherwise, x is obtained, through evaluating the fitness of x and the opposite x^* to get the better individual.

```

(1) Begin
(2) Set the generation counter  $G = 0$ , and randomly initialize with a population of NP animal  $X_i$ .
(3) Evaluate the fitness for each individual  $X_i$ .
(4) For  $i = 1$  to NP do
(5)   For  $j = 1$  to D do
(6)      $OPX_{i,j} = k(a_j + b_j) - X_{i,j}$ 
(7)   End for
(8) Calculate the fitness value of  $OPX_i$ 
(9) End for
(10) Select NP fittest individual from  $\{X_i\} \cup \{OPX_i\}$  as an initial population;
(11) While stopping criteria is not satisfied do
(12)   For  $i = 1$  to NP do
(13)     For  $j = 1$  to D do
(14)        $X_{i,G+1} = X_{i,G} + \delta \cdot (X_{\text{neighborhood},G} - X_{i,G})$ 
(15)     End for
(16)     If  $\text{rand} < Pa$  then
(17)       For  $j = 1$  to D do
(18)          $OPX_{i,G+1} = k(a_j + b_j) - X_{i,G+1}$ 
(19)       End for
(20)     End for
(21)   Select NP fittest particles from  $\{X_G\} \cup \{X_{G+1}\} \cup \{OPX_{G+1}\}$  as current population;
(22)   For  $i = 1$  to NP
(23)     For  $j = 1$  to D
(24)       Select randomly  $r1 \neq r2 \neq i$ 
(25)       If  $\text{rand} > Pa$  then
(26)          $X_{i,G+1} = X_{r1,G} + \text{rand} \cdot (X_{\text{best},G} - X_{i,G}) + \text{rand} \cdot (X_{r2,G} - X_{i,G})$ 
(27)       End if
(28)     End for
(29)   For  $i = 1$  to NP do
(30)     Evaluate the offspring  $x_{i,G+1}$ 
(31)     If  $x_{i,G+1}$  is better than  $x_i$  then
(32)        $x_i = x_{i,G+1}$ 
(33)     End If
(34)   End for
(35) Memorize the best solution achieved so far
(36) End while
(37) End

```

ALGORITHM 2: Opposition-based AMO algorithm.

4. Enhanced AMO Using Opposition-Based Learning

4.1. Basic Concept. According to the probability principle, each randomly generated candidate solution has fifty percent probability chance away from the optimal solution compared to its opposition solution; AMO algorithm usually generates candidate solutions randomly in the search space as initial population because of the lack of prior information. During the optimization process, the calculation time of finding optimal solution will be changed according to the distance between the candidate solution and the optimal solution. By applying opposition-based learning, we not only evaluate the current candidate x but also calculate its opposition candidate x^* ; this will provide more chance of finding the solution closer to the global optimal value.

Let x be a solution in current search space, $x \in [a, b]$; the new solution x^* in the opposite space is

$$x^* = \Delta - x. \quad (4)$$

From the above definition, we can infer $x^* \in [\Delta - b, \Delta - a]$. Obviously, through the opposite transformation, the center of search space is changed from $(a + b)/2$ to $(2\Delta - a - b)/2$.

Let $\Delta = k(a + b)$, where k is a real number in $[0, 1]$, and it is often generated randomly; this denotes the probability of reverse learning.

4.2. Opposition-Based AMO Algorithm. In this paper, we introduce a novel algorithm which combines opposition learning algorithm with AMO algorithm. Through the introduction of opposition learning mechanism, we consider the candidate solution in both current search space and its opposite space simultaneously. By selecting the better individuals in the two search spaces, it can provide more chance to find the global optimal solution and largely speed up the convergence.

During the process of population initialization and evolution, through the introduction of opposition learning mechanism, more candidate solutions will be considered, and we choose the most likely candidate solutions for evolution.

TABLE 1: Benchmark functions based on our experimental study.

Test function	D	Range	Optimum	
$f_{01} = \sum_{i=1}^n x_i^2$	30	$[-100, 100]$	0	
$f_{02} = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$[-10, 10]$	0	
$f_{03} = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	$[-100, 100]$	0	
$f_{04} = \max_i \{ x_i , 1 \leq i \leq D\}$	30	$[-100, 100]$	0	
$f_{05} = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]$	0	
$f_{06} = \sum_{i=1}^D (\lfloor x_i + 0.5 \rfloor)^2$	30	$[-100, 100]$	0	
$f_{07} = \sum_{i=1}^D i x_i^4 + \text{random}[0, 1)$	30	$[-1.28, 1.28]$	0	
$f_{08} = \sum_{i=1}^D -x_i \sin(\sqrt{ x_i })$	30	$[-500, 500]$	-418.9829 * n	
$f_{09} = \sum_{i=1}^D [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]$	0	
$f_{10} = -20 \exp(-0.2 \sqrt{(1/D) \sum_{i=1}^D x_i^2}) - \exp((1/D) \sum_{i=1}^D \cos 2\pi x_i) + 20 + e$	30	$[-32, 32]$	0	
$f_{11} = (1/400) \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(x_i / \sqrt{i}) + 1$	30	$[-600, 600]$	0	
$f_{12} = (\pi/D) \{10\sin^2(\pi y_i) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10\sin^2(\pi y_i + 1)] + (yD - 1)^2 + \sum_{i=1}^D u(x_i, 10, 100, 4)\}$				
$y_i = 1 + ((x_i + 1) / 4)$	$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	$[-50, 50]$	0
$f_{13} = 0.1 \{10\sin^2(\pi y_i) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10\sin^2(\pi y_i + 1)] + (yD - 1)^2\} + \sum_{i=1}^D u(x_i, 10, 100, 4)$	30	$[-50, 50]$	0	
$f_{14} = [(1/500) + \sum_{j=1}^{25} (1 / (j + \sum_{i=1}^2 (x_i - a_{ij})^6))]^{-1}$	2	$[-65.53, 65.53]$	0.998004	
$f_{15} = \sum_{i=1}^{11} [a_i - x_1(b_i^2 + b_i x_i) / (b_i^2 + b_i x_3 + x_4)]^2$	4	$[-5, 5]$	0.0003075	
$f_{16} = 4x_1^2 - 2.1x_1^4 + (1/3)x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	-1.0316285	
$f_{17} = (x_2 - (5.1/4\pi^2)x_1^2 + (5/\pi)x_1 - 6)^2 + 10(1 - (1/8\pi)) \cos x_1 + 10$	2	$[-5, 10] * [0, 15]$	0.398	
$f_{18} = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$[-5, 5]$	3	
$f_{19} = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2)$	3	$[0, 1]$	-3.86	
$f_{20} = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2)$	6	$[0, 1]$	-3.32	
$f_{21} = -\sum_{i=1}^5 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.1532	
$f_{22} = -\sum_{i=1}^7 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.4029	
$f_{23} = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.5364	

We described the opposition-based population initial process in detail.

- (1) Initialize the population $P(N_p)$ in the search space $[a, b]$ randomly.
- (2) Calculate opposition population according to initial population; each dimension is calculated as follows:

$$\text{OP}_{ij} = k(a_j + b_j) - P_{ij}, \quad i = 1, 2, \dots, N_p; j = 1, 2, \dots, D, \quad (5)$$

where P_{ij} and OP_{ij} denote the j th variable of the i th vector of the population and opposite population, respectively, and k is a real number in $[0, 1]$ which denotes the probability of reverse learning.

- (3) Choose N_p fittest individual as initial population from the union of random population P and opposition population OP according to the value of fitness.

During the evolution process, we still adopt opposition learning method to increase the opportunity of finding the optimal solution. When a new individual is generated or joined, its opposition value is considered; if the fitness of opposite solution is better than the new individual, the opposite solution is adopted; otherwise, the new individual is obtained.

However, opposition learning method could not be suitable for all kinds of optimization problems. For instance, when calculating the opposition candidate, the solution may jump away from the solution space. If this happens,

TABLE 2: Minimization result of benchmark functions f_1-f_7 for different algorithms.

Function	Algorithm	PSO	DE	FA	GSA	ABC	AMO	OP-AMO
f_{01}	Mean	$3.3340e - 10$	$5.6016e - 14$	0.0017	$3.3748e - 18$	$2.9860e - 20$	$8.6464e - 40$	$1.4222E - 61$
	Rank	6	5	7	4	3	2	1
f_{02}	Mean	$6.6598e - 11$	$4.7348e - 10$	0.0453	$8.92115e - 09$	$1.4213e - 15$	$8.2334e - 32$	$1.5342E - 40$
	Rank	4	5	7	6	3	2	1
f_{03}	Mean	2.9847	$2.8038e - 11$	0.0182	0.1126	$2.4027e + 03$	$8.8904e - 04$	$1.0303E - 60$
	Rank	6	2	4	5	8	3	1
f_{04}	Mean	7.9997	0.2216	0.0554	$9.9302e - 10$	18.5227	$2.8622e - 05$	0.6132
	Rank	6	4	3	1	7	2	5
f_{05}	Mean	46.9202	0.2657	38.1248	20.0819	0.0441	4.1817	$2.9114E - 06$
	Rank	7	3	6	5	2	4	1
f_{06}	Mean	$3.6925e - 10$	$4.5028e - 14$	0.0017	$3.3385e - 18$	$3.0884e - 20$	0	0
	Rank	6	5	7	4	3	1	1
f_{07}	Mean	0.0135	0.0042	0.0082	0.0039	0.0324	0.0017	0.0004
	Rank	6	4	5	3	7	2	1
Average rank		5.86	4	5.57	4	4.71	2.28	1.57
Overall rank		7	3	6	3	5	2	1

TABLE 3: Minimization result of benchmark functions f_8-f_{13} for different algorithms.

Function	Algorithm	PSO	DE	FA	GSA	ABC	AMO	OP-AMO
f_8	Mean	$-8.8278e + 03$	$-1.1276e + 04$	$-6.2238e + 03$	$-3.0499e + 03$	$-1.2507e + 04$	$-1.2569e + 04$	$-1.1562E + 04$
	Rank	5	4	6	7	3	2	1
f_9	Mean	18.2675	134.6789	23.5213	7.2831	0	0	0
	Rank	5	8	6	4	1	1	1
f_{10}	Mean	$3.8719e - 06$	$7.4739e - 08$	0.0094	$1.4717e - 09$	$1.1946e - 09$	$4.4409e - 15$	1.7183
	Rank	5	4	6	3	2	1	7
f_{11}	Mean	0.0168	0	0.0025	0.01265	0	0	1
	Rank	6	1	4	5	1	1	7
f_{12}	Mean	0.0083	$4.7114e - 15$	$8.8694e - 06$	$2.0358e - 20$	$1.1928e - 21$	$1.5705e - 32$	$1.5704E - 32$
	Rank	7	5	6	4	3	2	1
f_{13}	Mean	$4.6694e - 07$	$3.1598e - 14$	$1.2812e - 04$	$5.6991e - 33$	$2.2990e - 20$	$1.3498e - 32$	$1.3496E - 32$
	Rank	6	5	7	1	4	3	2
Average rank		5.67	4.5	5.83	4	2.33	1.67	3.17
Overall rank		6	4	5	3	2	1	3

the solution will be invalid; to avoid this case, the transformed candidate is assigned to a random value as follows:

$$x^* = \text{rand}(a, b) \quad \text{if } x < a \text{ || } x > b, \quad (6)$$

where $\text{rand}(a, b)$ is a random number between a and b .

The opposition learning based AMO algorithm is described in Algorithm 2.

5. Experimental Results

To evaluate the performance of our algorithm, we applied it to 23 standards benchmark functions as shown in Table 1. These functions have been widely used in the literature.

The maximum numbers of generations are 1500 for f_1 , f_6 , f_{10} , f_{12} , and f_{13} , 2000 for f_2 and f_{11} , 3000 for f_7 , f_8 , and f_9 , and 5000 for f_3 , f_4 , and f_5 , 400 for f_{15} , 100 for f_{14} , f_{16} , f_{17} , f_{19} , f_{21} , f_{22} , and f_{23} , 30 for f_{18} , and 200 for f_{20} .

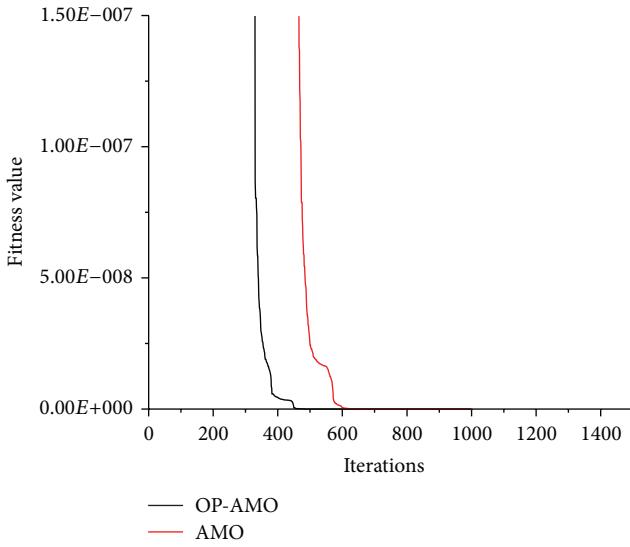
The population size is 50 because this algorithm has two phases. The results of the algorithm on the 23 test problems are presented in Tables 2, 3, and 4.

As seen from Table 2 to Table 4, opposition-based AMO algorithm significantly improves the results on functions f_1 , f_2 , f_3 , f_5 , f_6 , and f_7 and achieves the same performance with AMO on most other functions, and it is better than PSO, DE, ABC, and FA. This denotes that the algorithm can accelerate convergence rate and find better solutions for some optimization problems, and it did not lead to the premature of algorithm.

Figure 1 shows the evolutionary process of test function f_2 by AMO and opposition learning based AMO. The horizontal ordinate denotes the evolution iterations, and the vertical ordinate is the optimal value of objective function. We can see from the graph that the convergence rate of opposition-based AMO is obviously accelerated and the accuracy of optimization is also enhanced.

TABLE 4: Minimization result of benchmark functions f_{14} – f_{23} for different algorithms.

Function	Algorithm	PSO	DE	FA	GSA	ABC	AMO	OP-AMO
F_{14}	Mean	0.9980	0.9980	3.0273	5.9533	0.9980	0.9980	0.9980
	Rank	3	2	6	7	1	4	4
F_{15}	Mean	$6.5867e - 04$	$4.5400e - 04$	0.0010	0.0048	$7.4715e - 04$	$3.9738e - 04$	$3.0748e - 04$
	Rank	4	3	6	7	5	2	1
F_{16}	Mean	-1.0316	-1.0316	-1.0314	-1.03163	-1.0316	-1.0316	-1.0316
	Rank	4	3	7	1	2	5	5
f_{17}	Mean	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979
	Rank	5	1	6	1	7	1	1
f_{18}	Mean	3.0001	3.0000	3.0123	3.7403	3.0000	3.0018	2.9999
	Rank	4	2	5	7	2	6	1
f_{19}	Mean	-3.8628	-3.8628	-3.8613	-3.8625	-3.8628	-3.8628	-3.8629
	Rank	4	3	8	7	4	2	1
F_{20}	Mean	-3.2554	-3.2174	-3.2741	-3.3220	-3.3220	-3.3220	-3.3223
	Rank	6	7	5	1	2	3	4
F_{21}	Mean	-7.6393	-10.1532	-6.5633	-4.784	-10.1528	-10.0592	-10.1532
	Rank	5	1	6	7	3	4	1
F_{22}	Mean	-7.3602	-10.4029	-10.4027	-6.5797	-10.4012	-10.3899	-10.4029
	Rank	6	1	3	7	4	6	1
F_{23}	Mean	-8.9611	-10.5364	-10.2297	-8.2651	-10.5339	-10.4990	-10.5364
	Rank	5	1	6	7	3	4	1
Average rank		4.6	2.4	5.8	5.2	3.3	3.7	2
Overall rank		5	2	7	6	3	4	1

FIGURE 1: Evolutionary process of the two algorithms on function f_2 .

6. Conclusions

A novel opposition-based AMO algorithm is proposed in this paper. This approach can provide more chance to find better solutions by transforming candidate solutions from current search space to a new search space. Experimental results show that, compared with previous AMO, the proposed algorithm

is efficient in most of the test functions. However, we can also see from the experimental results that this algorithm is not suitable for all kinds of problems; for some optimization problems, algorithm has no significant improvement. How to improve the algorithm to adapt to a more optimization problem is worth of further study.

Conflict of Interests

The authors declare that they do not have any commercial or associative interest that represents a conflict of interests in connection with the work submitted.

References

- [1] M. Melanie, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, Mass, USA, 1999.
- [2] S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms*, Springer, Berlin, 2008.
- [3] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 14, pp. 1942–1948, December 1995.
- [4] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons, Hoboken, NJ, USA, 2005.
- [5] 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.
- [6] D. Karaboga and B. Basturk, “On the performance of artificial bee colony (ABC) algorithm,” *Applied Soft Computing Journal*, vol. 8, no. 1, pp. 687–697, 2008.

- [7] M. Dorigo, *Optimization, learning and natural algorithms [Ph.D. dissertation]*, Politecnico di Milano, Milano, Italy, 1992.
- [8] M.-H. Lin, J.-F. Tsai, and L.-Y. Lee, "Ant colony optimization for social utility maximization in a multiuser communication system," *Mathematical Problems in Engineering*, vol. 2013, Article ID 798631, 8 pages, 2013.
- [9] X. Li, J. Zhang, and M. Yin, "Animal migration optimization: an optimization algorithm inspired by animal migration behavior," *Neural Computing and Applications*, 2013.
- [10] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in *International Conference on Computational Intelligence for Modelling, Control and Automation (CIMCA '05)*, pp. 695–701, November 2005.
- [11] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition versus randomness in soft computing techniques," *Applied Soft Computing Journal*, vol. 8, no. 2, pp. 906–918, 2008.
- [12] Z. Lin and L. Wang, "A new opposition-based compact genetic algorithm with fluctuation," *Journal of Computational Information Systems*, vol. 6, no. 3, pp. 897–904, 2010.
- [13] H. Lin and H. Xingshi, "A novel opposition-based particle swarm optimization for noisy problems," in *Proceedings of the 3rd International Conference on Natural Computation (ICNC '07)*, pp. 624–629, Haikou, China, August 2007.
- [14] H. Wang, H. Li, Y. Liu, C. Li, and S. Zeng, "Opposition-based particle swarm algorithm with Cauchy mutation," in *Proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 4750–4756, Singapore, September 2007.
- [15] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, and M. Ventresca, "Enhancing particle swarm optimization using generalized opposition-based learning," *Information Sciences*, vol. 181, no. 20, pp. 4699–4714, 2011.
- [16] H. Wang, Z. Wu, S. Rahnamayan, and J. Wang, "Diversity analysis of opposition-based differential evolution—an experimental study," in *Proceedings of the International Symposium on Intelligence Computation and Applications*, pp. 95–102, 2010.
- [17] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution algorithms," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 2010–2017, Vancouver, Canada, July 2006.
- [18] R. S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64–79, 2008.
- [19] M. Ventresca and H. R. Tizhoosh, "Improving the convergence of backpropagation by opposite transfer functions," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '06)*, pp. 4777–4784, Vancouver, Canada, July 2006.
- [20] M. Ventresca and H. R. Tizhoosh, "Opposite transfer functions and backpropagation through time," in *Proceedings of the IEEE Symposium on Foundations of Computational Intelligence (FOCI '07)*, pp. 570–577, Honolulu, Hawaii, USA, April 2007.
- [21] A. R. Malisia and H. R. Tizhoosh, "Applying opposition-based ideas to the Ant Colony System," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '07)*, pp. 182–189, Honolulu, Hawaii, USA, April 2007.
- [22] W.-f. Gao, S.-y. Liu, and L.-l. Huang, "Particle swarm optimization with chaotic opposition-based population initialization and stochastic search technique," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 11, pp. 4316–4327, 2012.

Research Article

Time-Domain Joint Parameter Estimation of Chirp Signal Based on SVR

Xueqian Liu and Hongyi Yu

Zhengzhou Information Science and Technology Institute, Zhengzhou 450002, China

Correspondence should be addressed to Xueqian Liu; liuxqpaper@tom.com

Received 12 July 2013; Revised 7 August 2013; Accepted 29 August 2013

Academic Editor: Zhiqiang Ma

Copyright © 2013 X. Liu and H. Yu. 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.

Parameter estimation of chirp signal, such as instantaneous frequency (IF), instantaneous frequency rate (IFR), and initial phase (IP), arises in many applications of signal processing. During the phase-based parameter estimation, a phase unwrapping process is needed to recover the phase information correctly and impact the estimation performance remarkably. Therefore, we introduce support vector regression (SVR) to predict the variation trend of instantaneous phase and unwrap phases efficiently. Even though with that being the case, errors still exist in phase unwrapping process because of its ambiguous phase characteristic. Furthermore, we propose an SVR-based joint estimation algorithm and make it immune to these error phases by means of setting the SVR's parameters properly. Our results show that, compared with the other three algorithms of chirp signal, not only does the proposed one maintain quality capabilities at low frequencies, but also improves accuracy at high frequencies and decreases the impact with the initial phase.

1. Introduction

Chirp signals, that is, second-order polynomial phase signals, are common in various areas of science and engineering. For example, in a synthetic aperture radar (SAR) system, when the target is moving, the regulated signals will change into chirp ones after being reflected [1]. The SAR imaging quality may be degraded by shifting and/or defocusing due to inaccurate estimation of instantaneous frequency (IF) and instantaneous frequency rate (IFR), respectively [2]. In optical communications, coding or instability of the laser diode results in chirp phenomenon [3]. The research on estimating these chirp parameters is divided into two parts: one is based on cubic phase function (CPF), even high-order phase function (HPF) [4–6] and the other is based on maximum likelihood (ML) [7–9]. The former has the advantage in fast calculation of IFR, but costs more to find IF. The latter tries its best to maximize an ML function mainly in frequency domain, which involves a two-dimensional nonlinear optimization. Unfortunately, there is no exact, closed-form solution for solving this general nonlinear programming problem. The solution either resorts to a burdensome numerical search or is approximated by

linearized techniques. Djuric and Kay proposed an efficient time-domain ML estimator [10], which can achieve optimum estimation at a moderate complexity. By extending the phase noise model of [11] to chirp signals, Li et al. derived an improved ML estimator and analyzed its performance in the time domain [12, 13]. However, both of them are suitable only at high signal-to-noise ratio (SNR), for the reason of approximations of noise phase model and imperfections in phase unwrapping process.

By introducing structural risk minimization (SRM) principle, support vector regression (SVR) exhibits excellent capabilities for generalizing and learning. From the viewpoint of the quadratic relationship between absolute signal phase and time series, therefore, this paper employs SVR to unwrap phases and estimate IF and IFR recursively. We avoid taking the rationality of phase noise model into account by not making any approximations in it. At one time, we reduce the estimation performance's dependence on phase unwrapping process with a proper choice of SVR's parameters. It has been shown that, except for the property of low sensitivity to initial phase and closely approaching the Cramer-Rao lower bound (CRLB) at low frequencies, the proposed algorithm improves its estimation performance at high frequencies.

2. The Proposed Algorithm

2.1. Signal Model. The signal model used here is similar to that in [10, 12]. The complex, baseband chirp signal polluted by noise is modelled as

$$\begin{aligned} r_n &= a_n A \exp \left\{ j \left[\phi + 2\pi f_d n T_s + 2\pi f_r (n T_s)^2 \right] \right\} + w_n, \\ n &= 0, \dots, N-1. \end{aligned} \quad (1)$$

Here, a_n is an independent symbol; $A > 0$, f_d , $f_r \in [-0.5, 0.5]$, and $\phi \in [-\pi, \pi]$ are the amplitude, deterministic but unknown IF, IFR, and initial phase, respectively; T_s is the sampling period; N is the sample size; w_n is an independent complex additive white Gaussian noise (AWGN) with zero mean and variance σ^2 . For the sake of simplicity, we set $T_s = 1$ and rearrange (1):

$$\begin{aligned} r_n &= \exp \left[j(\phi + 2\pi f_d n + 2\pi f_r n^2) \right] \\ &\times \left\{ A + a_n^* \exp \left[-j(\phi + 2\pi f_d n + 2\pi f_r n^2) \right] w_n \right\} \\ &= \exp \left[j(\phi + 2\pi f_d n + 2\pi f_r n^2) \right] (A + w'_n), \\ n &= 0, \dots, N-1, \end{aligned} \quad (2)$$

where $w'_n = a_n^* \exp[-j(\phi + 2\pi f_d n + 2\pi f_r n^2)] w_n$ is still an independent complex AWGN with zero mean and variance σ^2 . Absolute phase of r_n is presented as

$$\angle r_n = \phi + 2\pi f_d n + 2\pi f_r n^2 + \angle(A + w'_n), \quad n = 0, \dots, N-1. \quad (3)$$

When SNR = A^2/σ^2 is large enough, the approximate model in [10] is expressed as

$$\begin{aligned} \angle(A + w'_n) &\approx \arctan \frac{\text{Im}(w'_n)}{A + \text{Re}(w'_n)} \\ &\approx \arctan \frac{\text{Im}(w'_n)}{A} \\ &\approx \frac{\text{Im}(w'_n)}{|r_n|}, \quad n = 0, \dots, N-1. \end{aligned} \quad (4)$$

A more reasonable model proposed in [12] is given:

$$\begin{aligned} \angle(A + w'_n) &\approx \arcsin \frac{\text{Im}(w'_n)}{|r_n|} \\ &\approx \frac{\text{Im}(w'_n)}{|r_n|}, \quad n = 0, \dots, N-1. \end{aligned} \quad (5)$$

2.2. SVR-Based Phase Unwrapping Process and Frequency Estimation Algorithm. In (5), there are two approximations still existing. Based on the quadratic relation in (3), this paper utilizes SVR's excellent capability for learning unknown models to unwrap phases and estimate frequency. We yield a

training set $S_k = \{(x_i^k, y_i^k) \mid i = 1, \dots, k\}$, $x_i^k = i - 1$, $y_i^k = \angle \hat{r}_{i-1}$ at time point k ($1 \leq k \leq N$), where $\angle \hat{r}_{i-1}$ denotes the estimation value of $\angle r_{i-1}$.

At first, we present a line $f_k(x) = (\mathbf{w}_k \cdot \phi(x)) + b_k$, where (\cdot) is an inner product operator and $\phi(\cdot)$ is a nonlinear mapping from low to high dimension feature space. And also, we define its ε -insensitive loss function as

$$\begin{aligned} L(y_i^k, f_k(x_i^k)) &= |y_i^k - f_k(x_i^k)|_\varepsilon \\ &= \begin{cases} 0, & |y_i^k - f_k(x_i^k)| \leq \varepsilon_k, \\ |y_i^k - f_k(x_i^k)| - \varepsilon_k, & \text{else,} \end{cases} \end{aligned} \quad (6)$$

where ε_k is insensitive loss coefficient at time point k .

Next, we assume $f_k(x)$ insofar as for ε_k to completely fit all elements of S_k , and we denote d_i^k as the distance from point $(x_i^k, y_i^k) \in S_k$ to $f_k(x)$:

$$\begin{aligned} d_i^k &= \frac{|(\mathbf{w}_k \cdot \phi(x_i^k)) + b_k - y_i^k|}{\sqrt{1 + \|\mathbf{w}_k\|^2}} \\ &\leq \frac{\varepsilon_k}{\sqrt{1 + \|\mathbf{w}_k\|^2}}, \quad i = 1, \dots, k. \end{aligned} \quad (7)$$

According to (7), we optimize $f_k(x)$ through maximizing $\varepsilon_k / \sqrt{1 + \|\mathbf{w}_k\|^2}$, that is, minimizing $\|\mathbf{w}_k\|^2$. Thereby, SVR is presented as

$$\begin{aligned} \min J(\mathbf{w}_k, b_k) &= \frac{1}{2} \|\mathbf{w}_k\|^2 \\ \text{s.t.} \quad &|(\mathbf{w}_k \cdot \phi(x_i^k)) + b_k - y_i^k| \leq \varepsilon_k, \quad i = 1, \dots, k. \end{aligned} \quad (8)$$

In fact, fitting errors larger than ε_k always exist. By introducing slack variables $\xi_i^k, (\xi^*)_i^k \geq 0$, $i = 1, \dots, k$, and penalty factor C_k at time point k , (8) is converted into

$$\begin{aligned} \min J(\mathbf{w}_k, b_k, \xi_k, \xi_k^*) &= \frac{1}{2} \|\mathbf{w}_k\|^2 + C_k \sum_{i=1}^k [\xi_i^k + (\xi^*)_i^k] \\ \text{s.t.} \quad &\begin{cases} y_i^k - (\mathbf{w}_k \cdot \phi(x_i^k)) - b_k \leq \varepsilon_k + \xi_i^k \\ (\mathbf{w}_k \cdot \phi(x_i^k)) + b_k - y_i^k \leq \varepsilon_k + (\xi^*)_i^k, \end{cases} \quad i = 1, \dots, k, \end{aligned} \quad (9)$$

where $\xi_k = [\xi_1^k, \xi_2^k, \dots, \xi_k^k]^T$, $\xi_k^* = [(\xi^*)_1^k, (\xi^*)_2^k, \dots, (\xi^*)_k^k]^T$, $[\cdot]^T$ is a transpose operator, and C is a positive constant to take compromise in SVR's generalization capability and fitting errors which are denoted by the first and second items of $J(\mathbf{w}_k, b_k)$, respectively.

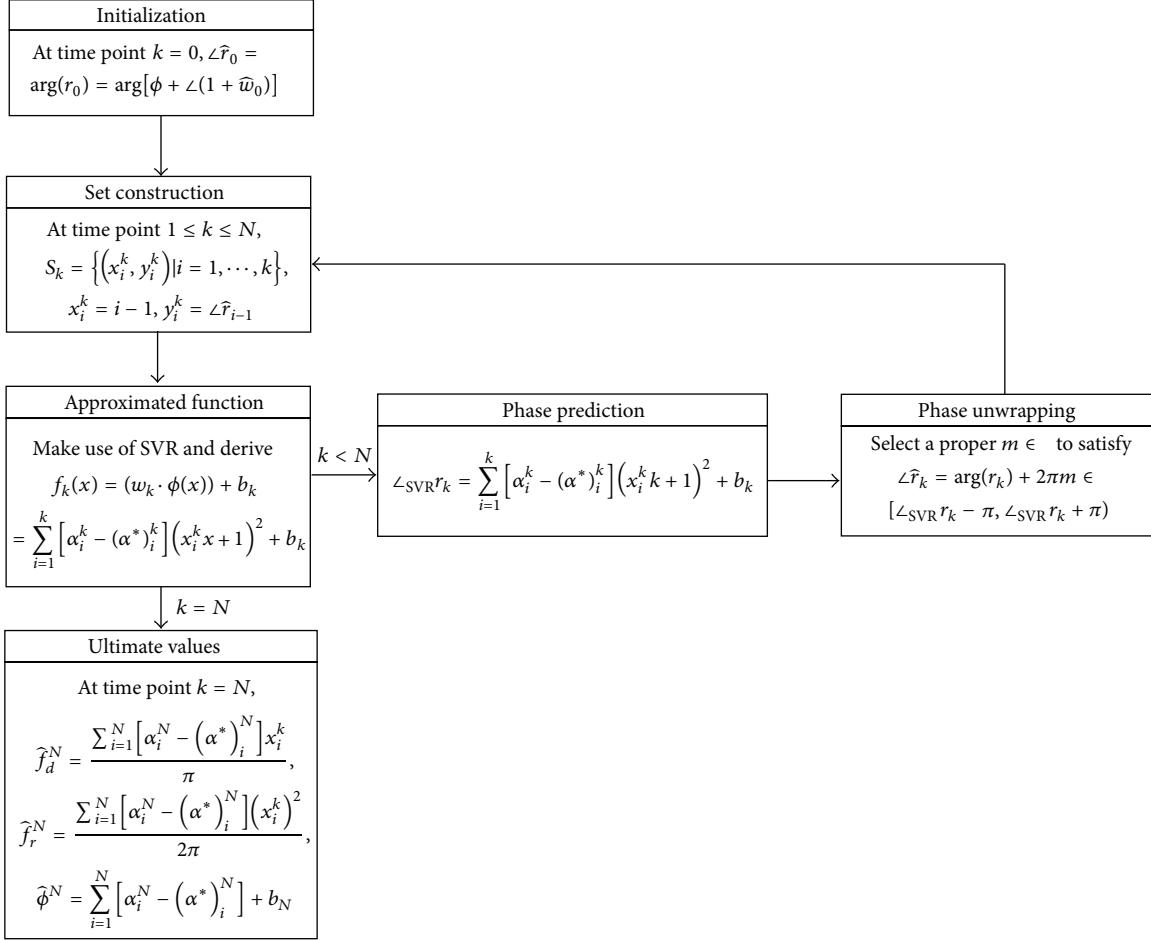


FIGURE 1: Recursive implementation of SVR-based algorithm.

Equation (9) is a strict convex quadratic programming (QP) problem in optimal theories. Then, using Lagrange multiplier method:

$$\begin{aligned}
 \mathbf{w}_k &= \sum_{i=1}^k [\alpha_i^k - (\alpha^*)_i^k] \phi(x_i^k), \\
 \sum_{i=1}^k [(\alpha^*)_i^k - \alpha_i^k] &= 0, \\
 C_k - \alpha_i^k - \gamma_i^k &= 0, \quad i = 1, \dots, k, \\
 C_k - (\alpha^*)_i^k - (\gamma^*)_i^k &= 0, \quad i = 1, \dots, k,
 \end{aligned} \tag{10}$$

where α_i^k , $(\alpha^*)_i^k$, γ_i^k , and $(\gamma^*)_i^k$ are the i th Lagrange multipliers of $y_i^k - (\mathbf{w}_k \cdot \phi(x_i^k)) - b_k \leq \varepsilon_k + \xi_i^k$, $(\mathbf{w}_k \cdot \phi(x_i^k)) + b_k - y_i^k \leq \varepsilon_k + (\xi^*)_i^k$, $\xi_i \geq 0$, and $(\xi^*)_i^k \geq 0$, respectively.

Substituting (10) into (9), replacing $(\phi(x_i^k) \cdot \phi(x_j^k))$ with quadratic kernel function $K(x_i, x_j) = (x_i^* x_j + 1)^2$ in this study, and deriving the wolf dual problem of (9):

$$\begin{aligned}
 &\max W(\boldsymbol{\alpha}_k, \boldsymbol{\alpha}_k^*) \\
 &= -\frac{1}{2} \sum_{i,j=1}^k [\alpha_i^k - (\alpha^*)_i^k] [\alpha_j^k - (\alpha^*)_j^k] (x_i^* x_j + 1)^2 \\
 &\quad + \sum_{i=1}^k y_i^k [\alpha_i^k - (\alpha^*)_i^k] - \varepsilon_k \sum_{i=1}^k [\alpha_i^k + (\alpha^*)_i^k] \\
 &\text{s.t. } \sum_{i=1}^k [\alpha_i^k - (\alpha^*)_i^k] = 0 \\
 &\quad 0 \leq \alpha_i^k, \quad (\alpha^*)_i^k \leq C_k, \quad i = 1, \dots, k,
 \end{aligned} \tag{11}$$

where $\boldsymbol{\alpha}_k = [\alpha_1^k, \alpha_2^k, \dots, \alpha_k^k]^T$, $\boldsymbol{\alpha}_k^* = [(\alpha^*)_1^k, (\alpha^*)_2^k, \dots, (\alpha^*)_k^k]^T$.

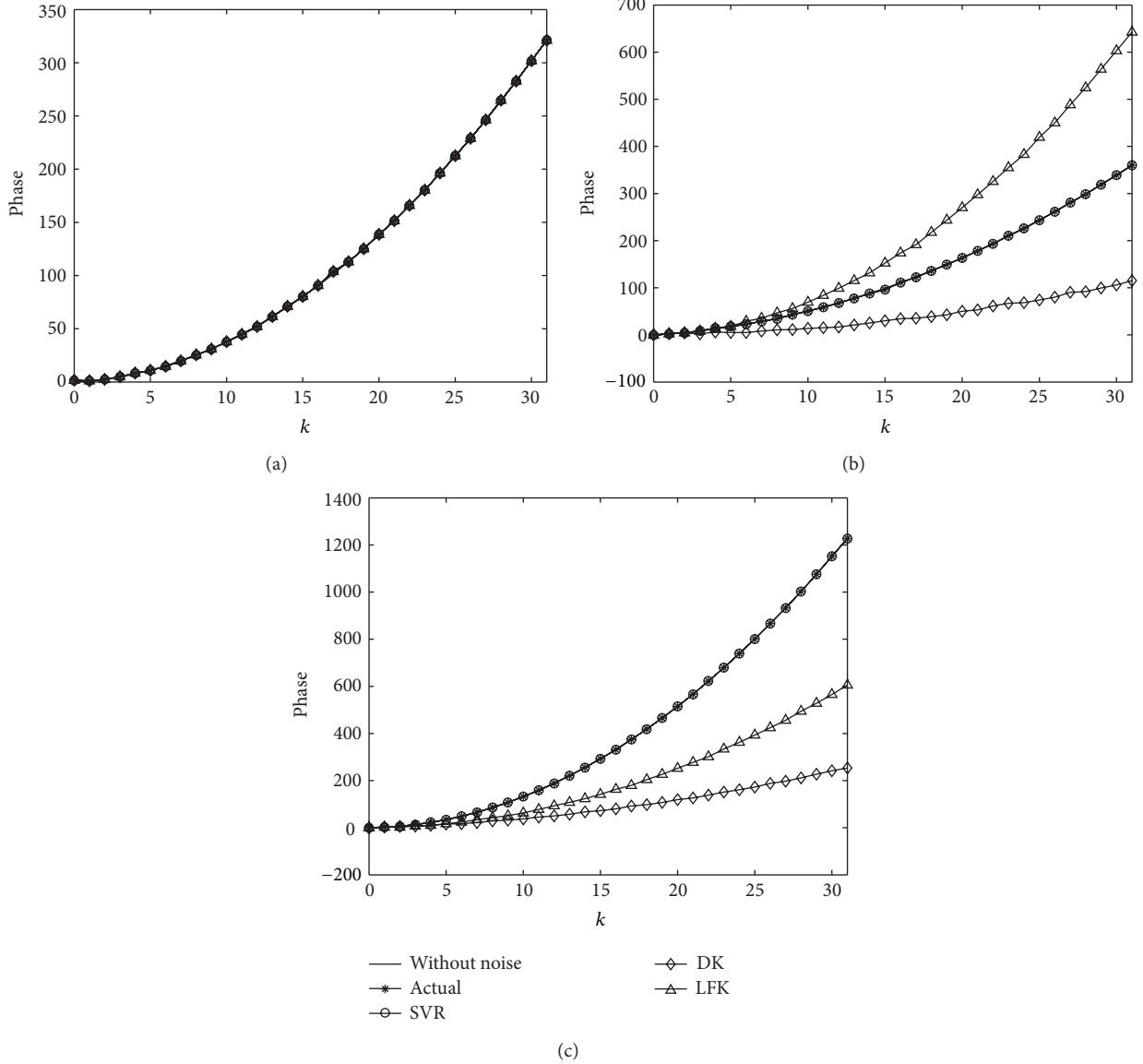


FIGURE 2: (a) Arbitrary phase unwrapping processes with $f_d = 0.1$, $f_r = 0.05$, $\phi = 0$, $N = 32$, and SNR = 8 dB. (b) Arbitrary phase unwrapping processes with $f_d = 0.3$, $f_r = 0.05$, $\phi = 0$, $N = 32$, and SNR = 8 dB. (c) Arbitrary phase unwrapping processes with $f_d = 0.1$, $f_r = 0.2$, $\phi = 0$, $N = 32$, and SNR = 8 dB.

We obtain α_k , α_k^* through solving (11). Ultimately, we get the best approximation of (3) by using Karush-Kuhn-Tucker (KKT) conditions:

$$f_k(x) = (\mathbf{w}_k \cdot \phi(x)) + b_k \\ = \sum_{i=1}^k \left[\alpha_i^k - (\alpha^*)_i^k \right] (x_i^* x_j + 1)^2 + b_k, \quad (12)$$

where $b_k = y_i^k - \sum_{j=1}^k [\alpha_j^k - (\alpha^*)_j^k] (x_i^* x_j + 1)^2 - \varepsilon_k$, $\alpha_i^k \in (0, C_k)$
or $b_k = y_i^k - \sum_{j=1}^k [\alpha_j^k - (\alpha^*)_j^k] (x_i^* x_j + 1)^2 + \varepsilon_k$, $(\alpha^*)_i^k \in (0, C_k)$
is bias at time point k .

Not only does (6) guarantee less fitting errors, but it also possesses a good generalization capability. It means that, after

learning previous unwrapped phases $\angle \hat{r}_0, \dots, \angle \hat{r}_{k-1}$, (6) can predict the variation trend of a phase efficiently for the next time point k and derive $\angle_{\text{SVR}} r_k = \sum_{i=1}^k [\alpha_i^k - (\alpha^*)_i^k] (x_i^* k + 1)^2 + b_k$. The phase $\angle \hat{r}_k$ is then unwrapped within a 2π interval centered around $\angle_{\text{SVR}} r_k$, by adding multiples of $\pm 2\pi$ to the principal value of $\angle \hat{r}_k$ when the absolute difference between $\angle_{\text{SVR}} r_k$ and the principal value of $\angle \hat{r}_k$ is greater than π . Namely, we select a proper m to satisfy $\angle \hat{r}_k = \arg(r_k) + 2\pi m \in [\angle_{\text{SVR}} r_k - \pi, \angle_{\text{SVR}} r_k + \pi]$. Till time point $k = N$, we get $\hat{f}_d^N = (\sum_{i=1}^N [\alpha_i^N - (\alpha^*)_i^N] x_i^N)/\pi$, $\hat{f}_r^N = (\sum_{i=1}^N [\alpha_i^N - (\alpha^*)_i^N] (x_i^N)^2)/2\pi$, and $\hat{\phi}^N = \sum_{i=1}^N [\alpha_i^N - (\alpha^*)_i^N] + b_N$ as the ultimate estimation values. This algorithm can be implemented recursively in time, as shown schematically in Figure 1.

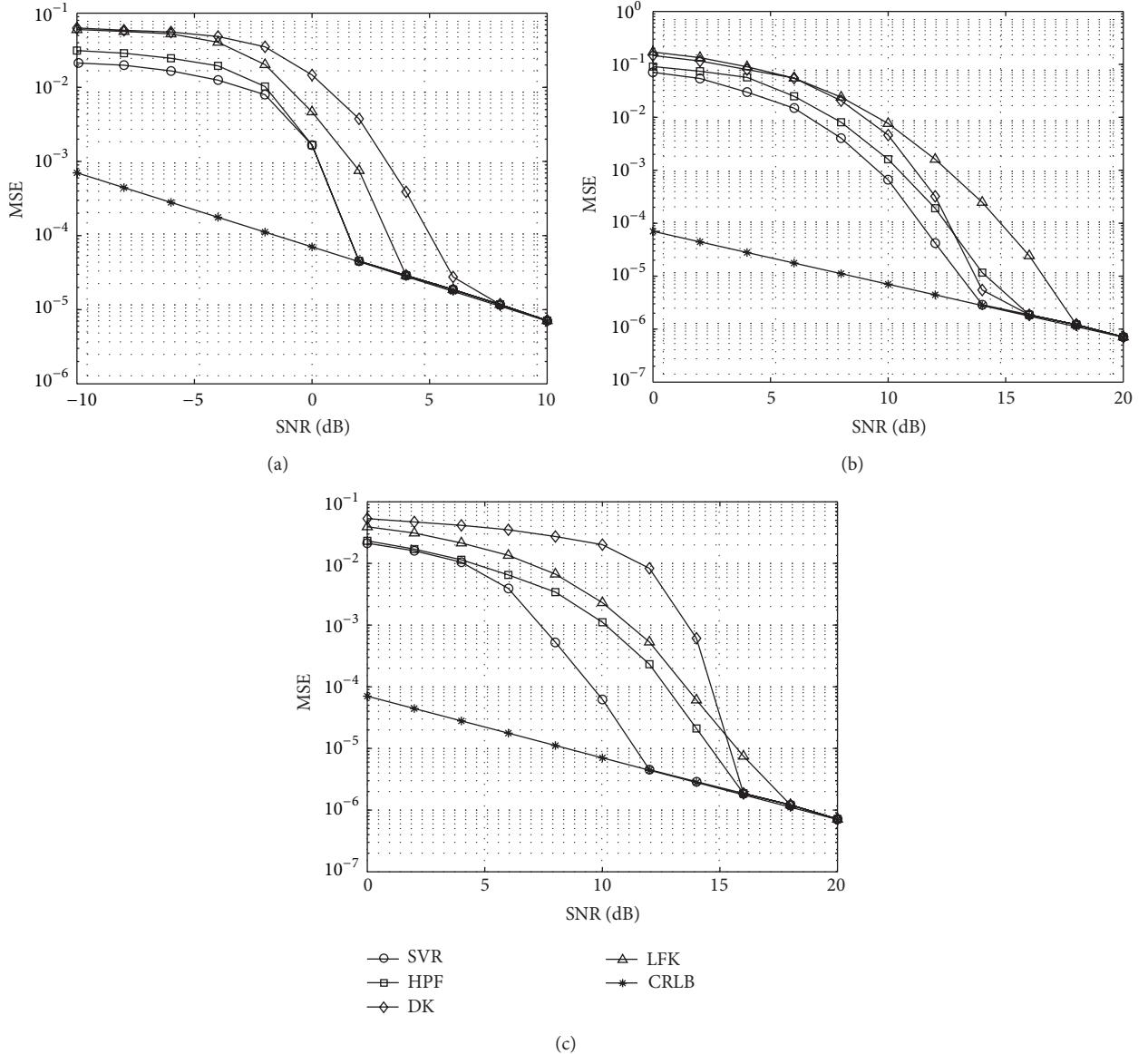


FIGURE 3: (a) MSE of IF with $f_d = 0.1$, $f_r = 0.05$, $\phi = 0$, and $N = 32$. (b) MSE of IF with $f_d = 0.3$, $f_r = 0.05$, $\phi = 0$, and $N = 32$. (c) MSE of IF with $f_d = 0.1$, $f_r = 0.2$, $\phi = 0$, and $N = 32$.

2.3. SVR's Parameter Settings. Setting SVR's parameters is a difficult problem, but has a pronounced impact on SVR's performance, for example, insensitive loss coefficient ϵ , penalty factor C . There is no complete theoretical basis or explicit closed form. Cross-validation is a widely used method until now, but it is complex and time consuming. In this study, we formulate proper parameter values by understanding SVR's theory and integrating a large number of references and experiments.

When k is small, imperfect phase unwrapping at a particular time point can easily have an impact on the variation trend of $f(x)$. In order to avoid this discrepancy, a new model must be constructed with the aptitude of better generalization capability. As k increases, its necessity decreases contrarily, which is due to the degrading impact of

improperly unwrapped phase. Simple speaking, the model's generalization capability is inversely proportional to the size of the set S .

Intuitively, insensitive loss coefficient ε is the vertical height of ε -tube. The larger ε is, the less support vectors there are. Nevertheless, too large ε will cause unfixable b . Noted that, if the predicted phase is in the vicinity of π 's odd times, the estimation performance is deteriorated rapidly for its ambiguous phase characteristic. Selecting a proper ε can reduce this impact. At the same time, ε is inversely proportional to SNR, so insensitive loss coefficient at time point k ($1 \leq k \leq N$) is given by [14]

$$\varepsilon_k = \tau \sqrt{\frac{1}{\text{SNR}} \frac{\ln k}{k}}, \quad k = 1, \dots, N, \quad (13)$$

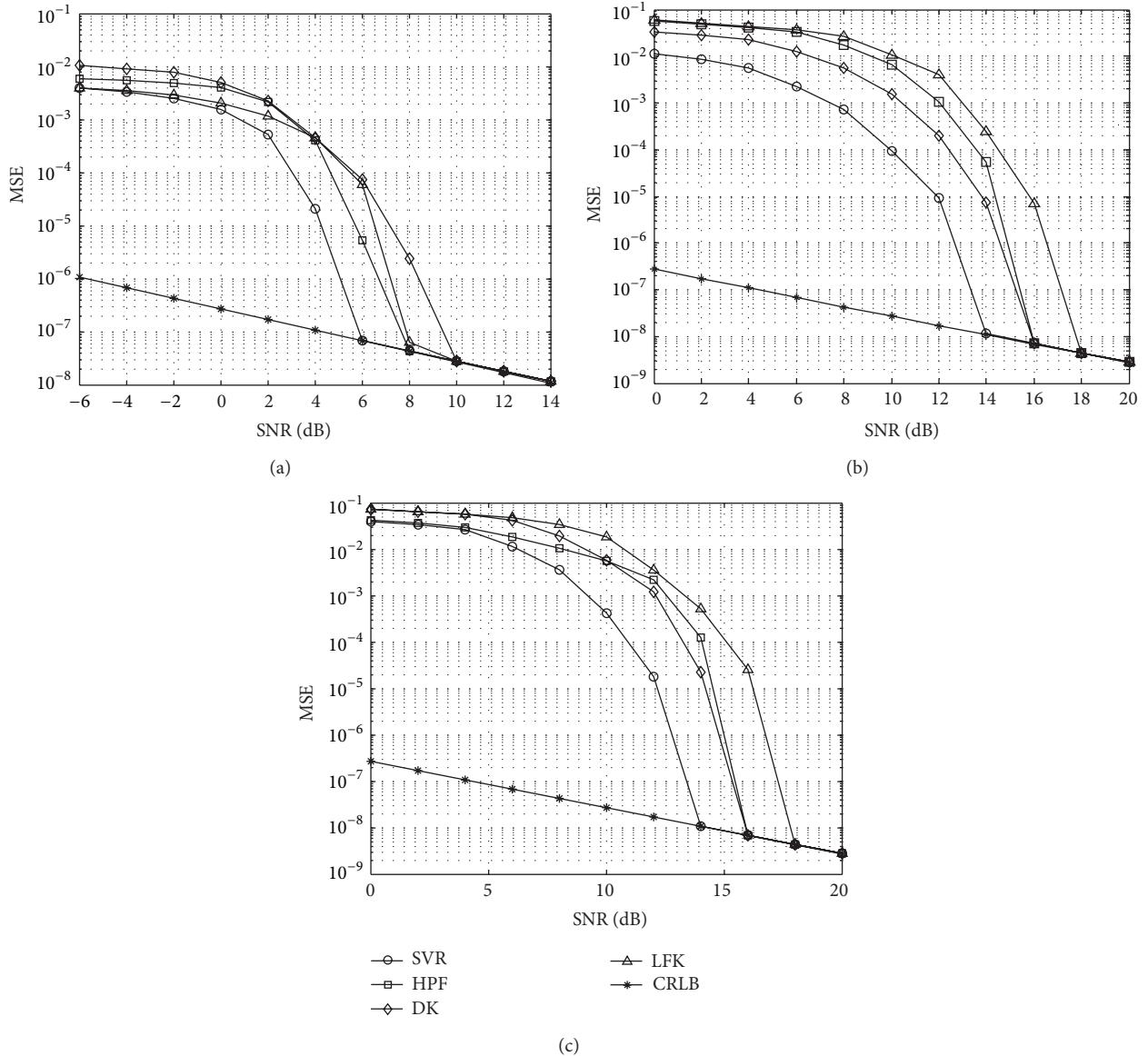


FIGURE 4: (a) MSE of IFR with $f_d = 0.1$, $f_r = 0.05$, $\phi = 0$, and $N = 32$. (b) MSE of IFR with $f_d = 0.3$, $f_r = 0.05$, $\phi = 0$, and $N = 32$. (c) MSE of IFR with $f_d = 0.1$, $f_r = 0.2$, $\phi = 0$, and $N = 32$.

where τ is a positive constant, and $\tau = 0.8$ is set in this study after experimental comparisons and validations, and SNR is assumed to be known.

Penalty factor C controls the penalty degree of vectors outside the ε -tube and determines SVR's generalization capability. C is directly proportional to the sample size and SNR. As (12) is a line in a two-dimension plane, the first item of target function during solving SVR's QP problem is directly proportional to its slope; so a very small C will result in a horizontal and underfitting C . Inspired by [14, 15], penalty factor at time point k ($1 \leq k \leq N$) is given by

$$C_k = \delta \sqrt{\text{SNR}} \sqrt[3]{k} \max(|\bar{g}_k - \lambda \sigma_k|, |\bar{g}_k + \lambda \sigma_k|), \quad (14)$$

$$k = 1, \dots, N;$$

where $\bar{g}_k = (1/k) \sum_{n=0}^{k-1} |y_n|^2$, $\sigma_k = \sqrt{(1/k) \sum_{n=0}^{k-1} (|y_n|^2 - \bar{g}_k)^2}$, $y_n = \angle \hat{r}_{n-1}$ is a section of the element of training set S_k ; as (13), SNR is assumed to be known; δ, λ are positive constants and $\delta = 0.1, \lambda = 0.5$ are set in this study, by the same way as τ .

3. Results and Analyses

We have compared the proposed algorithm entitled as SVR estimator with the other three: the HPF estimator proposed in [6]; the DK estimator proposed in [10]; the LFK estimator proposed in [12].

3.1. Phase Unwrapping Process. Because HPF estimator does not need the phase unwrapping process, Figure 2 illustrates

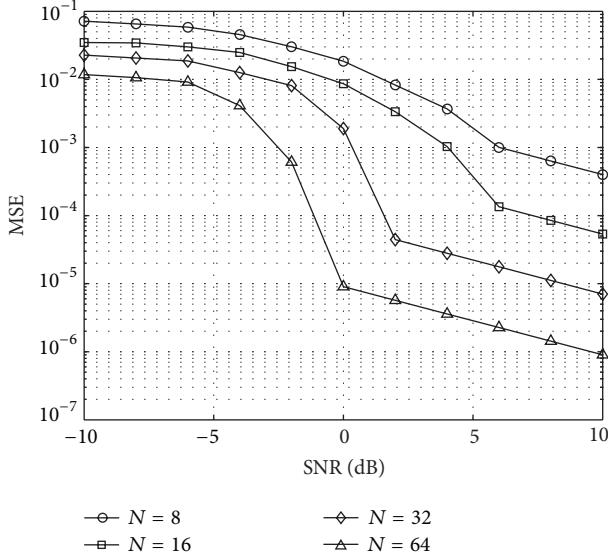


FIGURE 5: MSE of IF of SVR estimator with $f_d = 0.1, f_r = 0.05, \phi = 0$.

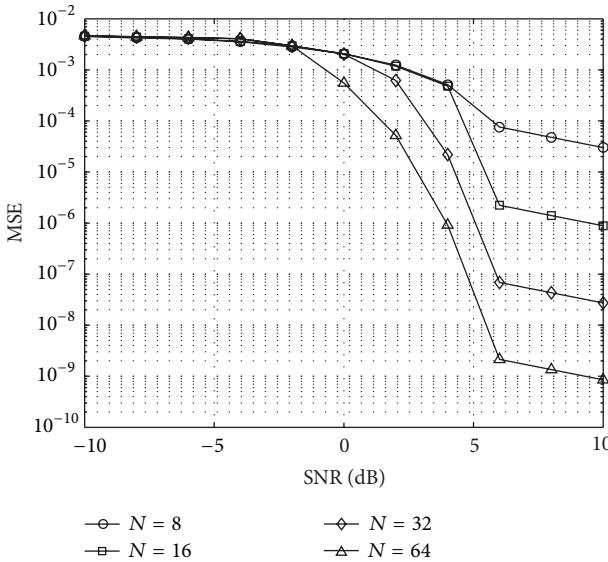


FIGURE 6: MSE of IFR of SVR estimator with $f_d = 0.1, f_r = 0.05, \phi = 0$.

the arbitrary phase unwrapping processes of the other three, while $f_d = 0.1, f_r = 0.05$, $f_d = 0.3, f_r = 0.05$, and $f_d = 0.1, f_r = 0.2$. The number of Monte Carlo experiments is 10000, and $\phi = 0$, $N = 32$, and SNR = 8 dB. It is shown that SVR estimator can unwrap phase accurately, whatever the value of f_d and f_r is. However, errors emerge in DK and LFK ones as f_d and f_r increase.

3.2. Estimation Performance. Figures 2, 3, and 4 illustrate the MSE curves of IF and IFR versus SNR, respectively, where MSEs are defined as $E[(\hat{f}_d - f_d)^2]$, $E[(\hat{f}_r - f_r)^2]$, \hat{f}_d, \hat{f}_r are

the estimation values of f_d, f_r . CRLBs are given by [12]

$$\begin{aligned} \text{CRLB}_{f_d} &= \frac{1}{\text{SNR}} \frac{6(8N-11)(2N-1)}{(2\pi)^2 N(N-1)(N^3+N^2-4N-4)}, \\ \text{CRLB}_{f_r} &= \frac{1}{\text{SNR}} \frac{360}{(2\pi)^2 N(N-1)(N^3+N^2-4N-4)}, \\ \text{CRLB}_\phi &= \frac{1}{\text{SNR}} \frac{3(3N^2-3N+2)}{2N(N^2+3N+2)}. \end{aligned} \quad (15)$$

It is shown that whether $f_d = 0.1, f_r = 0.05, f_d = 0.3, f_r = 0.05$, or $f_d = 0.1, f_r = 0.2$, SVR estimator is the best one during both IF and IFR estimations all the while; also, MSE performances of IF and IFR totally decrease as f_d and f_r increase, but SVR estimator is the most robust and has had a great advantage already when $f_d = 0.3, f_r = 0.05$ or $f_d = 0.1, f_r = 0.2$.

3.3. Impact of the Sample Size N . Everything is as in Figure 2 other than $f_d = 0.1, f_r = 0.05$; the MSE curves of IF and IFR of SVR estimator versus SNR while N is 8, 16, 32, and 64 are plotted in Figures 5 and 6, respectively. It is clear that as N increases, MSE performances of IF and IFR of SVR estimator are both improved.

3.4. Impact of Initial Phase ϕ . Everything is as in Figure 3 except that $\phi = 0.4\pi, 0.8\pi$; the MSE curves of IF and IFR are plotted against SNR in Figures 7 and 8, respectively. Comparing with Figures 3(a) and 4(a), we can see that SVR estimator is immune to ϕ , but the other three are not.

3.5. Computational Complexity. Because we translate SVR into QP problem and need to search the minimums during the process, we can not derive the explicit form of the computational complexity of SVR estimator. So everything is as in Figure 2 except that the number of Monte Carlo experiments is 100, SNR is 8 dB, and $f_d = 0.1, f_r = 0.05$; the consuming times are listed in Table 1, while N is 8, 16, 32, and 64, respectively. The running computer is ASUS-PC1111 having, Intel(R) Pentium 2.13 GHz CPU and 2.00 GB RAM.

We can see that SVR estimator's consuming times are more than the others', especially when N becomes larger. As a matter of fact, however, it is acceptable and tolerable.

4. Conclusions

Phase unwrapping process is a key point in phase-based frequency estimation of chirp signal. Firstly, we adopt SVR to learn the unwrapped phases at previous time points, predict the variation trend of phase efficiently, and derive the estimation value for the next time point. Once acquired, in terms of relationship between absolute signal phase and time series, we address a simple and effective frequency estimation algorithm of chirp signal. The proposed algorithm completely exhibits its advantages of higher estimation accuracy, lower sensitivity of frequency, and initial phase, by sacrificing more consuming times.

Because SVR predicts the curve's variation trend merely in terms of training set which consists of previous points'

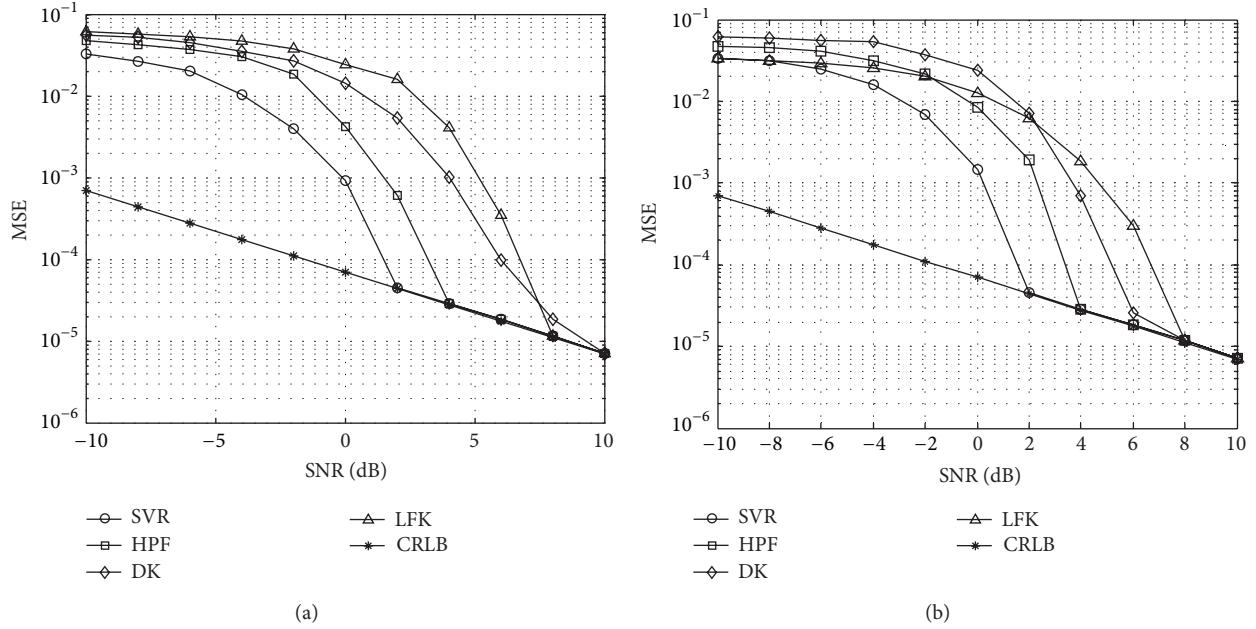


FIGURE 7: (a) MSE of IF with $f_d = 0.1$, $f_r = 0.05$, $\phi = 0.4\pi$, and $N = 32$. (b) MSE of IF with $f_d = 0.1$, $f_r = 0.05$, $\phi = 0.8\pi$, and $N = 32$.

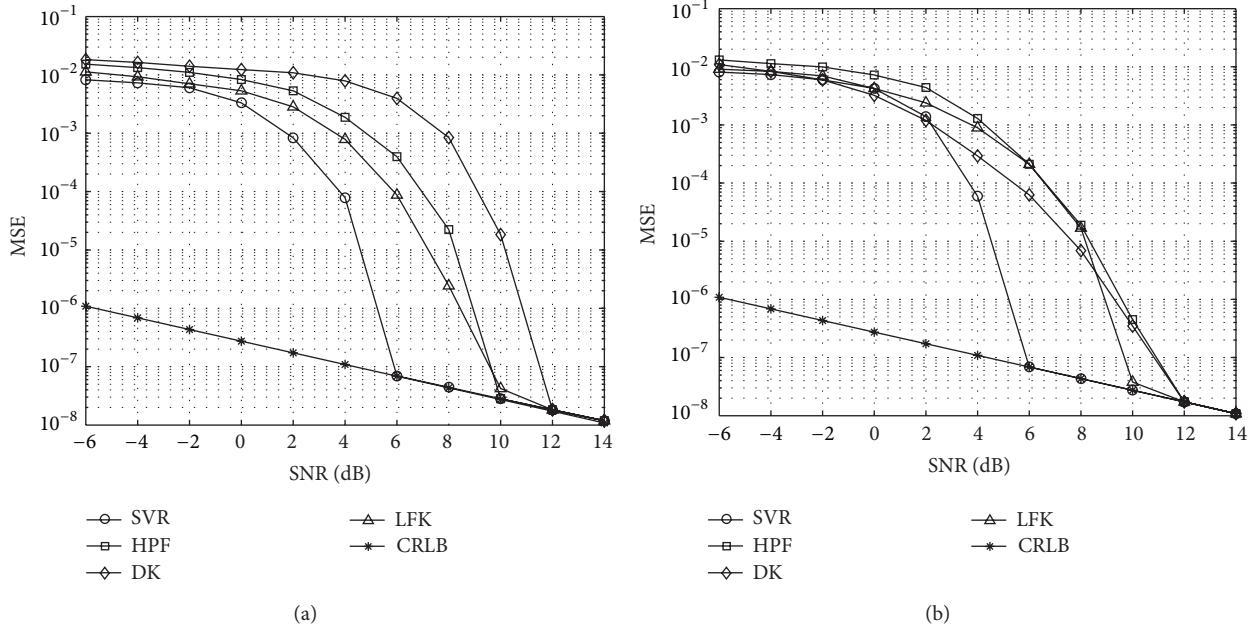


FIGURE 8: (a) MSE of IFR with $f_d = 0.1$, $f_r = 0.05$, $\phi = 0.4\pi$, and $N = 32$. (b) MSE of IFR with $f_d = 0.1$, $f_r = 0.05$, $\phi = 0.8\pi$, and $N = 32$.

TABLE 1: Consuming time with different N (ms).

Algorithm	$N = 8$	$N = 16$	$N = 32$	$N = 64$
HPF	380	942	3721	10654
DK	203	476	1098	3711
LFK	299	784	2362	8643
SVR	437	1107	4805	13074

values, we even can estimate the frequency of chirp signal under the nonGaussian condition by the same way.

Stressing that the proposed algorithm learns training set and gets the approximate values of SVR's insensitive loss coefficient ε and penalty factor C , as a next step, therefore, improving SVR's parameter setting is an important research point.

Acknowledgment

This research is supported by the National Natural Science Foundation of China (Grant no. 61201380).

References

- [1] J. C. Curlander and R. N. McDonough, *Synthetic Aperture Radar Systems and Signal Processing*, Wiley, New York, NY, USA, 1992.
- [2] J. Xu, Y. Peng, and X. Xia, “Parametric autofocus of SAR imaging— inherent accuracy limitations and realization,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 11, pp. 2397–2411, 2004.
- [3] R. Ramaswami, K. N. Sivarajan, and G. H. Sasaki, *Optical Networks: A Practical Perspective*, Morgan Kaufmann, San Mateo, Calif, USA, 1998.
- [4] P. O’Shea, “A new technique for instantaneous frequency rate estimation,” *IEEE Signal Processing Letters*, vol. 9, no. 8, pp. 251–252, 2002.
- [5] P. Wang, I. Djurović, and J. Yang, “Generalized high-order phase function for parameter estimation of polynomial phase signal,” *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3023–3028, 2008.
- [6] P. Wang, H. Li, I. Djurović, and B. Himed, “Performance of instantaneous frequency rate estimation using high-order phase function,” *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2415–2421, 2010.
- [7] M. Benidir and A. Ouldali, “Polynomial phase signal analysis based on the polynomial derivatives decompositions,” *IEEE Transactions on Signal Processing*, vol. 47, no. 7, pp. 1954–1965, 1999.
- [8] P. O’Shea, “A fast algorithm for estimating the parameters of a quadratic FM signal,” *IEEE Transactions on Signal Processing*, vol. 52, no. 2, pp. 385–393, 2004.
- [9] S. Ma, J. Jiang, and Q. Meng, “A fast, accurate and robust method for joint estimation of frequency and frequency rate,” in *Proceedings of the 2011 International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS ’11)*, pp. 1–6, Chiang Mai, Thailand, December 2011.
- [10] P. M. Djuric and S. M. Kay, “Parameter estimation of chirp signals,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 12, pp. 2118–2126, 1990.
- [11] H. Fu and P.-Y. Kam, “Phase-based, time-domain estimation of the frequency and phase of a single sinusoid in AWGN—the role and applications of the additive observation phase noise model,” *IEEE Transactions on Information Theory*, vol. 59, no. 5, pp. 3175–3188, 2013.
- [12] Y. Li, H. Fu, and P. Y. Kam, “Improved, approximate, time-domain ML estimators of chirp signal parameters and their performance analysis,” *IEEE Transactions on Signal Processing*, vol. 57, no. 4, pp. 1260–1272, 2009.
- [13] Y. Li and P. Y. Kam, “Improved chirp parameter estimation using signal recovery method, approximate, time-domain ML estimators of signal s and their performance analysis,” in *Proceedings of the IEEE 71st Vehicular Technology Conference*, pp. 1–5, Taipei, Taiwan, May 2010.
- [14] V. Cherkassky and Y. Ma, “Selection of meta-parameters for support vector regression,” in *Proceedings of the International Conference on Artificial Neural Networks*, pp. 687–693, Madrid, Spain, August 2002.
- [15] V. Cherkassky, X. Shao, F. M. Mulier, and V. N. Vapnik, “Model complexity control for regression using VC generalization bounds,” *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1075–1089, 1999.

Research Article

Multiscale Feature Model for Terrain Data Based on Adaptive Spatial Neighborhood

Huijie Zhang,^{1,2} Yun Ma,^{1,2} Zhiqiang Ma,^{1,2} Xinting He,^{1,2} Yaxin Liu,^{1,2} and Zijun Feng^{1,2}

¹ School of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, China

² Key Laboratory of Intelligent Information Processing of Jilin Universities, Changchun 130117, China

Correspondence should be addressed to Zhiqiang Ma; mazq@nenu.edu.cn

Received 10 July 2013; Accepted 18 August 2013

Academic Editor: William Guo

Copyright © 2013 Huijie Zhang 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.

Multiresolution hierarchy based on features (FMRH) has been applied in the field of terrain modeling and obtained significant results in real engineering. However, it is difficult to schedule multiresolution data in FMRH from external memory. This paper proposed new multiscale feature model and related strategies to cluster spatial data blocks and solve the scheduling problems of FMRH using spatial neighborhood. In the model, the nodes with similar error in the different layers should be in one cluster. On this basis, a space index algorithm for each cluster guided by Hilbert curve is proposed. It ensures that multi-resolution terrain data can be loaded without traversing the whole FMRH; therefore, the efficiency of data scheduling is improved. Moreover, a spatial closeness theorem of cluster is put forward and is also proved. It guarantees that the union of data blocks composites a whole terrain without any data loss. Finally, experiments have been carried out on many different large scale data sets, and the results demonstrate that the schedule time is shortened and the efficiency of I/O operation is apparently improved, which is important in real engineering.

1. Introduction

Terrain model is widely applied in actual engineering, such as film, game, and simulation. One of the grand challenges is that it requires huge amounts of data set, in order to guarantee its precision. Especially in recent years, with the rapid development of spatial visualization technology, the requirement for terrain details is higher and higher; therefore, the terrain scene is more and more complex. Owing to the massive data, it is very difficult on the establishment, management, and index of terrain model in the scientific computing visualization engineering.

The capacity of RAM is limited for storing huge amounts of data set; therefore, the support of external storage is needed, when rendering a large scale of terrain. According to the requirements of scene, it is important to dynamically load data into the memory interactively for realizing to render the scene in real-time visualization engineering. However, there is a serious speed difference between RAM and external storage, so that the interactive operation between them frequently makes a big influence to the efficiency of the system directly.

It has become a bottleneck for the process of large scale data. As a result, how to establish the model with huge amounts of data set and provide effective scheduling method is one of the research hotspots.

The pyramid is a kind of data model composed by multiresolution levels [1]. It can provide all kinds of resolution of the data without real-time sampling, because the model generates the different resolution data levels of data elevation model (DEM) in advance. It has been widely applied to commercial software. However, there are two aspects of drawbacks.

- (1) The pyramid model increases the storage space of data; that is to say, a lot of data is repeated on the different layers of it. Hence, decreasing the amount of loaded data is at the cost of consuming external storage.
- (2) There is still mass of redundant data in the data set which has been loaded in memory. No matter the terrain is plain or rugged, in the classical method, the data resolution for each level is uniform, and

interlaced sampling is adopted in the various adjacent levels, which does not consider the terrain geometry feature.

Lindstrom and Silva [2] proposed the dynamic data loading method using the function of mirror file. This method is extremely simple, and it takes advantage of operation system to divide data pages automatically. Dai et al. [1] brought up a method of incremental data update, which dynamically updated local terrain data, according to the move of viewpoint and the geometry center of the data page. Li et al. [3] adopted the technique of incremental horizon to eliminate the invisible part of the model.

In order to further improve the efficiency of data schedule, grid simplified model is expressed as data stream by many scholars [4]. Space-filling curve including Hilbert curve [5], Z curve [6], and Π Curve [7] in terrain model is a common approach of linearization for data in external storage. It still takes additional time on the index of external storage even if those methods improve the efficiency of data schedule in different ways. In recent years, a new function based on cluster analysis has been proposed to simplify the data model in external storage, and improve the efficiency of data schedule [8, 9]. But the out-of-core methods based on cluster almost aimed at irregular data set. However, the primary research object in terrain modeling engineering is large scale data set of grid model; therefore, the methods mentioned above are not capable of disposing the large scale data in real engineering. In addition, the parallel algorithm using hardware to deal with out-of-core terrain data was proposed. But it is not applied widely, because of its requirement for high quality hardware [10].

This paper utilized a multiresolution out-of-core model based on geometry terrain feature. There is the following advantages (1) It consists of physical model and logic model, in which the physical model is original full resolution data, so the model is independent of the data scale. (2) The logical model is a multiresolution hierarchical structure on physical model, which establishes the index of data blocks, according to the geometry feature of terrain. (3) It decreases the amount of data loaded to memory while keeping the external storage constant.

In this paper, we proposed a strategy of scheduling data, which is based on logic model after the cluster analysis. It decreases the time cost of searching the target data in external storage, by clustering the data blocks whose static geometry errors are similar. Meanwhile, we put forward the method that sorts the data blocks in the cluster via space-filling curve and presents encoding strategy on multiresolution terrain model. It resolves two primary problems (1) One is to map terrain data blocks in two-dimension structure into the form of one dimension. (2) The other is to solve the problem of how to intercept the data blocks in cluster for the multiresolution model in a local area.

2. Related Work

As we all know, the effect of 3D modeling based on vectors is usually good; however, there are a lot of problems such as

the process of large scale data and accelerative rendering. In recent years, few scholars have achieved significant research progress in the following aspects.

Level of details (LOD) is a classical simplification method for multiresolution model. There are three categories of algorithms for LOD, that is, triangulation algorithm based on quadtree [11], simplified algorithm based on adaptive grids [12], and progressive mesh algorithm [13].

Triangulation algorithm based on quadtree is to divide one data block into four blocks according to a screen error and viewpoint. Its advantage is that it built the hierarchy using Quad-tree code, which is easy in solving the crack problem of adjacent blocks on the common edges. The representative method contains algorithm based on constrained Quad-tree [14], simplified method based on implicit restricted quadtree [15], and so on. The advantage of this kind of algorithm is that it adopts the efficient strategy to refine layers from the top to the bottom. However, a mass of information of each quadtree node needs to be stored. Thus it consumes massive storage resources.

The simplified algorithm based on adaptive regular grid evaluates whether model data should be eliminated according to model error and then determined to divide grid area adaptively [12].

Progressive mesh algorithm adopts embedded grids to remove crack between layers by setting “skirt” for different multiresolution layers [13]. The literature [16, 17] takes advantage of geometry clipping to optimize algorithm that it creates data buffer in order to accelerate real-time roaming of terrain scene. Meanwhile, Sun et al. [18] described the technique to compress the texture based on mipmap structure. Although this kind of algorithm improves model precision, it needs higher CPU consumption. So there are difficulties for general 3D vector model, when dealing with large scale data sets.

Cluster analysis is also a valid strategy of solving problems in construction of multiresolution model. It is widely applied in many areas. The literature [19, 20] described the method of solving problems of computing coordinates mean by using cluster analysis.

When rendering multiresolution terrain model, it is necessary to deal with huge scale data. Jeong et al. [21] proposed a new method to render the complex scene. It is hard to imagine rendering such complex scene in early years. However, with the development of graphic hardware, GPU has become an important computing resource to deal with huge scale data. The literature [22, 23] shows more details using this new method.

3. Cluster Analysis Based on Similar Error

3.1. Cluster with Similar Error. Multiresolution hierarchy based on features (FMRH) is a structure on the basis of physical model. According to the structure of FMRH proposed in the literature [24], a mass of redundant data was eliminated from the view of terrain features, so that the multiresolution model could be stored in the minimum capacity of external storage. However, the efficiency of scheduling the data in external storage would not be improved even though the amount of physical data decreased. In fact, the information

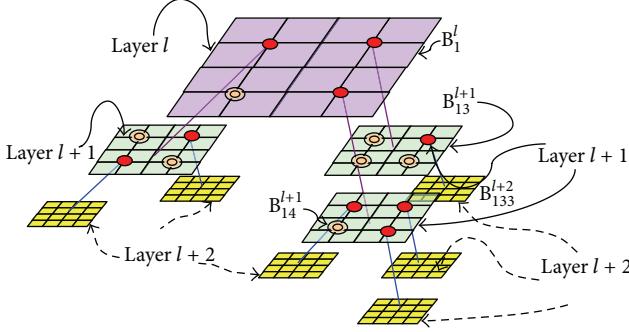


FIGURE 1: Multiresolution layer hierarchy of logical model in external storage.

of level of data blocks is not inheritance with its errors; that is to say, the errors of data blocks in the same level may not be similar. The error in current data block is only related to its source data block or its sub-data block, because the hierarchical structure proposed in our paper is based on a realization of depth-first traversal. As shown in Figure 1, B_1^l , and B_{13}^{l+1} , B_{14}^{l+1} are the blocks on the same branch in FMRH, and they satisfy the following relation:

$$\text{CovSE}(B_1^l) \geq \text{CovSE}(B_{13}^{l+1}) \geq \text{CovSE}(B_{133}^{l+2}). \quad (1)$$

The error of nodes in the same layer such as B_{13}^{l+1} and B_{14}^{l+1} is uncorrelated, while the one of B_{133}^{l+2} and B_{134}^{l+2} is possibly great, so it cannot load data into memory as the unit of layer. Hence, it is necessary to traverse the whole hierarchical structure in external storage to load data. As a consequence, selecting data blocks of multiresolution on different levels increases the burden of rendering scene in real time.

A cluster algorithm, which satisfies spatial constraint and maximum static error of closure based on FMRH, is raised in this section. The main idea of this algorithm is to make cluster partition for nodes which own similar errors in external storage and propose the method of spatial index for each cluster. According to this strategy, it achieves multiresolution terrain data of the current scene rapidly for each frame, every error threshold, position of viewpoint and visible area, without traversing the entire FMRH structure of terrain.

According to the literature [24] and formula (1) in this paper, the static spatial error threshold can be computed, and a series of error thresholds will be gained by adjusting the parameter ρ . The bigger the threshold, the smaller the amount of extracted data and, the lower the resolution of model; otherwise, the larger the amount of data, the higher the data resolution for memory. Let $\delta_{\text{metric}}^{\max}$ and $\delta_{\text{metric}}^{\min}$ be the maximum and minimum static spatial error threshold. $\delta_{\text{metric}}^{\max}$ equals to $\delta_{\text{metric}}^{\min}$ when establishing the FMRH; namely, $\rho = 0.03$. $\delta_{\text{metric}}^{\max}$ is the basic condition which reserves the profile feature of terrain, so it adopts $\rho = 0.2$ in the experiment. The threshold $\delta_{\text{metric}}^{\min}$ aims at guaranteeing the required higher resolution data in memory by FMRH, while the goal of threshold $\delta_{\text{metric}}^{\max}$ is to provide the lower resolution data which maintains the profile feature of terrain with the lowest requirement of the

simplified model. Thus, the static error in all data blocks of FMRH ranged from $\delta_{\text{metric}}^{\min}$ to $\delta_{\text{metric}}^{\max}$.

Definition 1. Let $\delta_{\text{metric}}^1, \delta_{\text{metric}}^2, \dots, \delta_{\text{metric}}^i, \delta_{\text{metric}}^{i+1}, \dots, \delta_{\text{metric}}^m$ be the static error threshold, and satisfy the constraint: $\delta_{\text{metric}}^{\min} \leq \delta_{\text{metric}}^1 \leq \delta_{\text{metric}}^2, \dots, \delta_{\text{metric}}^i \leq \delta_{\text{metric}}^{i+1}, \dots, \leq \delta_{\text{metric}}^m \leq \delta_{\text{metric}}^{\max}$. Assume the set $G = \{\xi_1, \xi_2, \dots, \xi_{m-1}\}$ and each ξ_i is in the set of data block B_{index}^l in different layers; then define ξ_i as a cluster based on $\delta_{\text{metric}}^{i-1}$ and δ_{metric}^i (similar error cluster, SEC), where it satisfies the following constraints:

- (1) $\delta_{\text{metric}}^i \geq \text{CovSE}(B_{\text{index}}^l) \geq \delta_{\text{metric}}^{i-1}$, where l can be the index of every spatial layer in the structure of FMRH,
- (2) $\bigcup_{B_{\text{index}}^l \in \xi_i} B_{\text{index}}^l = \Omega$, where Ω denotes an entire closed terrain space.

3.2. Cluster Algorithm Based on FMRH. A cluster instance is described as Figure 2, and Figure 2(a) is a cluster, which satisfies the constraints in Definition 1 as well. The data blocks marked within a transparent rectangle build the terrain area in Figure 2(b). These data blocks distribute on four levels in FMRH and satisfy the uniform error threshold. These data blocks obviously construct a closed space area which is to be proved in Section 3.3.

According to algorithm of construction of FMRH [24], it records the static error of closure of each data block in FMRH when constructing the FMRH model. Hence, the work is to traverse the structure of FMRH and put data blocks which own similar static error into the proper clusters, according to their static error. In order to satisfy the second constraint in Definition 1, traversing the structure of FMRH by depth-first strategy is better. More details of the cluster algorithm based on FMRH are shown as follows (Algorithm 1).

According to Algorithm 1, the data block in the same interval is aggregated into one cluster, while these different resolution data blocks distribute in different layers, and their union covers the terrain space of FMRH.

3.3. Proof of Cluster Space Closeness Proposition. The majority of data block with different resolution on the same branch distributes into different clusters by depth-first traverse, and the static error satisfies formula (1) on a branch from the top to the bottom in FMRH. The special case is that the error of data blocks, which are partly adjacent or distributed in closed layers, are similar on the same branch, so that they locate in the same error threshold interval. As you can see in Figure 3, three branches are marked by black dotted lines. Assume that there is small difference between $\text{CovSE}(B_{13}^1)$ and $\text{CovSE}(B_{134}^2)$; B_{13}^1 and B_{134}^2 are probably classified into one cluster, only through error threshold which data block belongs to. But it goes against the second constraint in Definition 1; in fact there is an overlapped partition in space between B_{13}^1 and B_{134}^2 .

In order to solve this problem, the current_block is not classified as cluster ξ_i , if there is ancestor data block in the current cluster ξ_i , in Algorithm 1. Thus, there is no overlapped area in the covered place of different data blocks for each

Input: The structure in external storage of FMRH.
Output: The set of cluster $G = \{\xi_1, \xi_2, \dots, \xi_{m-1}\}$.
Description: This algorithm generates the set of cluster by traversing the structure of FMRH with depth-first order, according to the relationship of static error and threshold.

(1) Initialization:

- (1.1) According to Definition 1, set $\delta_{\text{metric}}^1, \delta_{\text{metric}}^2, \dots, \delta_{\text{metric}}^i, \delta_{\text{metric}}^{i+1}, \dots, \delta_{\text{metric}}^m$ and make $\delta_{\text{metric}}^{\min} \leq \delta_{\text{metric}}^1 \leq \delta_{\text{metric}}^2, \dots, \delta_{\text{metric}}^i \leq \delta_{\text{metric}}^{i+1}, \dots, \delta_{\text{metric}}^m \leq \delta_{\text{metric}}^{\max}$.
- (1.2) Clear the clusters $\xi_1, \xi_2, \dots, \xi_{m-1}$ in aggregation G.

(2) Recall the function of **Traverse_FMRH** (FMRH, $B_1^0, 0$).
(3) Generate the index file of clusters, and save the index of data block for each cluster.
(4) The end.

Traverse_FMRH (FMRH, current_Block, level_S)
Function Description: Load static error of data blocks in FMRH, and decide which cluster it belongs to.
Parameter Description: FMRH is a multi-resolution level of structure based on features. The parameter current_Block is the current data block. The parameter level_S is the index of current data block in the level.

(1) Load the static error of current_Block from FMRH as current_Block.CovSE.
(2) Decide which cluster current_Block belongs to:

- (2.1) Construct $m - 1$ intervals according to m error thresholds $\delta_{\text{metric}}^1, \delta_{\text{metric}}^2, \dots, \delta_{\text{metric}}^i, \delta_{\text{metric}}^{i+1}, \dots, \delta_{\text{metric}}^m$
- (2.2) Each threshold interval corresponds a cluster from $i = 1$ to $m - 1$:
 - (2.2.1) Decide the size of current_Block.CovSE, if it meet two constraints:
① $\delta_{\text{metric}}^i \leq \text{current_Block.CovSE} \leq \delta_{\text{metric}}^{i+1}$;
② add data block current_Block into the cluster ξ_i , if there is no ancestor data block of current_Block including in the cluster ξ_i .
 - (2.2.2) When current_Block.leaf is false, then, get four data blocks sub_Block_j in the next level from current_Block in FMRH, and recall the function of DeepTraverse_FMRH(FMRH, sub_Block_j, level_S + 1).

(3) The end.

ALGORITHM 1: The cluster algorithm based on FMRH.

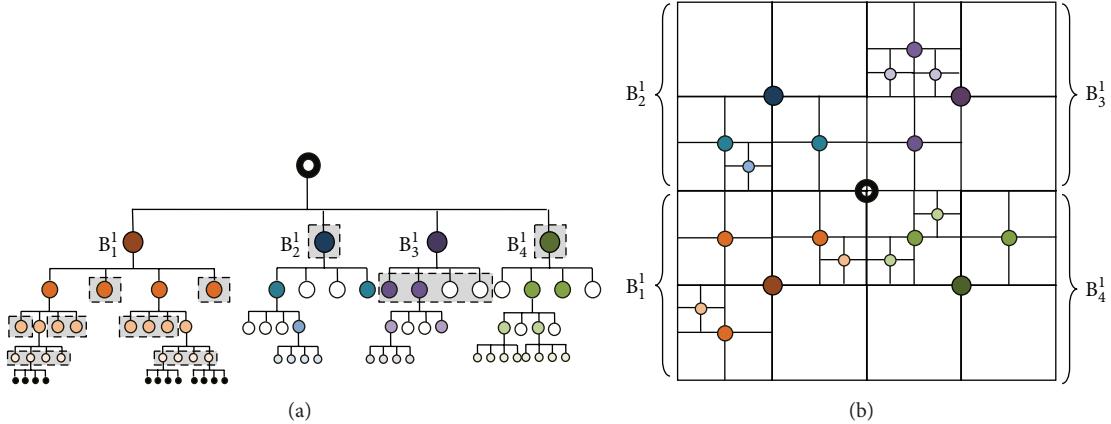


FIGURE 2: A cluster instance based on similar static error. (a) Structure of FMRH (transparent boxes denote data blocks which are gathered as the same cluster of the one corresponding to (b)). (b) The space closed terrain area of cluster corresponding to (a).

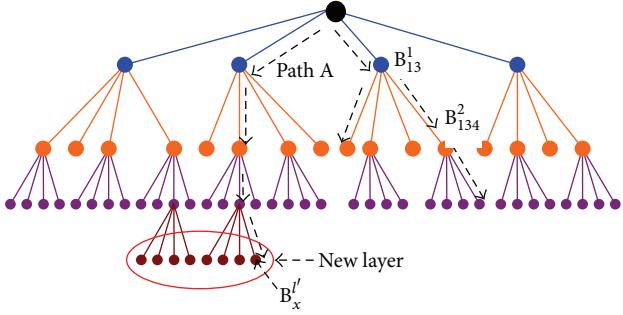


FIGURE 3: Analysis of problems about depth-first traversing FMRH.

cluster, and the union of all data blocks constructs the completed closed terrain. The following proof attempted to demonstrate the inevitability in theory.

Theorem 2. Let $G = \{\xi_1, \xi_2, \dots, \xi_{m-1}\}$ be the set of cluster generated by Algorithm 1; then for any ξ_i , $\bigcup_{B'_x \in \delta_i} B'_x = \Omega$ is held, where Ω denotes a completed terrain closed in the space.

Proof. Use induction method to prove the theorem. Let l be the maximum spatial layer in FMRH.

When $l = 0$, there is only one spatial layer in FMRH_0 .

According to Definition 1, $S^0 = \{B_1^0\}$ is consisting of single data block, so that the space is obviously closed.

Assume the theorem holds for any $\xi_i, l' \leq k$ and $\bigcup_{B'_x \in \xi_i} B'_x = \Omega$, when $l = k$. So that for FMRH_{k+1} a layer of data block is increased (denoted as Path A) on branch of each FMRH_k or remaining the branch (denoted as Path B), when $l = k + 1$. In the case of Path B, since there is no change on branch, all clusters, generated on the basis of FMRH_k , have already been closed in the space, when $l = k$. As a consequence, the closeness proposition holds. The remaining work is to prove spatial closeness proposition for the branch of Path A in cluster.

Without loss of generality, we chose any branch on Path A (shown as Figure 3) for any cluster ξ_i . According to the method of depth-first traversal in Algorithm 1, for the branch on Path A from the top to the bottom, the current data block B'_x , where $l' \leq k + 1$, shown in Figure 3, is divided into two categories. If the ancestor data block of B'_x is included in ξ_i , it will no longer be in ξ_i , before traversing the $k + 1$ layer. On the contrary, B'_x and its three brother data blocks are listed into ξ_i , conducted by recursive traversal for four sub-data blocks in step 2.2.2. For the former situation, keep ξ_i constant; thus the closeness proposition is held. B'_x and its three brother data blocks are overlapped with their father data blocks, so that the latter one still keeps the closeness proposition hold. This ends the proof of Theorem 2. \square

4. Index of Levels of Data Blocks in Cluster

The union of entire data blocks in each cluster composites a closed space of terrain through our algorithm. When loading

data, a proper cluster is obtained accurately as long the scope of error threshold as is ascertained. Therefore, it realizes multiresolution data loading with no redundancy, which does not need to search and judge further. However, one frame roamed in a scene just requires a small part of data to load in memory. Thus, the work of encoding and sorting data blocks in each cluster by characteristics of space is necessary in order to establish the relationship of multiresolution model of level and data blocks in cluster. It realizes the index of local data blocks rapidly as well.

As a matter of fact, the reconstruction of large scale grid data is applied widely in terrain visualization, volume rendering, and matrix operation, and its advantage is that it verifies data access locally. The best strategy of linearization of multidimensional data is using the technique of space-filling curves. The reconstructed data can be intercepted as the rule of filling, according to the requirements, while the closeness proposition in space is still held. This characteristic is corresponded to the goal of indexing the feature cluster and takes advantage of the local proposition of space-filling curves. It can also realize the interception of data in any scope.

4.1. Strategy of Hilbert Space-Filling Curves. This paper focuses on the space-filling curves of two-dimension, because the data of elevation can be mapped into the a tow-dimension space. The characters of these curves are listed as follows. (1) They not only verify to traverse every data block in the space, but the data blocks are adjacent in local area of filling curves. (2) It is easy to form the triangle strips rendering real time, and the public vertices can be used more than once. It takes advantage of the modern graphic hardware resource. These papers adopted Hilbert curves to guide the process of index coding for data blocks in cluster, so that it improves the efficiency of data loading of terrain.

The basic approach of generating Hilbert curves, formed with the process of subdividing the current square to four small ones and connecting the center of these squares, is recursion. The literature [5] proposes a serial of indexes for Hilbert curve along different coordinate's axes. However, the Hilbert curve cannot be put into use directly, because data blocks in each cluster are presented with multiresolution.

It is necessary to consider the level information of data blocks, in order to apply it to Hilbert curve in the multiresolution model. The train of thought is as follows. First of all, ascertain the filling order of children data blocks where its number is less than or equal to four, which is determined by their father data blocks, by using the level information of data block in cluster. The data block traversed forms a multiresolution filling curve in a certain arrangement from the top to the bottom, by executing the process of above judgments. The details are stated as follows.

- (1) Divide the data block into four types in cluster ξ_i as type I, II, III, and IV. This division plays two roles in classifying the data blocks. On one hand, it can make sure the sequence in space-filling curve of the sub-data block is of current data block. On the other hand, it takes advantage of self-similarity of Hilbert curve to summarize rules of production. Therefore, the type of

- sub data block is ascertained, and it can be used for recognizing where the position of data block of next layer in the curve is.
- (2) Sort sub-data blocks, according to the current type of data block. For type I, sort sub-data blocks as mark 0 to 3. For type II, sort sub-data blocks as mark 0 to 3. For type III, sort sub-data blocks as mark 0 to 3. For type IV, sort sub-data blocks as mark 0 to 3.
 - (3) Let the type of current data block be T_B , and the type of its child's data block is probably T_0 , T_1 , T_2 , or T_3 . In order to make sure of the type of its child's data block, the following rules of production are proposed.

Rule 1. If $T_N = \text{I}$ then $T_0 = \text{III}$, $T_1 = \text{I}$, $T_2 = \text{I}$, and $T_3 = \text{IV}$.

Rule 2. If $T_N = \text{II}$ then $T_0 = \text{IV}$, $T_1 = \text{II}$, $T_2 = \text{II}$, and $T_3 = \text{III}$.

Rule 3. If $T_N = \text{III}$ then $T_0 = \text{I}$, $T_1 = \text{III}$, $T_2 = \text{III}$, and $T_3 = \text{II}$.

Rule 4. If $T_N = \text{IV}$ then $T_0 = \text{II}$, $T_1 = \text{IV}$, $T_2 = \text{IV}$, $T_3 = \text{I}$, and $T_4 = \text{I}$.

4.2. Algorithm of Generating Multiresolution Model Using Hilbert Space-Filling Curve. The necessary condition that each cluster can be sorted through space-filling curve is proved as follows. Due to Definition 1 and Theorem 2, there is only one data block on the same branch in the same cluster, and the set of all data blocks in a cluster composites a closed space area. Thus, cluster is a set with no intersections and spatial closeness data blocks. An instance of cluster is shown as follows.

By traversing data blocks in cluster from layer l , top layer in cluster, to layer l' , the lowest layer in cluster, it forms space-filling curves. Data blocks are sorted in cluster, because each cluster is produced by the manner of depth-first recursion. The principle is that it backtracks to the original data blocks when a branch traverses all of the data blocks in current cluster, and then executes the depth-first recursive operation of their brother data blocks. As a result, adjacent data blocks selected in cluster are the nearest ones in FMRH space. As shown in Algorithm 2, taking an example for a cluster, more details of generating space-filling curves are described.

An instance of coding cluster is as follows. Taking the area of left bottom corner as an example, it contains 11 data blocks B_{11}^1 , B_{11}^2 , B_{11}^3 , B_{11}^4 , B_{1131}^3 , B_{1132}^3 , B_{1133}^3 , B_{1134}^3 , B_{1141}^3 , B_{1142}^3 , and B_{1143}^3 . Checking the first and the last data blocks of the original cluster, their types can be deduced, supposing the type of the top layer l is type III. Therefore, the type of data block B_{11}^1 is type I, according to rule 3 in Algorithm 2. As sorting its sub-data block by type I, B_{111}^2 , B_{112}^2 , B_{113}^2 , B_{114}^2 , B_{111}^2 , and B_{112}^2 are coded directly as 0 and 1, because they are data blocks in cluster. As for B_{113}^2 and B_{114}^2 , the algorithm examines their sub-data blocks furthermore because they are not in the cluster. The types of B_{113}^2 and B_{114}^2 are type I and type IV, respectively, because of the type of B_{11}^1 is type I.

The sub-data blocks B_{1131}^3 , B_{1132}^3 , B_{1133}^3 , and B_{1134}^3 of B_{113}^2 are not changed after sorting because the type is type I. In the

same way, the sub-data blocks B_{1141}^3 , B_{1142}^3 , B_{1143}^3 , and B_{1144}^3 of B_{114}^2 have been changed after sorting as B_{1143}^3 , B_{1142}^3 , B_{1141}^3 , and B_{1144}^3 ; therefore, the codes for them are $B_{1143}^3 \rightarrow 6$, $B_{1142}^3 \rightarrow 7$, $B_{1141}^3 \rightarrow 8$, and $B_{1144}^3 \rightarrow 9$. As a result, the filling curve in area of the left bottom corner consists of B_{11}^1 , B_{111}^2 , B_{112}^2 , B_{1131}^3 , B_{1132}^3 , B_{1133}^3 , B_{1134}^3 , B_{1143}^3 , B_{1142}^3 , B_{1141}^3 , and B_{1144}^3 .

4.3. Update Model with Large Scale Data Sets. The out-of-core management framework allow the update of original data sets and expansion of new data sets. The steps of updating the physical model are as follows.

- (1) Dividing new data source as square physical data block on the basis of the form of $n \times (2^m + 1) \times (2^m + 1)$. For the irregular shape data source, the manner of mending boundaries is adopted to form many physical data block files. Then arranging physical file blocks, it remains the original irregular size of terrain, which satisfies the condition of $(2^m + 1) \times (2^m + 1)$ for each file size.
- (2) Storing a physical block used head file and data body file. Replacing the old data file with new data file, if it is the one to be updated, and then modifying the head file; otherwise, adding new head file and body file into physical model. For each physical block, establishing the corresponding logical model, and adding cluster analysis. The steps of updating logical model are as follows.
 - (1) Construct the structure of FMRH.
 - (2) Get the set of cluster G , and make cluster analysis for FMRH.
 - (3) Establish the adjacent cluster set G' for each couple of neighboring physical file blocks.

5. Multiresolution Data Schedule

The majority of data schedule methods is view-dependent, which computes the error of simplified model at the same time to schedule data, so that it realizes loading multiresolution model, in order to decrease amount of loaded data in memory. The efficiency of the operation for I/O is low, because the burden of schedule program is too heavy. The multiresolution model is established directly with calculation, as long as it is achieving the error threshold of model, because the schedule program is based on multiresolution logical model and data block cluster proposed previously. Figure 4 describes the framework of data schedule based on the above thinking. The majority of works has been completed already in Sections 3 to 4. The questions still need to be solved are as follows.

- (1) Establishing the relationship between screen error of view-dependent model and the static error of closure, so that it ascertains the error threshold of loaded data in memory, according to the requirements for memory, and determines to which cluster the data source belongs.

Input: Cluster ξ .
 Output: Space-filling curve (code arrangement of data block) CodeString.
 Description: This algorithm generates the space-filling curve which contains all data blocks in cluster ξ , by judging the type and position of data blocks in cluster ξ .

(1) Initialization:

- (1.1) Define the top layer l and bottom layer l' in cluster ξ .
- (1.2) Extract the first and the last data block from cluster ξ . According to this, decide the type T of data block B^l in layer l .
- (1.3) Clear CodeString.

(2) Recall the function **DeepTraverse_Cluster** (B^l, T, l).

(3) Modify the index file of cluster, and save the new index code of data blocks in cluster.

(4) The end

DeepTraverse_Cluster (currentBlock, Type, l)

Function description: As Hilbert space filling curve, generate code of data block in cluster.

Parameter description: currentBlock is the current data block, Type is the type of current data block, l is the layer of current data block.

- (1) Get the four data blocks B_1, B_2, B_3 , and B_4 of currentBlock.
- (2) Sort B_1, B_2, B_3 , and B_4 , as the type of Type (shown in Figures 4(a)–4(d)).
- (3) Judge the data block is whether in cluster ξ :
 - (3.1) If B_i is a data block in cluster ξ , then encode B_i by the order after sorting, and add it into CodeString, where $i = 0, 1, 2$, and 3.
 - (3.2) If B_i is not a data block in cluster ξ , then ascertain the type of B_i by using rules, and make the recursion furthermore, where $i = 0, 1, 2$, and 3.
 - (3.2.1) If Type is I, then the types of B_1, B_2, B_3 , and B_4 are Type III, Type I, Type I, and Type IV.
 - (3.2.2) If Type is II, then the types of B_1, B_2, B_3 , and B_4 are Type IV, Type II, Type II, and Type III.
 - (3.2.3) If Type is III, then the types of B_1, B_2, B_3 , and B_4 are Type I, Type III, Type III, and Type II.
 - (3.2.4) If Type is IV, then the types of B_1, B_2, B_3 , and B_4 are Type II, Type IV, Type IV, and Type I.
 - (3.2.5) According to the new order, judge data blocks on next layer
 - For $i = 1$ to 4:
 If $l + 1 \leq l'$, then recall **DeepTraverse_Cluster** (B_i , Type B_i , $l + 1$), where Type B_i denotes the new type of data block B_i .
- (4) The end of function.

ALGORITHM 2: The algorithm of generating multi-resolution model of Hilbert space-filling curve.

- (2) Locating and gaining the real data block according to the index of cluster in physical model.

5.1. Searching Strategy of Target Cluster. The view-dependent simplification technology is that it selects the proper multiresolution terrain model according to the position and direction of viewpoint. Finer layer is used for terrain near the viewpoint, and coarser layer for the area is abundant to the viewpoint. We adopt the static error of closure to realize the multi-resolution model in external storage, because there is no information of viewpoint, when organizing the data in

external storage. In the schedule program, determining the needed physical data block in practice, which takes advantage of static error, should be calculated according to the error of screen to select the proper index of cluster. The relation between the screen error of view-dependent model and the static error of closure is set up in this section. Let τ be the dynamic error threshold of viewpoint model, and let ε be the static error threshold. The relation is stated below:

$$\varepsilon = \frac{\tau}{\lambda} \cdot P_{\min}. \quad (2)$$

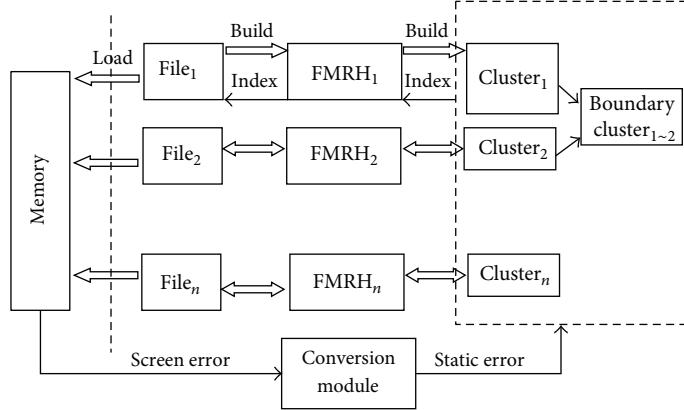


FIGURE 4: Framework of data schedule in external storage.

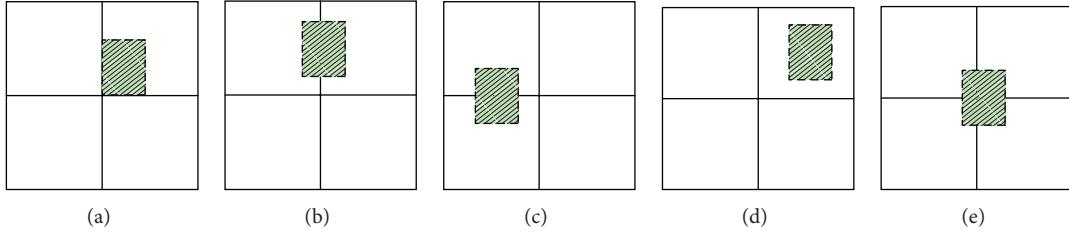


FIGURE 5: Five types of position relation of index of multifile data block.

Let λ be the number of pixels of screen in a unit distance, and let P_{\min} be the nearest distance from the viewpoint to screen. Applying formula (2), set the value of τ , such as 0.5, 1, 1.5, or 2, which are the values of pixel. So the required static error threshold ε of current multiresolution scene can be obtained. Then the index of cluster is ascertained based on the size relation between the cluster static error threshold preestablished in advance and that in ε . Assume that the arrangement of static error threshold in cluster which is $\delta_{\text{metric}}^{\min} \leq \delta_{\text{metric}}^1 \leq \delta_{\text{metric}}^2, \dots, \delta_{\text{metric}}^i \leq \delta_{\text{metric}}^{i+1}, \dots, \leq \delta_{\text{metric}}^m \leq \delta_{\text{metric}}^{\max}$ is in the interval between δ_{metric}^i and $\delta_{\text{metric}}^{i+1}$, so that the multiresolution data block recorded by cluster ξ_i loads into memory.

5.2. Data Prefetching and Strategy of Incremental Data Schedule. For data prefetching operation, double-threads pattern is adopted, in order to render the terrain system in real time and guarantee roaming the scene coherently. Incremental data schedule is used to decrease the amount of data loaded in memory, by taking advantage of the correlation of different frames in space and time. The main thread is used to render foreground scene, and the other one realizes preschedule of data blocks stored in external storage.

As for single file, apply the index of cluster in Section 5.1 to ascertain the selected cluster ξ_i . Determine the code of the beginning and the end data block in cluster ξ_i , according to the required position and size of data in the current scene. For example, assuming that the codes of the beginning and the end block are 10 and 24, this part of multiresolution data clips the required data and loads into memory, through searching

the related file of physical data block from the structure of FMRH.

The aim is to locate data blocks in which files for the multifile data schedule operation. It is easy to determine the required data blocks, by traversing latitude and longitude of physical data blocks contained in it and position relation of current scene according to the index of head file of each data source. There are only five kinds of position relation as shown in Figure 5, because data blocks are classified by rules. So a set of cluster corresponds with each physical data block. The method in Section 5.1 selects the proper cluster ξ_i for each physical data block in the set of cluster and implements the same process for each ξ_i as single file. In addition, it is very important to match them using the adjacent cluster of the boundary of two files. Finally, we prepare the gotten data by the file mapping technology in data buffer.

Most operation of data schedule is for single file; the amount of data for one frame in roaming scene is not large. The maximum number of files is four, when it satisfies the condition of multifile (illustrated in Figure 5), so that it creates four file mappings at most. This question can be solved using file mapping API supported by VC. Another simple method, which can improve the efficiency of data schedule and decrease the amount of loaded data, is using incremental data schedule. A method which is similar to the literature [25] is adopted, which forms the supplication data block of strip shape on the direction of which viewpoint changes and puts them into a data buffer, in order to support data for next frame. The difference in our method against the above method is based on multiresolution data strips of the FMRH

Input: The dynamic error of view-dependent τ ,
 data scale of current data of scene m , data
 position: longitude LONG, latitude LAT.

Output: Pre-fetch data set D , and load file mapping buffer.

Description of function: Locate physical data block, and
 load data buffer in memory through the threshold of screen
 error and the position of scene, as the requirement of
 the amount of data in memory.

- (1) Initialization: Clear data set D .
- (2) Traverse the head file of data source (physical model),
 and determine which data block belongs to it
 according to (LONG, LAT), meanwhile, record the number of involved files.
- (3) For each file from $n = 1$ to fileN, do the process as follows:
 //If it is a single file, then execute once,
 otherwise, execute more than once
 - (3.1) According to τ and Formula (2), compute the error
 threshold ε needed by current scene.
 - (3.2) Get cluster G , and let the arrangement of static error
 thresholds of G be: $\delta_{\text{metric}}^{\min} \leq \delta_{\text{metric}}^1 \leq \delta_{\text{metric}}^2, \dots, \delta_{\text{metric}}^i \leq \delta_{\text{metric}}^{i+1}, \dots,$
 $\leq \delta_{\text{metric}}^m \leq \delta_{\text{metric}}^{\max}$, determine which interval belongs to,
 and define the cluster which meets the requirements as ξ_{in} .
 - (3.3) If n is less than fileN, then go to step 3.
- (4) According to the moving direction of viewpoint, update the
 value of LONG and LAT, and establish fileN file mapping buffers.
- (5) If fileN is 1, then determine the initial position of ξ_{i1}
 (according to LONG, LAT), and intercept data block D' .
- (6) If fileN is greater than 1, then determine the position
 of cluster ξ_{in} , and let n be 1, 2, ..., and fileN:
 - (6.1) Judge the relation of the file of two adjacent data block
 (as Figures 5(b), 5(c), and 5(e));
 - (6.2) Distribute data blocks in ξ_{in} , and form D' .
 According to the position relation Z-shape filling adjacent
 data block cluster after that.
- (7) If the intersection of D' and the existed data in data
 buffer is empty, then load D' , otherwise load $D' - D_{\text{old}} \cap D'$,
 namely load data incrementally.
- (8) The end.

ALGORITHM 3: The algorithm of data prefetching and incremental data schedule.

structure. But the method supposed by the literature [25] is based on the nested strip-shaped data structure. Specifically, each strip is the same resolution data block with no relation to terrain features. So the amount of loaded data is much more than the incremental data schedule. More details on data prefetching and the strategy of incremental data schedule are described in Algorithm 3.

6. Results and Discussion

We have researched on ten data sources and analyzed results in our experiment, in order to verify the model based on huge dataset in external storage and the validity of scheduling. These data sources contain GTOPO30 global elevation data, the elevation data of Jilin province, Zhuijiang, and Tian-chi (crater) lake of the Changbai Mountains in China, Colorado Grand Canyon, Mount Rainier, Crater Lake, Puget Sound, Seattle, and Yakima in America. The sum amount of elevation data is about 8,111,735,217 Byte \approx 7.554 G, and the one of

texture is 21.37 G. We used a PC with a 3.0 GHz Pentium 4 CPU, 1 GB system memory, and a Geforce 5900fx graphic card. We also used VC++ as the IDE for software developing.

We have experimented with the amount of loaded data in memory and schedule time for 10 datasets by setting error threshold τ 1 and 2.5, respectively. The resolution of screen is 1024×768 , where the clipped window size of GTOPO30 and Zhuijiang is 129×129 , the one of Jilin province, Tian-chi crater, Colorado Grand Canyon, Puget Sound, and Mount Rainier is 257×257 , and the one of Seattle, Yakima, and Crater Lake is 513×513 .

Table 1 describes the result of the time and the amount of loaded data, when screen error is 0.5 pixels. There are four situations. The first one is loading data from physical block model directly. Without loss of generality, levels of detail are adopted as well, which is similar to pyramid model, but the data is the same resolution loaded in the same time. The second one is loading multiresolution data block through the FMRH structure in our logic model. The third one is

TABLE 1: Time and amount of data load when screen error is 1 pixel.

Data source	Using physical model directly		Using logic structure FMRH		Schedule time using cluster analysis (s)	Schedule time using space-filling curve (s)
	Loaded triangles	Schedule time (s)	Loaded triangles	Schedule time (s)		
GTOPO30	29,491	0.18	22,938	0.15	0.08	0.06
Jilin province	115,343	0.52	89,129	0.48	0.26	0.20
Zhujiang	29,164	0.20	21,955	0.13	0.06	0.03
Tian-chi	120,586	0.56	95,683	0.51	0.24	0.12
Colorado Grand Canyon	125,829	0.55	98,304	0.53	0.23	0.11
Seattle	477,102	2.20	367,002	1.90	0.98	0.70
Yakima	466,616	2.10	356,516	1.87	0.90	0.69
Crater Lake	498,074	2.03	398,459	1.96	0.94	0.70
Puget Sound	121,897	0.70	98,304	0.54	0.29	0.18
Mount Rainier	125,829	0.69	95,683	0.53	0.26	0.15

TABLE 2: Experiment result of the quantity of data load in different frames when screen error is 2.5 pixels.

Data source	1st frame		10th frame		100th frame		1034th frame	
	Loaded triangles	Schedule time (s)						
GTOPO30	19,661	0.04	1,967	0.010	1,804	0.009	1,924	0.010
Jilin province	76,022	0.16	4,842	0.050	4,947	0.052	4,859	0.050
Zhujiang	18,678	0.02	2,241	0.010	2,306	0.015	2,598	0.015
Tian-chi	85,289	0.10	5,063	0.070	49,040	0.067	5,813	0.720
Colorado Grand Canyon	83,886	0.09	5,381	0.030	5,412	0.035	5,569	0.035
Seattle	319,816	0.65	10,231	0.400	10,059	0.390	10,952	0.420
Yakima	325,059	0.60	10,565	0.360	10,884	0.370	10,237	0.340
Crater Lake	309,330	0.61	10,967	0.390	10,932	0.390	11,098	0.410
Puget Sound	85,197	0.15	4,568	0.069	4,623	0.069	4,754	0.070
Mount Rainier	83,886	0.12	4,922	0.060	4,867	0.053	4,733	0.051

loading data block by using the hierarchical cluster to index FMRH structure. The fourth one is loading data block by using cluster to index FMRH structure, with encoding and arranging nodes in cluster through space-filling curve.

Three results can be obtained from Table 1. On one hand, compared with using physical files, the data quantity in memory and the time of data schedule are reduced apparently after using the logic structure of FMRH. On the other hand, data quantity loaded in memory is invariant; however, the time of data schedule is shortened obviously, after using the data schedule of cluster analysis. In addition, it is significant to use space-filling curve on the base of cluster analysis, so that the time of data schedule is further shortened. The main reason for this result is that the physical model has been simplified according to the ups and downs of terrain by the structure FMRH. So a mass of redundant data is eliminated before loading in memory. It ensures loading small scale of multiresolution data into memory each time, so that the time of data schedule is decreased. Another reason for this result is that the cluster analysis is taken for data block of similar errors to the structure of FMRH when establishing the logic

model. Locating the suitable cluster for data block, when data schedule is preceded, shortens the time of searching data block and realizes the high efficient data schedule. Finally, it is an efficient way which is proper to file mapping, taking advantage of space-filling curve to code and arrange data blocks in cluster. Meanwhile, owing to space-filling curve is local, the method can retrieve the local data blocks rapidly.

With the threshold of screen error increased, the quantity of data loaded in memory and schedule time decreases obviously. Data in Table 1 is the result of loading in memory at the first time, but in fact the incremental data schedule technology is adopted when it roams the scene. As a result, with the rate of frame increased, the time of data schedule and I/O response is decreased furthermore. With screen error threshold 2.5, data quantity loaded in different frames is shown in Table 2. Data quantity in first frame and the time of loading data are large, and the ones in 10th, 100th, and 1034th frame tend to be stable. Compared to the first frame, data quantity in other frames in memory is decreased obviously, and the efficiency of data schedule is increased significantly, because it adopts incremental data schedule.

7. Conclusions

This paper proposed the strategy of rapid data schedule based on cluster analysis, which improves the efficiency of I/O operation. It is very significant to expand terrain modeling method for large scale data in application of real engineering. More details follow.

Firstly, the cluster analysis method based on FMRH is proposed to solve the linearization problem of multiresolution terrain model. In details, each cluster satisfies two constraints. One is that data blocks owned similar error are classified into one cluster; the other is that the multiresolution data blocks in cluster covered the whole closed terrain spatial zone. Because data needs to satisfy some screen error threshold when it loads into memory, the first constraint conforms to the data requirement of simplified model in memory. The second constraint verifies loading data in each cluster as a unit, so that the situation of data block loss does not appear.

Moreover, the proposed strategy sorts each cluster through space-filling curve and encodes it. Because the nodes within the cluster are closed in space and obtained through the depth-first traversal. Small adjustments and local updating on the clustering can be mapped to form a one-dimensional sequence of contiguous spatial data blocks. The advantage of those strategies is that there is no need traversing the whole data set.

Finally, our strategy solves the problem of file-block splicing and adopts incremental data scheduling strategy to further reduce the amount of data loaded into memory when roaming the scene. As a result, our method improved the scheduling efficiency, which is important in real engineering. In our future work, the theory and idea about scheduling [26], planning [27], and phase transitions [28] will be introduced to improve the efficiency of terrain modeling further.

Acknowledgments

The authors are grateful for the helpful comments and suggestions of the reviewers, which have improved the presentation. This work is supported by the Research Fund for the Doctoral Program of Higher Education of China (no. 20100043120012) and supported by the national Natural Science Foundation of China for Young Scholars (no. 41101434).

References

- [1] C.-G. Dai, Y.-S. Zhang, and X.-Q. Deng, "An organization and management approach of data for real-time visualization of massive terrain dataset," *Journal of System Simulation*, vol. 17, no. 2, pp. 406–413, 2005.
- [2] P. Lindstrom and C. T. Silva, "A memory insensitive technique for large model simplification," in *Proceedings of the IEEE Conference on Visualization*, pp. 121–126, October 2001.
- [3] S. Li, J.-F. Ji, X.-H. Liu, and E.-H. Wu, "High performance navigation of very large-scale terrain environment," *Journal of Software*, vol. 17, no. 3, pp. 535–545, 2006.
- [4] H. T. Vo, S. P. Callahan, P. Lindstrom, V. Pascucci, and C. T. Silva, "Streaming simplification of tetrahedral meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 1, pp. 145–155, 2007.
- [5] Y. Bandoh and S. Kamata, "An address generator for an N-dimensional pseudo-Hilbert scan in a hyper-rectangular parallelepiped region," in *Proceedings of the International Conference on Image Processing (ICIP '00)*, pp. 737–740, September 2000.
- [6] P. Lindstrom and V. Pascucci, "Terrain simplification simplified: a general framework for view-dependent out-of-core visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 3, pp. 239–254, 2002.
- [7] J. Wu, X. Liu, and E. Wu, "Real-time rendering for out-of-core terrain model," *Journal of Computer-Aided Design and Computer Graphics*, vol. 17, no. 10, pp. 2196–2202, 2005.
- [8] E. Shatter and M. Garland, "A multiresolution representation for massive meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 2, pp. 139–148, 2005.
- [9] S.-E. Yoon, B. Salomon, R. Gayle, and D. Manocha, "Quick-VDR: out-of-core view-dependent rendering of gigantic models," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 4, pp. 369–382, 2005.
- [10] D. Dalecki, J. Lebiedz, K. Mieloszyk, and B. Wiszniewski, "Efficiency of interactive terrain visualization with a PC-cluster," in *Parallel Processing and Applied Mathematics*, vol. 4967 of *Lecture Notes in Computer Science*, pp. 391–399, 2008.
- [11] P. Lindstrom, D. Koller, W. Ribarsky, L. F. Hodges, N. Faust, and G. A. Turner, "Real-time, continuous level of detail rendering of height fields," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*, pp. 109–118, August 1996.
- [12] H. Zhang, Y. Lü, and S. Liu, "A terrain model simplification method based on adaptive areas division," *Journal of Computer Research and Development*, vol. 47, no. 1, pp. 53–61, 2010.
- [13] H. Hoppe, "Progressive meshes," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*, pp. 99–108, August 1996.
- [14] R. Pajarola, "Large scale terrain visualization using the restricted quadtree triangulation," in *Proceedings of the IEEE Visualization Conference*, pp. 19–26, October 1998.
- [15] H.-J. Zhang, J.-G. Sun, Y.-H. Lu, N. Lu, and Y.-Z. Wang, "A new simplification method for terrain model based on divergence function," *Chinese Journal of Computers*, vol. 32, no. 5, pp. 962–973, 2009.
- [16] F. Losasso and H. Hoppe, "Geometry clipmaps: terrain rendering using nested regular grids," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 769–776, 2004.
- [17] M. Duchaineau, M. Wolinsky, D. E. Sigeti, M. C. Miller, C. Aldrich, and M. B. Mineev-Weinstein, "ROAMing terrain: real-time optimally adapting meshes," in *Proceedings of the IEEE Visualization Conference*, pp. 81–88, October 1997.
- [18] C.-H. Sun, Y.-M. Tsao, and S.-Y. Chien, "High-quality mipmapping texture compression with alpha maps for graphics processing units," *IEEE Transactions on Multimedia*, vol. 11, no. 4, pp. 589–599, 2009.
- [19] B. Georgescu, I. Shimshoni, and P. Meer, "Mean shift based clustering in high dimensions: a texture classification example," in *Proceedings of the 9th IEEE International Conference on Computer Vision*, pp. 456–463, October 2003.
- [20] T. Ju, S. Schaefer, and J. Warren, "Mean value coordinates for closed triangular meshes," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 561–566, 2005.

- [21] Y. Jeong, K. Kim, and S. Lee, "Multi-resolution depth-of-field rendering," in *Proceedings of the International Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH '12)*, p. 107, 2012.
- [22] J. D. Owens, D. Luebke, N. Govindaraju et al., "A survey of general-purpose computation on graphics hardware," *Computer Graphics Forum*, vol. 26, no. 1, pp. 80–113, 2007.
- [23] J. Vanek, B. Beneš, A. Herout, and O. Stava, "Large-scale physics-based terrain editing using adaptive tiles on the GPU," *IEEE Computer Graphics and Applications*, vol. 31, no. 6, pp. 35–44, 2011.
- [24] H. Zhang, Z. Ma, M. Zhang, H. Sun, and Y. Li, "An out-of-core model for large terrain based on right-triangle irregular network," *ICIC Express Letters B*, vol. 2, no. 5, pp. 1021–1026, 2011.
- [25] F. Losasso and H. Hoppe, "Geometry clipmaps: terrain rendering using nested regular grids," in *Proceedings of the International Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH '04)*, pp. 769–776, Los Angeles, Calif, USA, August 2004.
- [26] X. Li and M. Yin, "An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure," *Advances in Engineering Software*, vol. 55, pp. 10–31, 2013.
- [27] D. Cai and M. Yin, "On the utility of landmarks in SAT based planning," *Knowledge-Based Systems*, vol. 36, pp. 146–154, 2012.
- [28] J. Zhou, P. Huang, M. Yin, and C. Zhou, "Phase transitions of expspace-complete problems," *International Journal of Foundations of Computer Science*, vol. 21, no. 6, pp. 1073–1088, 2010.

Research Article

Message Passing Algorithm for Solving QBF Using More Reasoning

Su Weihua, Yin Minghao, Wang Jianan, and Zhou Junping

School of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, China

Correspondence should be addressed to Wang Jianan; wangjn@nenu.edu.cn and Zhou Junping; zhoujp877@nenu.edu.cn

Received 26 June 2013; Accepted 24 July 2013

Academic Editor: William Guo

Copyright © 2013 Su Weihua 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 present a novel solver for solving Quantified Boolean Formulae problem (QBF). In order to improve the performance, we introduce some reasoning rules into the message passing algorithm for solving QBF. When preprocessing the formulae, the solver incorporates the equality reduction and the hyperbinary resolution. Further, the solver employs the message passing method to obtain more information when selecting branches. By using the unit propagation, conflict driven learning, and satisfiability directed implication and learning, the solver handles the branches. The experimental results also show that the solver can solve QBF problem efficiently.

1. Introduction

Quantified Boolean Formulae can be seen as an extension of the well-known canonical NP-complete problem of Propositional Satisfiability (SAT) with existential or universal quantifiers to every variable in the propositional formulae. Given a Quantified Boolean Formula, the question of deciding the satisfiability of the formula is called a Quantified Boolean Satisfiability problem (QBF). QBF is an important issue in artificial intelligence because it is the prototypical PSPACE-complete problem. Many practical problems can be transformed into QBF, for example, conformant planning [1], verification [2], nonmonotonic reasoning [3], and reasoning about knowledge [4].

Now most researchers make great efforts on designing excellent QBF solvers in order to increase the efficiency for solving QBF problem. WalkQSAT is the first incomplete QBF solver using stochastic local search methods, and it applies a SAT solver—walksat for selecting the next branch [5]. WATCHEDCSB features watched data structure [6]. QuBE is an adaptation of the classic DPLL (Davis, Putnam, Logemann, and Loveland) backtracking search algorithm. Moreover, it adopts watched data structure and learning [7]. QSAT employs the maximum occurrences in minimum sized clauses (MOMS) heuristic [8]. Quaffle extends variable state independent decaying sum (VSIDS) heuristic and conflict driven learning [9].

In this paper, we propose a novel QBF solver based on a DPLL algorithm, called EHSPQBF. The system combines many advanced techniques together. In preprocessing, we adopt the equality reduction and the hyperbinary resolution to simplify the formulae. In choosing branches process, we employ a message passing method—survey propagation. By giving exact information, the approach selects branches more exactly, which can decrease the search space and improve the backtracking time. In branched treatment process, we utilize conflict reasoning, conflict driven learning, and satisfiability directed implication and learning to help pruning the search space. Owing to these outstanding techniques, the ability of solving QBF problems has greatly improved.

2. Quantified Boolean Formulae

We begin this section by presenting some notions in this paper. A literal is either a Boolean variable v or its negation $\neg v$. If a literal is l , the negation of the literal is $\neg l$. A clause is a disjunction of literals which does not contain a complementary pair v and $\neg v$ simultaneously. A propositional logic formula in Conjunction Normal Form (CNF) is a conjunction of clauses. A Quantified Boolean Formula (QBF) has the form

$$Q_1 x_1 \cdots Q_n x_n E(x_1, x_2, \dots, x_n), \quad (1)$$

where $E(x_1, x_2, \dots, x_n)$ is a propositional logic formula in CNF involving Boolean variables x_1, x_2, \dots, x_n , and every Q_i ($1 \leq i \leq n$) is either an existential quantifier \exists or a universal quantifier \forall . A variable is existential if the restriction quantifier of the variable is \exists , and universal otherwise. Any Boolean variable v can take a value *true* or *false*. A truth assignment for a propositional logic formula is a map that assigns each variable a value. The satisfying assignment, called model, is the truth assignment that makes the propositional logic formula evaluated to *true*. The partial variable assignment is a map that assigns some variables values. We say a formula composed of zero clause is an empty formula, denoted by $E = \emptyset$, which is interpreted as *true*. An empty clause is a clause containing zero literal, which means that the clause is *false*.

The Quantified Boolean Formulae Satisfiability problem is to decide whether a Quantified Boolean Formula is satisfiable. For example, given a QBF $\forall x_1 \exists x_2 ((x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2))$, the QBF is satisfiable if and only if for each truth assignment to x_1 there exists a truth assignment to x_2 . In the rest of the paper, we will use QBF to denote the formulae and the satisfiability problems.

For a QBF of the form (1), because $\exists x_1 \exists x_2 E(x_1, x_2) \equiv \exists x_2 \exists x_1 E(x_1, x_2)$ and $\forall x_1 \forall x_2 E(x_1, x_2) \equiv \forall x_2 \forall x_1 E(x_1, x_2)$, we can group in the same set all consecutive variables having the same quantifier. Therefore, equality (1) can be rewritten into the following form:

$$Q_1 X_1 \cdots Q_k X_k E(x_1, x_2, \dots, x_n) \quad (1 \leq k \leq n), \quad (2)$$

where X_i ($1 \leq i \leq k$) is a set of propositional variables and all such sets are mutual disjoint. In such a format, each quantifier is applied to a set of variables rather than to a single Boolean variable. Moreover, the sequence of quantifiers alternates; an existential quantifier follows a universal quantifier and vice versa. With the quantifier restricted, each variable has a quantification level, which increases from the outermost quantification set to the innermost quantification set; that is, the variables belonging to the outermost quantification set have quantification level 1 and so forth.

3. The Relationship between SAT and QBF

The SAT problem is a typical NP-complete problem, which has a close relationship with QBF problem. More specifically, SAT can be regarded as an especial form of QBF with only existing existential quantifier, and QBF can be regarded as an extended form of SAT with additional prefixes. In addition, they are both prototypical problems for complexity classes. SAT is an NP-complete problem while QBF is a PSPACE-complete problem, which is inherently more difficult than SAT. In the following, we will discuss the similarities and dissimilarities between QBF and SAT.

Orders of Disposing Variables. When dealing with SAT, we can select variables in a random order. While dealing with QBF, we must select variables in a sequence of quantifier level that is, the variables whose quantifier level is minimal will be assigned first and so on. As a matter of fact, the reason

of doing so is that the variables in QBF have a dependency relationship. For example, given a QBF $\forall x_1 \exists x_2 ((x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2))$, the value of an existential x_2 that makes the formula true may depend directly on the value of a universal x_1 which is quantified prior to x_2 . Therefore, the key of dealing with QBF is to select the variables in the sequence of quantifier level.

Search Space. The search space of SAT is a tree while the search space of QBF is an AND/OR tree. They both regard the assignment of variables as weights and variables as nodes. The difference is that QBF has two types of nodes: AND node and OR node. In the AND/OR tree, the universal variables correspond to the AND nodes and the existential variables correspond to the OR nodes.

Hardness of the Problems. The QBF is more difficult than SAT. As we know, SAT is NP-complete while QBF is PSPACE-complete. For a SAT, the algorithm will stop when a satisfying assignment is found. However, for a QBF, the algorithm needs to continue searching because of the universal quantifiers.

Because of the close relationship between SAT and QBF, most of methods for QBF can be regarded as a generalization of algorithms commonly for SAT. Therefore, we can solve QBF with equality reduction and the hyperbinary resolution which is really effective in solving SAT.

4. QBF Algorithm Framework

Recently, search-based QBF solvers are based on an extension to the DPLL algorithm. Now we give the DPLL algorithm for QBF in Algorithm 1. In the algorithm, C_\emptyset is an empty clause and $C_{\text{all}\forall}$ is a clause composed of only universal variables. The DPLL algorithm works on the principle of preprocessing, choosing branches, branched treatment, and then recursively solving the simplified QBF. In preprocessing process (line 1), DPLL employs several reasoning rules to simplify QBF like pure literal rule, unit literal rule, and so forth. In choosing branches (variables) process (line 4), DPLL selects variables in a fixed order; that is, the variables in the outermost quantification set are chosen first and so on. In branched treatment process (lines 5–8), DPLL searches AND/OR tree depending on the chosen variable. If the chosen variable is an existential variable, an OR node is explored; otherwise an AND node is explored. DPLL terminates when the simplified QBF is an empty formula (line 2), which means the input formula $Q \cdot E$ is satisfiable; when the simplified QBF contains an empty clause or there is a clause made of all universally quantified variables (line 3), which means the input formula $Q \cdot E$ is unsatisfiable.

In Algorithm 2, we present the basic algorithm of EHSPQBF, which is based on DPLL algorithm. At first, we explain the terms occurred in the algorithm. The term C_\emptyset is an empty clause, and $C_{\text{all}\forall}$ is a clause composed of only universal variables. In preprocessing process (line 1), we simplify the input formula in CNF employing unit literal rule, the equality reduction, and the hyperbinary resolution. In choosing branches process (line 9), we use survey propagation as a heuristic. By giving exact information,

```

Procedure DPLL ( $Q \cdot E$ )
(1) preprocess ( $Q \cdot E$ );
(2) if  $E = \emptyset$  then return SAT;
(3) if  $(C_\emptyset \in E) \vee (C_{\text{allv}} \in E)$  then return UNSAT;
(4)  $v \leftarrow \text{choosevariable} (Q \cdot E)$ ;
(5) if ( $v$  is existential)
(6) then return DPLL ( $Q \cdot E|_{v=\text{true}}$ ) or DPLL ( $Q \cdot E|_{v=\text{false}}$ )
(7) if ( $v$  is universal)
(8) then return DPLL ( $Q \cdot E|_{v=\text{true}}$ ) and DPLL ( $Q \cdot E|_{v=\text{false}}$ )

```

ALGORITHM 1: DPLL for solving QBF.

```

Procedure EHSPQBF ( $Q \cdot E$ )
(1) preprocess ( $Q \cdot E$ );
(2) if ( $E = \emptyset$ ) then return SAT;
(3) if  $(C_\emptyset \in E) \vee (C_{\text{allv}} \in E)$  then return UNSAT;
(4) result = deduce();
(5) if (result = conflict)
(6) then analyze_conflict();
(7) if (result = satisfaction)
(8) then analyze_satisfaction();
(9)  $v \leftarrow \text{SP\_choosevariable} (Q \cdot E)$ ;
(10) if ( $v$  is existential)
(11) then return EHSPQBF ( $Q \cdot E|_{v=\text{true}}$ ) or EHSPQBF ( $Q \cdot E|_{v=\text{false}}$ )
(12) if ( $v$  is universal)
(13) then return EHSPQBF ( $Q \cdot E|_{v=\text{true}}$ ) and EHSPQBF ( $Q \cdot E|_{v=\text{false}}$ )

```

ALGORITHM 2: EHSPQBF algorithm.

the approach makes the choice of branch more exactly, which can decrease the search space and improve the runtime. In branched treatment process (line 4), we deduce the formula using conflict reasoning rule. The result of the deduction has three values: *undetermined*, *satisfaction*, and *conflict*. If result is *conflict* (*satisfaction*), that is, E evaluates to 0 (resp., 1) under the current partial variable assignment, the algorithm analyzes the conflict (resp., satisfaction); if the result is *undetermined*, that is, E evaluates neither 0 nor 1, the algorithm continues to choose branches. In this process, we employ the conflict driven learning and satisfiability directed learning to reduce the search space.

5. Preprocessing via Reasoning Rules

Besides the unit literal rule, the preprocess stage also uses more powerful reasoning rules, that is, the hyperbinary resolution and the equality reduction [10]. As a matter of fact, the hyperbinary resolution rule is a sequence of ordinary resolution steps. Though they are of similar function in simplifying the formulae, the hyperbinary resolution reduces the space which the sequence intermediate clauses occupy. And the equality reduction rule can further reduce the size of the input formula. It first finds the equivalences, and then it substitutes each equivalent variable, which can simplify the input formula. Now we describe them in the following.

Definition 1. The hyperbinary resolution rule of inference takes as input a $m + 1$ -ary clause $(l_1 \vee l_2 \vee \dots \vee l_m \vee l)$ and m binary clauses each of the form $(\neg l_i \vee l')$ ($1 \leq i \leq m$), where l_i ($1 \leq i \leq m$) is existential variable; then it deduces the clause $(l \vee l')$.

The advantage of the hyperbinary resolution is that it can save the solving time. The hyperbinary resolution can generate unit clauses or binary clauses in order to simplify the input formula. In addition, if the deduced clause contains only universal variables, the formula can be directly decided the QBF instance is *unsatisfiable*. For example, consider a formula $F = \forall x \exists x_1 x_2 x_3 x_4 x_5 x_6 (x_1 \vee x) \wedge (x_2 \vee x) \wedge (x_3 \vee x) \wedge (x_4 \vee x) \wedge (x_5 \vee x) \wedge (x_6 \vee x) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee \neg x_4 \vee \neg x_5 \vee \neg x_6)$. If we apply the hyperbinary resolution rule to F , the rule is able to obtain a deduced clause x which contains only one universal variable, and we can decide that the formula is *unsatisfiable*. By simplifying the formula, the hyperbinary resolution rule reduces the solving time.

Definition 2. The equality reduction of a QBF formula F by a pair of equivalent literals $l_1 \equiv l_2$ is a rule of replacing l_1 by l_2 if F satisfies

- (1) F contains the pair of binary clauses $(\neg l_1 \vee l_2)$ and $(l_1 \vee \neg l_2)$;
- (2) the quantification level of l_1 is greater than l_2 ;
- (3) l_1 and l_2 are both existential.

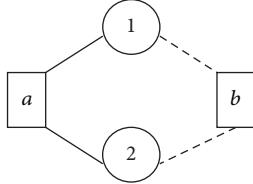


FIGURE 1: A simple example of a factor graph.

Note that the QBF formula F is *unsatisfiable* if there is a universal in the clause $\neg l_1 \vee l_2$. Once that equality reduction finds the equivalences, then it substitutes each variable with its definition in the formula. For example, let $F = \exists x_1 \forall x_3 x_4 \exists x_2 (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_4) \wedge (x_1 \vee x_2 \vee x_4)$. Since literals x_1 and x_2 are equivalent literals, we obtain the simplified QBF. F' is $\exists x_1 \forall x_4 (x_1 \vee \neg x_4) \wedge (x_1 \vee x_4)$ by using equality reduction. Notice that when the hyperbinary resolution rule applies to F' , the input formula can be further simplified.

Above all, we can find that the hyperbinary resolution rule and equality reduction have the power to reduce the size of the formula. They cannot only occupy less space but also increase the efficiency for solving QBF.

6. Choosing Branches with Survey Propagation

The survey propagation (SP) [11] method is a groundbreaking development for solving SAT. The occurrence makes the SAT problems that we can deal with scale from with ten thousand variables to with one million variables. As far as we know, SP is the only algorithm successful at solving random SAT problems with one million variables and beyond in near-linear time in the hardest region. In this section, we will discuss how to choose branches with the SP method. As a message passing algorithm, SP passes message on a factor graph which provides an easy graphical description to the message passing procedures. In the following, we will address the factor graph in detail.

6.1. Factor Graph. The factor graph is an undirected graph (see Figure 1). It has two types of nodes: one represents variables called “variable node” (circles in Figure 1) and the other represents clauses called “function node” (squares in Figure 1). A function node a is connected to a variable node i by an edge whenever the variable i (or its negation) appears in the clause a . In the graphical representation, we use a full (dashed) line between a and i whenever the variable appearing in the clause is i ($\neg i$). For every variable node i , we denote by $V(i)$ the set of function nodes to which it is connected, by $V_+(i)$ the subset of consisting of function nodes where the variable appears unnegated, and by $V_-(i)$ the complementary set of $V_+(i)$. $V(i) \setminus b$ denotes the set $V(i)$ without node b . Similarly, for each function node a , we denote by $V(a) = V_+(a) \cup V_-(a)$ the set of neighboring variable nodes. For example, given a QBF $\forall x_1 \exists x_2 (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$, we can represent the propositional logic formula with factor

graph in Figure 1, where clause a represents $x_1 \vee x_2$ and clause b represents $\neg x_1 \vee \neg x_2$.

6.2. Message Passing. The survey propagation algorithm passes two types of messages: one passed from a function node to a variable node; the other passed from a variable node to a function node. The update of both messages is realized by the message update rule which is defined as follows.

- (1) The message passed from a function node a to a variable node i : the message, called “survey”, is a real number $\eta_{a \rightarrow i} \in [0, 1]$:

$$\eta_{a \rightarrow i} = \prod_{j \in V(a) \setminus i} \left[\frac{\Pi_{i \rightarrow a}^u}{\Pi_{i \rightarrow a}^u + \Pi_{i \rightarrow a}^s + \Pi_{i \rightarrow a}^0} \right]. \quad (3)$$

If $V(a) \setminus i$ is empty, then $\eta_{a \rightarrow i} = 1$. In equality (3), the variable j sends a particular symbol to clause a saying that the variable cannot satisfy the clause (“ u ”), that the variable can satisfy the clause (“ s ”), or that it is indifferent (“ 0 ”).

- (2) The message passed from a variable node i to a function node a : each variable $I \in V$ passes a triplet of real numbers $\Pi_{i \rightarrow a} = (\Pi_{i \rightarrow a}^u, \Pi_{i \rightarrow a}^s, \Pi_{i \rightarrow a}^0)$ to each of its clause neighbors $a \in V(i)$:

$$\begin{aligned} \Pi_{i \rightarrow a}^u &= \left[1 - \prod_{b \in V_a^u(i)} (1 - \eta_{b \rightarrow i}) \right] \prod_{b \in V_a^s(i)} (1 - \eta_{b \rightarrow i}), \\ \Pi_{i \rightarrow a}^s &= \left[1 - \prod_{b \in V_a^s(i)} (1 - \eta_{b \rightarrow i}) \right] \prod_{b \in V_a^u(i)} (1 - \eta_{b \rightarrow i}), \\ \Pi_{i \rightarrow a}^0 &= \prod_{b \in V(i) \setminus a} (1 - \eta_{b \rightarrow i}), \end{aligned} \quad (4)$$

where $V_a^u(i)$ and $V_a^s(i)$ denote neighbors which tend to make variable i dissatisfy or satisfy clause a . If a set like $V_a^s(i)$ is empty, the corresponding product takes value 1. In the above equalities, the variable i sends a particular symbol to clause a saying that the variable cannot satisfy the clause (“ u ”), that the variable can satisfy the clause (“ s ”), or that it is indifferent (“ 0 ”).

When SP converges to a fixed point set of messages $\eta_{a \rightarrow i}^*$, one can use it to estimate the statistic characteristic (called bias) of every variable. There are two types of biases: positive bias and negative bias. The positive (negative) bias is the probability that variable i is restricted to 1 (0). The computing formulae are as follows:

$$\begin{aligned} W_i^{(+)} &= \frac{\widehat{\Pi}_i^+}{\widehat{\Pi}_i^+ + \widehat{\Pi}_i^- + \widehat{\Pi}_i^0}, \\ W_i^{(-)} &= \frac{\widehat{\Pi}_i^-}{\widehat{\Pi}_i^+ + \widehat{\Pi}_i^- + \widehat{\Pi}_i^0}, \end{aligned} \quad (5)$$

where

$$\begin{aligned}\widehat{\Pi}_i^+ &= \left[1 - \prod_{a \in V_+(i)} (1 - \eta_{a \rightarrow i}^*) \right] \prod_{a \in V_-(i)} (1 - \eta_{a \rightarrow i}^*), \\ \widehat{\Pi}_i^- &= \left[1 - \prod_{a \in V_-(i)} (1 - \eta_{a \rightarrow i}^*) \right] \prod_{a \in V_+(i)} (1 - \eta_{a \rightarrow i}^*), \quad (6) \\ \widehat{\Pi}_i^0 &= \prod_{a \in V(i)} (1 - \eta_{a \rightarrow i}^*).\end{aligned}$$

As a whole, SP gathers the statistical information which can guide the choosing branches. In choosing branches process, we select branches obeying quantification orders. For variables in the same quantification level, we compute positive and negative biases with SP. Then we fix the biased variable in the same quantification level; that is, one variable has the largest bias $|W_i^{(+)} - W_i^{(-)}|$. Having selected the variable, we assign the variable depending on the current quantification level. If the current quantification level is existential, we assign the biased value to the variable; otherwise we assign the reverse biased value to the variable.

7. Branched Treatment

After choosing branches, we can perform branched treatment on the chosen variable. The purpose of treatment is to decide whether the current branch of search tree is satisfiable. In branched treatment process, we employ some efficient technologies, such as conflict reasoning, conflict driven learning, and satisfiability directed learning.

7.1. Conflict Driven Learning. Conflict driven learning utilizes the knowledge learned from failures in certain search space to help prune search in future spaces. If the current branch of the search tree is not satisfied, we can perform conflict driven learning. It is carried out by the analyze conflict routine in Algorithm 3.

The analyze conflict routine analyzes the current status and brings the search to a new space by backtracking. At first, we explain some terms in the routine. As an algorithm based on DPLL, EHSPQBF is a branch and search procedure. Each branch has a decision level (*dL*). The first branch has decision level 1 and so forth. And variables implied by a decision variable will have the same decision level as the decision variable. Now we introduce the analyze conflict routine. At the beginning, it decides whether there exist conflict clauses in the current QBF formula (line 1). If they exist, we utilize function *gen_clause()* for generating learned clause (line 2). The learned clause can guarantee that the decision level which the algorithm backtracks to is unique. Then function *add_clause()* performs conflict driven learning when the result of deduction is *conflict*, which causes a learned clause constructed and added to the database (line 3). At last, we decide whether the decision level that the algorithm backtracks to 0 (line 4). If it is not 0, EHSPQBF will backtrack to the decision level (line 5). The aim of conflict driven learning is to memorize the conflict branch. If search

```
Routine analyze_conflict()
(1) conflictcl = find_conflict_clause();
(2) newcl = gen_clause (conflict_cl);
(3) add_clause (new_cl);
(4) if (dl (new_cl) ≠ 0)
(5) then backtrack (dl(new_cl));
(6) end
```

ALGORITHM 3: Analyze conflict routine.

```
Routine analyze_SAT()
(1) entity = find_sat_entity();
(2) if (entity = NULL)
(3) then entity = gen_sat_induced_entity();
(4) if (!termnate_condition (entity))
(5) then newen = consensus_gen_entity (entity);
(6) add_newentity (newen);
(7) if (dl (newen) ≠ 0)
(8) then backtrack (dl (newen));
(9) end
```

ALGORITHM 4: Analyze SAT routine.

space has the same branch in the further, we can prune the branch directly, which need not go on searching.

7.2. Satisfiability Directed Learning. Satisfiability directed learning utilizes the knowledge learned from satisfactions to reduce the number of satisfying leaves which need to be visited. When the result of deduction is *satisfaction*, we can perform satisfiability directed learning. It is carried out by the analyze SAT routine in Algorithm 4. At first, we show relevant definitions used in the routine.

Definition 3. (extended CNF) Let $E(x_1, \dots, x_n)$ be a CNF. An extended CNF is $E' = E(x_1, \dots, x_n) \vee EN_1 \vee \dots \vee EN_m$, where EN_i ($1 \leq i \leq m$) is an entity, that is, a conjunction of literals.

Now let us introduce the detail of the routine. At the beginning, it judges whether there exist satisfying entities in the current QBF in extended CNF (line 1). If they exist, the routine goes to line 4; otherwise, a entity from the current variable assignment is generated by function *gen_sat_induced_entity()* (lines 2-3). In lines 4-5, a learned entity is not generated by function *consensus_gen_entity()* until the generated entity meets the following conditions. (1) Among all its universal variables, one and only one has the highest decision level, supposing the variable is V ; (2) V is at a decision level with a universal variable as the decision variable; (3) all existential literals with quantification level smaller than V 's are assigned *true* before V 's decision level. The decision level of universal variable V is the decision level which the algorithm will backtrack to. Then function *add_newentity()* performs satisfiability directed learning (line 6). It occurs when the result of deduction is *satisfaction*, which causes a learned entity constructed and added to the database. At last, we decide whether the decision level of variable V is

TABLE 1: Comparisons of EHSPQBF-EQR, EHSPQBF-HBR, and EHSPQBF.

Instances	EHSPQBF-EQR	EHSPQBF-HBR	EHSPQBF
(3, 60, 80, 5, 0)	10.470	15.390	10.209
(3, 60, 80, 5, 1)	56.342	41.230	40.725
(3, 60, 80, 5, 2)	230.230	153.690	153.436
(3, 60, 80, 5, 3)	1.783	3.467	1.659
(3, 60, 80, 5, 4)	98.456	132.678	87.513
(3, 60, 80, 5, 5)	12.479	30.745	11.873

0 or not (line 7). If it is not 0, EHSPQBF will backtrack to the decision level of variable V (line 5). The aim of satisfiability directed learning is memorizing the satisfied branch. If search space has the same branch in the further, we can decide the branch is satisfiable directly and go on searching the other branch of the universal variable.

8. Experimental Evaluation

In this section, to test the performance of the solver EHSPQBF, three algorithms are compared. Our solver EHSPQBF is written in C++. All experiments are carried out on a DELL PowerEdge 2650 computer with 2 Intel Xeon 2.00 GHz CPU and 1G RAM running RedHat ES3.0. Table 1 presents the results of the comparisons of the solvers. In the table, EHSPQBF-HBR is the solver that removes the hyperbinary resolution rule from EHSPQBF; EHSPQBF-EQR is the solver that removes the equality reduction rule from EHSPQBF. All instances are generated by a random generator. (3, 60, 80, 5, 0) denotes a formula indexing 0 that contains 3 quantification levels, 60 variables, 80 clauses, and each clause length is not more than 5.

From the results of the EHSPQBF, we can see that both hyperbinary resolution rule and equality reduction rule contribute much compared with the results of EHSPQBF-EQR and EHSPQBF-HBR. Consequently the run-time of the algorithms decreases. Therefore, by introducing the hyperbinary resolution rule and equality reduction rule, EHSPQBF algorithm increases the efficiency for solving QBF problem.

9. Conclusions

In this paper, we present a novel QBF solver EHSPQBF. In preprocessing, we adopt the equality reduction and the hyperbinary resolution to simplify the formulae. When choosing branches, the system uses survey propagation to select next branch; when handling branches, the system employs conflict reasoning and conflict driven learning as well as satisfiability directed learning together. Experimental results show that, as a heuristic, SP can choose branches effectively. And other techniques such as conflict driven learning also speed up the solving process. As a whole, EHSPQBF is an efficient QBF solver.

Acknowledgments

This work was fully supported by the National Natural Science Foundation of China (Grant nos. 11226275, 61103136, and 61070084), the Fundamental Research Funds for the Central

Universities (Grant no. 11QNJJ006), the Ministry of Education (Grant no. 20120043120017), the Postdoctoral Science Foundation of China (2012M520658), and the opening fund of top key discipline of computer software and theory in Zhejiang provincial colleges at Zhejiang Normal University (Grant no. ZSDZZZXK37).

References

- [1] J. Rintanen, “Asymptotically optimal encodings of conformant planning in QBF,” in *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pp. 1045–1050, July 2007.
- [2] P. Marin, C. Miller, and B. Becker, “Incremental QBF preprocessing for partial design verification,” in *Proceeding of the 15th International Conference of Theory and Applications of Satisfiability Testing*, pp. 473–474, 2012.
- [3] U. Egly, T. Eiter, H. Tompits, and S. Woltran, “Solving advanced reasoning tasks using quantified Boolean formulas,” in *Proceeding of the 12th AAAI Conference on Artificial Intelligence*, pp. 417–422, 2000.
- [4] G. Pan and M. Y. Vardi, “Optimizing a BDD-based modal solver,” in *Proceedings of the 19th International Conference on Automated Deduction*, pp. 75–89, August 2003.
- [5] I. P. Gent, H. H. Hoos, A. G. D. Rowley, and K. Smyth, “Using stochastic local search to solve quantified Boolean formulae,” in *Proceeding of the Principles and Practice of Constraint Programming*, pp. 348–362, 2003.
- [6] I. Gent, E. Giunchiglia, M. Narizzano, A. Rowley, and A. Tacchella, “Watched data structures for QBF solvers,” in *Proceeding of the 6th International Conference on Theory and Applications of Satisfiability Testing*, pp. 348–355, 2003.
- [7] E. Giunchiglia, M. Narizzano, and A. Tacchella, “QuBE: a system for deciding quantified Boolean formulas satisfiability,” in *Proceeding of the 17th International Joint Conference on Artificial Intelligence*, pp. 18–23, 2001.
- [8] J. Rintanen, “Partial implicit unfolding in the Davis Putnam procedure for quantified Boolean formulae,” in *Proceeding of the 8th International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, pp. 362–376, 2001.
- [9] L. Zhang and S. Malik, “Towards a symmetric treatment of satisfaction and conflicts in quantified Boolean formula evaluation,” in *Proceeding of the Principles and Practice of Constraint Programming*, pp. 200–215, 2002.
- [10] F. Bacchus, “Enhancing Davis Putnam with extended binary clause reasoning,” in *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI ’02)*, pp. 613–619, August 2002.
- [11] M. Mézard, G. Parisi, and R. Zecchina, “Analytic and algorithmic solution of random satisfiability problems,” *Science*, vol. 297, no. 5582, pp. 812–815, 2002.

Research Article

A Topology Structure Based Outer Membrane Proteins Segment Alignment Method

Han Wang,^{1,2} Bo Liu,³ Pingping Sun,^{1,2,4} and Zhiqiang Ma^{1,2}

¹ School of Computer Science and Information Technology, Northeast Normal University, Changchun 130024, China

² Key Laboratory of Intelligent Information Processing of Jilin Universities, Northeast Normal University, Changchun 130117, China

³ School of Physical Education, Northeast Normal University, Changchun 130117, China

⁴ National Engineering Laboratory for Druggable Gene and Protein Screening, Northeast Normal University, Changchun 130024, China

Correspondence should be addressed to Zhiqiang Ma; mazq@nenu.edu.cn

Received 15 July 2013; Accepted 7 September 2013

Academic Editor: William Guo

Copyright © 2013 Han 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.

Outer membrane proteins (OMPs) are transmembrane proteins (TMPs) located in outer membranes. These proteins perform diverse biochemical functions and have immediate medical relevance, so that their spatial structures are important for studying. But the special physicochemical properties of OMP make it hard to obtain their structures experimentally. For the purpose of predicting OMP structures, discriminating OMPs and aligning their sequences to native structures are indispensable steps. We developed a novel method OMSA (Outer Membrane Segment Alignment), which implemented both steps in one program. OMSA integrates OMP-specific topology features to implement a sequence-to-structure alignment, for example, segment type and segment orientation, while a segment-dependent gap penalty model is employed to improve the alignment. Compared to peer top-leading methods, OMSA achieved higher accuracy in both OMP discrimination and alignment, which may further improve OMP structure studying.

1. Introduction

Outer membrane proteins (OMPs) are the transmembrane proteins (TMPs) found in outer membranes of cell, mitochondria, or plastids. They perform diverse biochemical functions such as active ion transport, passive nutrient uptake and intake, or enzymatic activity and structural anchoring [1]. Meanwhile, they also are the potential targets for antimicrobial drugs and vaccines [2–4]. Native structures of OMP are extremely scarce in Protein Data Bank (PDB) [5], where hundreds of OMPs account for no more than 2% of all solved structures [6]. However, there are thousands of OMPs estimated to be existing in the genomic databases currently [7], and the number is increasing by ongoing large-scale sequencing [8].

Computational structure prediction provides a practical approach to bridge the gap between sequences and structures. A few efforts have been made to predict the general structures of membrane proteins [9, 10], but the prediction accuracy remains to be further improved. Some methods that focused

on the family of G-protein-coupled receptors (GPCRs) [11, 12] achieved better performances. However, those methods perform poorly on OMPs for lacking of homologous templates, predicting OMP structures is still a challengeable problem. For the purpose, the above all task is to accurately discriminating the OMP from sequences and aligning them to solved structures.

There are many methods that effort to discriminate the OMP from sequence databases; Gromiha and Suwa used OMP motifs for excluding and identifying OMP form globular proteins [13], and they further compared performances of several machine learning based methods; the highest accuracy is no more than 91% [14]. By contrast, few OMP alignment methods have been reported despite its importance for fold recognition [15]. The structure prediction of OMPs is estimated to obtain the accuracy as high as that of globular proteins under the condition that the alignment of OMPs achieves similar accuracy to its compeer [16], but alignment methods [17–21] for globular protein did not work well on OMPs, the reason being mainly because those

sequence-to-structure methods could not efficiently abstract the OMP-specific features used in the alignment process.

It is notable that OMPs have special physicochemical properties compared to globular proteins. They are composed of several beta strands which form a barrel to span outer membrane. These strands are enriched polar, charged, and hydrophobic residues, antiparallel crossing membrane alternatively, from one side of outer membrane to the other side. In addition, the number of transmembrane (TM) strands is mostly even. These remarkable properties can be clearly represented by topology structure of OMPs. Therefore, topology structure is considered helpful improving the OMP alignment as OMP-specific feature. Currently, several OMP topology structure predictors are available, such as TMBHMM [22], TMBETAPRED-RBF [23], and TMBpro [24]; these methods utilize many OMP-specific features to improve the prediction accuracy, including amino acid composition, alternating hydrophobicity pattern [25], or “positive-inside” rule [26, 27].

In this study, we developed a novel Outer Membrane Segment Alignment method (OMSA) to discriminate and align OMPs from sequences at the same time. The method for the first time used OMP-specific features abstracted from topology structure to improve the alignment, for example, TM segment type and TM segment orientation. Particular scoring function designed for OMP was applied to a local-global dynamic programming to optimize the alignment. Comprehensive OMP datasets have been used for training and testing, and the results represented that OMSA perform well on OMP discrimination and alignment compared to peer methods. Hence, our method will be useful for OMP structure studying.

2. Materials and Methods

2.1. Datasets. Orientations of Proteins in Membranes (OPM) database [28] was used in OMSA training and testing; it provides the most comprehensive collection of membrane proteins with calculated spatial arrangements. Differing to computational-based databases [29, 30], OPM database is more in agreement with the experimental data and further classifies the membrane proteins based on their main transmembrane domains by referencing SCOP [31] and TCDB [32]. In this database, 98 entries are classified to 26 superfamilies, and each of them is composed of one or more protein families. Here, proteins in the same superfamily are evolutionarily related and with superimposable tertiary structures, but in low sequence identity, while it is high among the proteins in the same family. We randomly picked two entries from each superfamily to comprise training and testing datasets, respectively. For those superfamilies which have only one entry, the entries were selected to training dataset. Finally, the training dataset is composed of 19 nonredundant entries, while testing dataset has 28 nonredundant entries (see Table S1 in supplementary material available online at <http://dx.doi.org/10.1155/2013/541359>).

For the purpose of benchmarking the performance of OMP discrimination, Gromiha and Suwa’s dataset (GS-dataset) [13] is used, which includes 377 OMPs, 268 α -helical

transmembrane proteins, and 674 globular protein chains. All these well-annotated transmembrane proteins included in the dataset were obtained from PSORT-B database [33], while those globular protein chains were obtained from the PDB40D_1.37 database of SCOP [34]. In this dataset, a few transmembrane proteins are homologous, and the globular proteins have sequence identity less than 30%.

2.2. Segment Type and Orientation. Segments on an OMP sequence are classified to TM segment (TMB) and non-TM segment according to their location relative to outer membrane. TM segment is the part of sequence that is across the outer membrane, and non-TM segments include outside segment and inside segment, where the outside segment locates outside the area surrendered by outer membrane, while the inside segment locates in the area. The segment orientation describes the direction of those segments across the outer membrane; it can be abstracted when segment types have been identified.

The segment type $\text{seg}(i)$ at sequence position i is given as

$$\text{seg}(i) = \begin{cases} \text{B}, & \text{TMB}, \\ \text{I}, & \text{Inside}, \\ \text{O}, & \text{Outside}, \\ \text{U}, & \text{Unknown}. \end{cases} \quad (1)$$

The orientation of each TM segment is determinated according to the inside/outside it forwards to. TM segment orientation $\text{orn}(i)$ at position i is valuated as follows:

$$\text{orn}(i) = \begin{cases} 1, & \text{from inside to outside}, \\ -1, & \text{from outside to inside}, \\ 0, & \text{else}. \end{cases} \quad (2)$$

Given an OMP as shown in Figure 1(a), segment types and orientations are clearly described by topology structure in Figure 1(b), and they have been labeled for each amino acid on its sequence using (1) and (2), respectively, shown in Figure 1(c). In each pairwise alignment, query sequences used predicted topologies, while the template sequences directly utilized native topology structures obtained from OPM database. We employed TMBHMM [22] to predict topology structures for its high performance and accuracy. The method integrates particular features to build a hidden Markov model (HMM) for topology structure predicting, for example, SASA (relative solvent-accessible surface area) feature [35] and frequency profile [36].

2.3. Sequence Profile. Almost all OMPs are homologous to each other [37], so that the sequence profile is especially conductive to identifying the compatibility within the same segment types by sequence patterns. This sequence-based profile generated by PSI-BLAST [36] is called Position Specific Scoring Matrix (PSSM), which is derived by aligning the amino acid sequence against NCBI’s nonredundant sequence database (NR) with three iterations. The profiles present the evolutionary conservation of sequences by large-scale searching, whereas it has a significant impact on protein fold

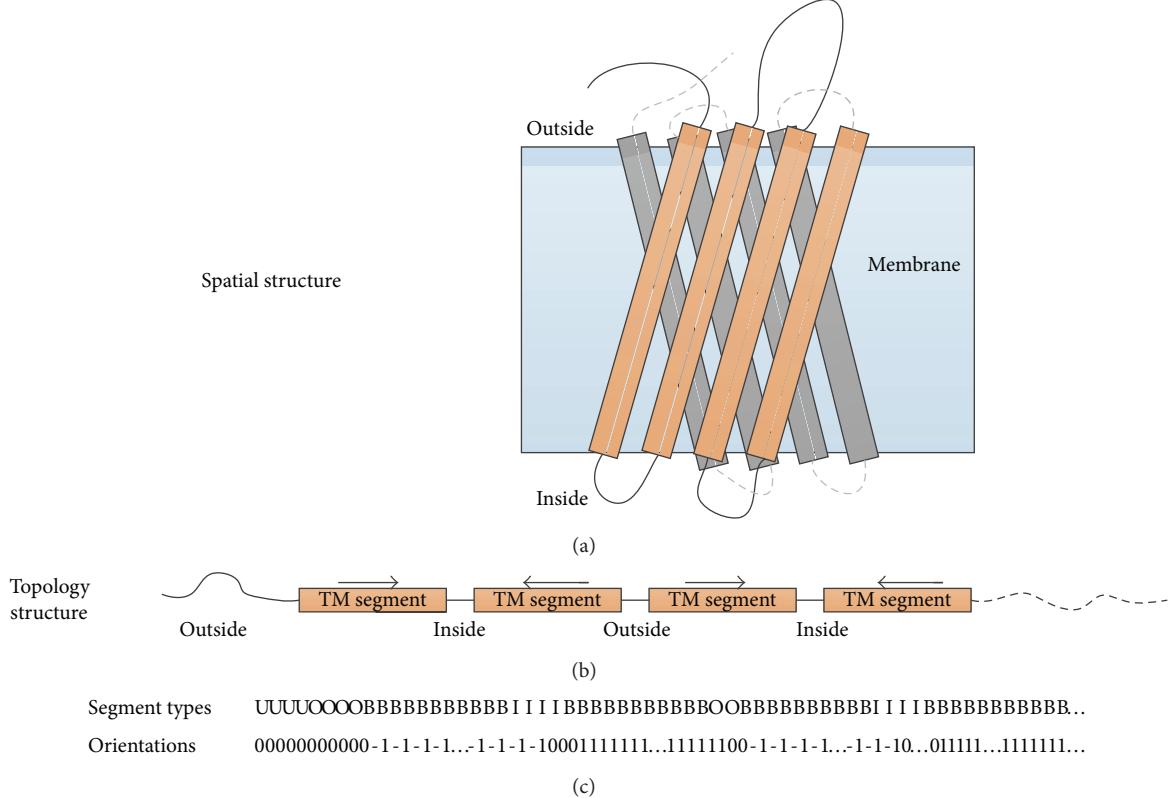


FIGURE 1: The special conformation of OMPs. (a) is the conformation of a sample OMP, and the rectangles represent the TM segments. Its N-terminus locates on the outside. (b) is the topology structure of the protein, which has been unfolded to sequence; the right arrow means that the orientation of the corresponding TM segment is from outside to inside and vice versa. Meanwhile, the non-TM segments alternatively locate on the outside and inside. (c) The protein sequences represented by segment types and orientations according to (1) and (2).

recognition [38]. A PSSM profile $\text{pm}[i, j]$ is a $n \times 20$ log-odds matrix, where the n represents the sequence length. Each element in $\text{pm}[i, j]$ negatively represents the frequency of the residue type j at position i .

2.4. Scoring Function. The scoring function is the kernel of dynamic programming (DP) which is used to derive the optimal path searching for alignment. It is composed of two major parts, fitness scoring and gap penalty model. The fitness scoring responds to measure the compatibility between any residue pair, while the gap penalty model affects the alignment accuracy by controlling the gap insertion. They coordinate with each other making a balance to achieve the best alignment accuracy. Tailing for the OMP, the scoring function used here is different from that of globular proteins, the fitness scoring adopts the OMP-specific features and correspondingly strategies, and the gap penalty is thereby particularly designed to support the OMP segment alignment. Here, the integrated three features, segment type, segment orientation, and sequence profile, compose a compact profile for the profile-to-profile scoring, where segment type guarantees the proteins are aligned by segment and segment orientation prevents the TM segments being incorrectly aligned, while sequence profile further improves the alignment accuracy between non-TM segments. Meanwhile,

a segment-dependent gap model is employed correspondingly.

(1) *Fitness Scoring.* Differing to the sequence-to-sequence alignment method, the profile-to-profile alignment offers the more opportunities to align the proteins according to their overall properties. The so-called profile is the general conception of assemble of the selected features, and those features must be efficient to describe the properties of target proteins and less redundant to decrease the computational complex. The selected three features can overall describe the sequence patterns (by sequence profile) and conformation properties (by segment type and TM segment orientation) of OMPs.

The fitness score of i th position on query sequence and j th position on template sequence is calculated by the equation

$$\begin{aligned} \text{Fitness}(i, j) = & w_1 \text{PRO}(i, j) - w_2 \text{SEG}(i, j) \\ & - w_3 \text{ORN}(i, j) + w_{\text{shift}}, \end{aligned} \quad (3)$$

where $\text{PRO}(i, j)$ is the fitness score of sequence profile, $\text{SEG}(i, j)$ is the segment type fitness score, and $\text{ORN}(i, j)$ is the segment orientation fitness score; w_1 , w_2 and w_3 are the weights of the three fitness scores, while w_{shift} is a to-be-determined constant that avoids the unrelated residues

aligned [39]. The segment type fitness score can be simply computed by

$$\text{SEG}(i, j) = \begin{cases} 2, & \text{if } \text{seg}(i) = \text{seg}(j) = "B", \\ 1, & \text{if } \text{seg}(i) = \text{seg}(j) \neq "B", \\ -1, & \text{else.} \end{cases} \quad (4)$$

Segment orientation fitness score is calculated according to the equation

$$\text{ORN}(i, j)$$

$$= \begin{cases} 1, & \text{orn}(i) = \text{orn}(j) \text{ and } \text{orn}(i) \neq 0 \text{ and } \text{orn}(j) \neq 0, \\ -1, & \text{orn}(i) \neq \text{orn}(j) \text{ and } \text{orn}(i) \neq 0 \text{ and } \text{orn}(j) \neq 0, \\ 0, & \text{else.} \end{cases} \quad (5)$$

The evolution fitness score is calculated according to the equation

$$\text{PRO}(i, j) = \sum_{k=0}^{20} (\text{pm}_{\text{query}}[i, k] \times \text{pm}_{\text{template}}[j, k]), \quad (6)$$

where $\text{pm}_{\text{query}}[i, k]$ is PSSM profile value of residue type k at position i on the target sequence, and $\text{pm}_{\text{template}}[j, k]$ follows the same meaning.

(2) *Gap Penalty.* Gap penalty is used to evaluate the cost of making an insertion (or deletion) option; the balance between fitness score and the gap penalty makes the decisions for the DP to trace the optimal aligning path. Various gap penalty models have been designed for globular protein alignment previously, for example, position-dependent gap penalty models used in [17, 39] or profile-based model [40]. Even a more complicate model [41] was used recently for low-homology protein threading. In this study, we employed a segment-dependent gap penalty model to satisfy the segment alignment, in which insertions in TM segments are more strictly punished compared to non-TM segments, because these segments are more conserved. Considering the predictions accuracy of topology, we do not simply forbid the insertions as H. Zhou and Y. Zhou [42] did for globular protein alignment but allow the gaps insert to query sequence using open-gap-penalty op_{tm} and one-time-penalty ep_{tm} , while they are forbidden to template sequences which use the native topology structures. Similarly, open-gap-penalty $\text{op}_{\text{non-tm}}$ and one-time-penalty $\text{ep}_{\text{non-tm}}$ are used for non-TM segments of query sequence. Here, the open-gap-penalty is used only when the gap opens, and the one-time-penalty is used for the continuous insertion after the gap opened.

2.5. Training Parameters. All parameters, $w_1, w_2, w_3, w_{\text{shift}}, \text{op}_{\text{tm}}, \text{op}_{\text{non-tm}}, \text{ep}_{\text{tm}},$ and $\text{ep}_{\text{non-tm}}$, used in the scoring function are trained using the same method as [42] on our training dataset. All the parameters are randomly assigned the start value and then optimized by grid search. Here, the gold standard TM-score [43] is used to supervise the searching. The higher TM-score derived by aligned sequences is considered the higher accuracy achieved. The iterations exit when the average TM-score does not increase any more.

2.6. Dynamic Programming. DP is the most popular paradigm in computational biology [44]. It is a method for solving complex problems by breaking them down into simpler subproblems and has been applied widely in sequence alignment or optimal searching. Decided by our scoring function, the DP process of OMSA is OMP specific, which can align the correct segments as much as possible. We use local-global algorithm to optimize the alignment path for requirement of segment alignment. By using the OMP-specific scoring function introduced above, the DP procedure of OMSA can find better path for an alignment. The segments with the same type are aligned preferentially, while different segment types are hard to match unless they are extremely compatible with the evolutionary conservation.

3. Results and Discussion

In this section, we will firstly show that the segment orientation significantly improves the alignment accuracy and then compare the OMSA to one of the top-leading generated profile-to-profile alignment methods, HHalign [45]. As the outputs of sequence-to-structure alignment, rawscores generated by OMSA negatively relate to the structural similarities. However, only the rawscores derived by the same query protein are comparable; they cannot be directly used as a criterion to evaluate the structure similarity between different alignment pairs. Thus, we transform the rawscores to rawscore correlation (RC), which bridges the rawscores and the structure similarity by normalizing the rawscores to interval (0, 1]. Finally, the RC is applied to OMP discrimination and the OMP fold recognition.

3.1. Topology Features Improve the Alignment Accuracy. OMSA improves alignment principally relied on topology features. We implemented method nOMSA that removed topology features for comparison, where the nOMSA used the secondary structure and sequence profile as features and trained using the same dataset. We made all-versus-all pairwise alignments on testing dataset using OMSA and nOMSA, respectively; average GDT_TS [46, 47] was used to evaluate the alignment accuracy. GDT_TS scores the structure similarity for two length-equal proteins using their 3D structures. The structure similarity is positively related with the GDT_TS when the score increases from 0 to 1. Therefore, for each alignment pair, the higher GDT_TS means the better alignment that has been made by the method, whereas the overall alignment accuracy of a query protein can be described using the average GDT_TS of all the templates.

Each query, there were aligned to 27 templates, and the average GDT_TS was statistic using the corresponding 27 GDT_TSs. The average GDT_TS of all the 28 queries derived by OMSA are shown using the solid polyline in Figure 2, while the corresponding nOMSA results are shown using dotted polyline, where the solid polyline lies obviously above the dotted polyline, which indicates that all the query proteins will obtain higher alignment accuracy when the topology features are involved.

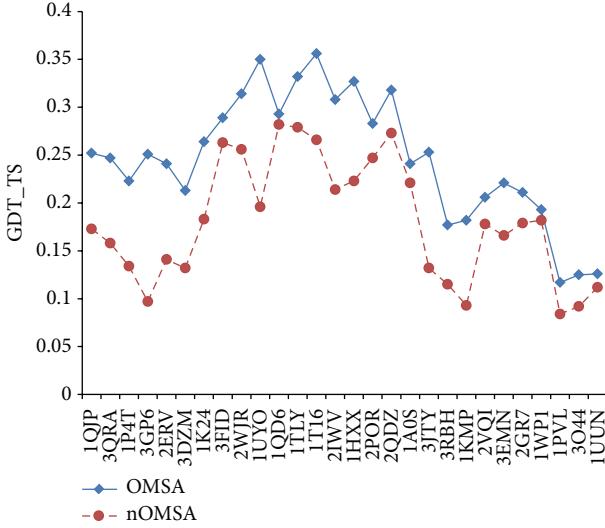


FIGURE 2: Feature of segment orientation improves the alignment accurate. The alignments of OMSA are more accuracy for all the testing proteins shown using solid polyline in blue than those of nOMSA (non-segment-orientation based). The results are derived based on the all-versus-all pairwise alignments on the testing dataset.

3.2. Performance of OMSA Alignment. Generally, comparison with the peer methods powerfully proves the improvements of the new methods; however, there is no such OMP pairwise alignment method currently available; thus, we chose top-leading general alignment method HHalign [45]. HHalign uses profile hidden Markov model (HMM) to make pairwise HMM-HMM (profile-HMM) alignments, and the confidence values and a full seven-state secondary structure are employed to improve the alignment quality; thus, it is a very sensitive repeat-identification tool.

To arrange the comparison, the profile-HMMs of 28 proteins of testing dataset were generated to make the all-versus-all pairwise alignment, in which the default parameters were used. Correspondently, the previous pairwise alignment results derived by OMSA are used for the comparison. Avoiding the pseudo increasing of alignment accuracy, those pairs which the queries align to themselves were excluded; finally, in total 28×27 pairs of alignments were used for comparison.

The alignment accuracy can be evaluated by two criterions: (1) calculating the percentage of correctly aligned positions [15] and (2) scoring the structural similarity between the aligned pairs [48]. For the first approach, each aligned residue pair in the alignment has to be verified whether it belongs to the correct alignment, whereas a golden standard of structural alignment is required, such as widely used method TM-align [49], since there is no unique solution that solves the problem finding the optimal structure alignment [50]. For the second one, structure similarity of alignment is scored directly by their native 3D structures of aligned parts, depicting whether they were correctly aligned or not; GDT_TS [46, 47] and TM-score [43] are commonly used for

the purpose. Notably, TM-score is designed to be independent of protein lengths, and the structures with a score higher than 0.5 assume roughly the same fold [51], while it indicates that the proteins are unrelated when the score is below 0.20. To comprehensively show the performances, we adopted both approaches to exhibit the alignment results, where the alignment accuracy (ACC) was used according to approach (1) and TM-score and GDT_TS were used according to approach (2).

As shown in Table 1, OMSA derived the better alignment in average against the testing dataset compared to the HHalign. Here, the accuracy of the TM segments and the non-TM segments is counted separately, and the overall accuracy is also given. The alignment accuracy is obtained using the TM-align as the standard. The HHalign aligns both types of segments with almost the same accuracy, while the OMSA shows much difference in accuracy between them. There is more than eleven percentage margin that the OMSA surpasses the HHalign in the alignment accuracy of the TM parts, while the margin is just no more than three percent of the non-TM parts; for this reason, the OMSA achieves the 50.6% of overall alignment accuracy and surpasses its companion nearly eight percent. The result is consistent with the TM-score and the GDT_TS, where the alignments for TM segments are much better than non-TM segments. Correspondingly, the improvement of TM-score achieves the eight percent, as well nine percent improvement of GDT_TS.

Comparing the integrated features, sequence profile has been commonly used in both methods, but the HHalign addresses the predicted secondary structure in alignment, while OMSA uses topology-based features instead. Obviously, the utilization of topology structure based features improves alignment accuracy than secondary structure, especially for the segments alignment methods. However, it lacks to determinate the structures similarity for non-TM segments which have secondary structures and decreases the alignment accuracy for non-TM segments, but this shortness has been obscured by the less secondary structures existing outside TM segments. The results illuminate that the TMB orientation improves the OMP segment alignment by decreasing the misleading of the incorrect topology prediction and further distinguishing the TM segments and thereby improves the alignment performance.

3.3. Alignment Rawscore and Structural Similarity. The OMSA generates a rawscore for each alignment pair, and its value directly reflects the structure similarity between the proteins. The aligned pair with smaller rawscore illustrates that the two proteins are more likely having similar spatial conformations, which is exhibited in Figure 3 using the alignment results of Porin (PDB_ID:2POR) [52].

The left-top point represents the alignment result of 2POR with itself, so that it totally matched in structure which is presented by GDT_TS of value 1 and derived the smallest negative rawscore at the same time. The nearest point represents protein OmpF porin (PDB_ID:1HXX), which belongs to the same superfamily but not the same family with 2POR. The two proteins have the most similar conformations with each other within the testing dataset. In similar manner, the points in the left-top area are mostly the proteins that have similar

TABLE 1: Comparison of alignment accuracy with HHalign. The performances are compared separately within TM segments, non-TM segments, and overall sequence. OMSA achieved the best alignment accuracy in all the fields, and that within the TM segments has significantly surpassed the general alignment program HHalign. It indicates that the segment orientation further improves the alignment for OMPs.

Methods	Acc (%)			TM-score			GDT_TS		
	TM	Non-TM	Overall	TM	Non-TM	Overall	TM	Non-TM	Overall
OMSA	53.3	45.1	50.6	0.403	0.314	0.322	0.335	0.295	0.307
HHalign	42.9	42.6	42.7	0.217	0.256	0.242	0.209	0.226	0.216

TM: TM segments; Non-TM: non-TM segments; Acc: accuracy.

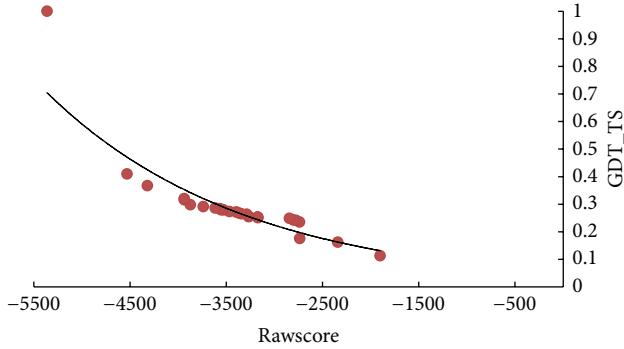


FIGURE 3: An example of 2POR shows rawscores negatively related to structure similarity. The structure similarity is represented using GDT_TS. The rawscores generated by OMSA negatively relate to the 3D structure similarity between the aligned proteins, where a smaller score indicates that the corresponding protein is similar to the query protein in spatial conformation.

structures with 2POR (16 TM segments each chain), such as Fatty Acid Transporter FadL (PDB_ID:1T16, 14 TM segments) [53], Porin OmpG (PDB_ID:2IWV, 14 TM segments) [54], and BenF-like porin (PDB_ID:3JTY, 18 TM segments) [55]. These proteins are similar to the topology structures, but it is not the only driving force for OMSA to recognize the proteins; even the difference among the same-type segments can be further distinguished, such as Alginate export protein (PDB_ID:3RBH) [56], which is reported recently with 18 TM segments, resembling to 2POR in topology structure but significantly differing in non-TM segments. Most proteins centralized in the middle area superimpose their 3D structures to 2POR partially, whereas they derived higher rawscores, while the other proteins fell to the right-bottom corner are totally different with 2POR, such as Leukocidin F (PDB_ID:1PVL) [57] and OprM (PDB_ID:1WP1) [58]. All the testing proteins showed similar rawscore distribution as 2POR; the results illuminate that the rawscore of OMSA negatively responds to the structure similarity.

3.4. Rawscore Correlations. Since the rawscores produced by OMSA are negatively related to the protein structural similarity, they are able to be used to discriminate the OMPs from the globular proteins. However, the comparison of the rawscores can be done only among those produced by aligning to the same query protein, otherwise, the rawscores are not comparable. The reason is easy to understand; assuming

aligning two pairs of proteins at the same time, proteins in one pair have short sequences and very similar structures, while their companions have much longer sequences and less similar structures; even when the proteins in shorter pair are perfectly aligned, they still may not obtain the smaller rawscore; it is decided by the alignment between the other two proteins. Considering the homologous among the OMPs, two proteins that have similar sequence length tend to have more similar local structures, so that proteins in longer pair have more chance to derive the smaller rawscore. Therefore, the rawscore cannot be directly used for the purpose.

It has been noticed that each query protein will achieve the best rawscore aligning to itself; the other rawscores are all smaller than the value and decrease almost according to the structure similarity, so that rawscore correlation (RC) between query protein and templates can be described by the relative values of rawscores. The RC of query protein i and template protein j is calculated using the equation

$$RC(i, j) = \frac{rs(i, j)}{rs(i, i)}, \quad (7)$$

where $rs(i, j)$ and $rs(i, i)$ are, respectively, the alignment rawscores of responding proteins. All the rawscores are negative numbers, so the RC is a positive number and belongs to $(0, 1]$. With the transformation, the rawscores are kind of normalized. The ranking of rawscores has not been changed for each query protein, but the structure similarity between the different query-template pairs becomes comparable.

To exhibit the correlation between rawscores and structures, all the testing pairwise alignment rawscores were transformed to RC value, each blue point in Figure 4 presents a RC value, and each column is composed of 28 points, which are the RCs of 28 templates (including itself). All the proteins are best matched with themselves, so the RC values are always 1 on the top of each column accordingly, and the others are smaller than 1. For the reason that we did not change the orders of rawscores, the RC values distribute similarly with rawscores, the primary function of the diagram is to present the commonness among the columns. It can be found that most RC values of each column concentrate between 0.45 and 0.8, and fewer points scatter above or under the region. The phenomenon indicate that most templates have some parts obviously similar with query protein in conformation, and fewer templates are significantly similar or different to that. Meanwhile, the last three columns which correspond to Leukocidin F (PDB_ID:1PVL) [57], Cytolysin and hemolysin HlyA Pore-forming toxin (PDB_ID:3O44) [59], and Porin

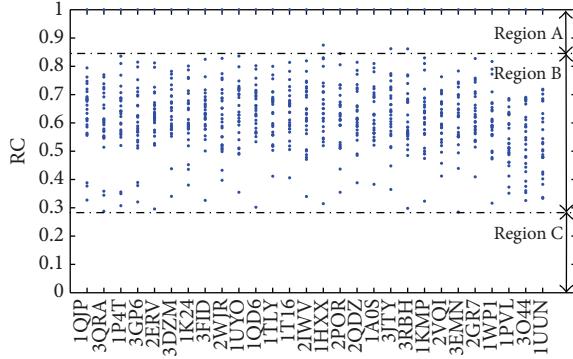


FIGURE 4: The rawscore correlations of testing proteins. All the RC values of the testing proteins are pointed in blue and shown in separate columns. Those $RC = 1$ points respond to the alignments of queries with themselves. According to the 3D structure similarity, RC values of testing proteins fall into three different regions: region A (proteins are in the same superfamily), region B (proteins are OMPs), and region C (proteins are not OMP). Therefore, RC can be further used to measure the similarity of 3D structures instead of rawscore.

MspA (PDB_ID:IUUN) [60] show the RC distributions slightly different that the average RC value smaller than the other proteins. The three proteins are all special OMPs that have particular conformations different from other proteins, so that the OMSA could not obtain the better alignment for them. It is also the partial reason that result the smaller RC values in other columns. The distribution illuminates the RC value responds to the structure similarity between the query and templates.

Although the structure differences exist among the OMPs, the RC values still remain bigger than 0.28 with each other, which indicates that they have common properties at least in some parts. Meanwhile, the scanty two superfamilies that have two proteins each are observed from Figure 4; all the four proteins derive RC values bigger than 0.84 and are the top 4 among the alignments between different query and template. According to the results, we divided the RC value into three intervals: region A: $(0.84, 1]$, region B: $(0.28, 0.84]$, and region C: $(0, 0.28]$. The proteins in region A are considered to share the same superfamily with corresponding query protein, those in region B can be regarded as OMPs, and those in region C are the non-OMPs. Because the OMSA effectively amplified the OMP-specific features, the non-OMPs are hard to achieve a bigger RC value than 0.28. The performance of discriminating OMPs is shown in the next section.

3.5. Improving OMP Discrimination. We adopted the GS-dataset to the benchmark the performance of OMSA to discriminate the OMPs, because the dataset was wildly used for the purpose as benchmark by the other OMP discrimination methods, such as DD [13], NN_AAC [14], and SVM_AAC_DPC [61]. As known, the testing dataset is nonredundant and covers all the OMP superfamilies; we used it as the OMP library. For each protein in GS-dataset, we calculated its RC values aligning to all the proteins of our

TABLE 2: Performance of OMP discrimination methods.

Methods	MCC	Acc (%)	SN (%)	SP (%)
OMSA	0.896	95.4	86.4	99.8
DD	0.541	82.4	78.8	83.3
NN_ACC	0.716	91.0	79.3	93.8
SVM_ACC_DPC	0.816	93.9	90.9	94.7

The results are respectively cited from [13, 14, 61]. MCC: Matthews correlation coefficient; Acc: accuracy; SN: sensitivity, SP: specificity.

OMP library; if the 26 of 28 RC values are bigger than 0.28, the corresponding protein is considered as the OMP.

To compare the discrimination accuracy, four criterions were used to show the performance: Matthews correlation coefficient (MCC), accuracy (AC), sensitivity (SN), and specificity (SP). As shown in Table 2, OMSA achieves the higher accuracy than DD, NN_ACC, and SVM_ACC_DPC in MCC, AC, and SP. There were 2 false positives (FPs) and 59 false negatives (FNs) in the results of OMSA. It is obvious that our method is highly reliable to determinate OMPs with 95.4% of AC, accompanied by 0.896 of MCC, which is much higher than the second top one of 0.816. The FNs decreased the accuracy of OMSA and also resulted in that the SN was slightly lower than that of SVM_AAC_DPC, but that can be improved by further optimizing the discriminating threshold of RC. Therefore, OMSA has the potential abilities to achieve higher accuracy of OMP discrimination. Notably, it is hard to discriminate the transmembrane strands and beta water soluble proteins for most methods, because both of them share some common features, such as amphipathicity [62]. Benefited by the predicted topology structure, most all-beta water soluble proteins showed no TM segments, whereas the OMSA has been never confused by the two kinds of proteins.

3.6. Improving Fold Recognition. The pairwise alignment results of the testing dataset are used to show the fold recognition performance of our method. For the comparison, we applied top-leading general fold recognition program HHsearch [45] to the same dataset. As mentioned, the testing dataset is redundant but small, all the proteins included are at the fold level, there are only two four proteins at the superfamily level, and no proteins at the family level, whereas the comparison conducted here is slightly different from that among the globular proteins, where the top 3 recognized folds are enough to show the performance on such a small dataset, and the recognition accuracy is compared using the average TM-score. Differing to the usage in alignment accuracy, TM-scores are obtained here using the native structures of the query-template pairs, instead of aligned parts, to show the native structure similarity of recognized folds. To further show the performance, the accuracy of recognizing the best one is exhibited, respectively. As shown in Table 3, OMSA derived the best performance comparing to HHsearch; it recognized the most similar folds for the queries with accuracy of 57.1%, while HHsearch achieved the accuracy above ten percent less, and the results are consistent with native TM-scores. The accuracy with which best protein fold appears within the top 2 recognized folds increases to

TABLE 3: Comparison of the performance of fold recognition for OMPs with HHsearch.

Methods	Top 1		Top 2		Top 3	
	Acc. (%)	TM-score	Acc. (%)	TM-score	Acc. (%)	TM-score
OMSA	57.1	0.621	66.7	0.576	76.7	0.491
HHsearch	46.4	0.553	60.0	0.512	66.7	0.407

Acc.: accuracy.

66.7% of OMSA, and that remains ten percent more than HHsearch in the top 3.

4. Conclusions

This paper describes a novel segment OMP alignment method, OMSA, which is designed based on OMP-specific features. OMPs have distinct physicochemical properties compared to globular proteins, which is the biggest obstacle for other methods to predict their structures but provides opportunities for our method on the contrary. We extract the segment type and segment orientation as the features from topology structures, combining with sequence profile to comprise the aligning profile, and segment alignment is firstly employed to address the problem. Correspondingly, we reassign the scoring functions to satisfy the requirement, in which the residue-residue compatibility is scored using the OMP-specific features and a segment-dependent gap penalty model is employed. OMSA has been tested based on a nonredundant testing dataset. Compared to HHalign, our method performs well as a sequence-to-structure alignment method, where the feature of segment orientation indispensably improves the alignment accuracy, especially for TM segments. The alignment rawscore of OMSA is observed negatively relating to the structure similarity, so that the method surpassed HHsearch certain percentage of accuracy in fold recognition. Furthermore, RC value which is derived from the rawscore can be directly used to measure the structure similarity between query and template pairs. By using the RC value, OMSA achieved the best accuracy of OMP discrimination compared to other existing methods. For the same reason, the RC can also be applied to the fold recognition by further adjusting the threshold. This state-of-the-art technology will thereby facilitate the discriminating, structure recognizing, and function predicting for OMPs.

Conflict of Interests

The authors declare that they have no financial and personal relationships with other people or organizations that can inappropriately influence their work; there is no professional or other personal interest of any nature or kind in any product, service, and/or company that could be construed as influencing the position presented in, or the review of, this paper.

References

- [1] M. Remmert, D. Linke, A. N. Lupas, and J. Söding, "HHomp—prediction and classification of outer membrane proteins," *Nucleic Acids Research*, vol. 37, no. 2, pp. W446–W451, 2009.
- [2] R. Jackups Jr. and J. Liang, "Interstrand pairing patterns in β -barrel membrane proteins: the positive-outside rule, aromatic rescue, and strand registration prediction," *Journal of Molecular Biology*, vol. 354, no. 4, pp. 979–993, 2005.
- [3] S. Galdiero, M. Galdiero, and C. Pedone, " β -barrel membrane bacterial proteins: structure, function, assembly and interaction with lipids," *Current Protein and Peptide Science*, vol. 8, no. 1, pp. 63–82, 2007.
- [4] R. Pajón, D. Yero, A. Lage, A. Llanes, and C. J. Borroto, "Computational identification of beta-barrel outer-membrane proteins in *Mycobacterium tuberculosis* predicted proteomes as putative vaccine candidates," *Tuberculosis*, vol. 86, no. 3-4, pp. 290–302, 2006.
- [5] H. M. Berman, "The Protein Data Bank: a historical perspective," *Acta Crystallographica A*, vol. 64, no. 1, pp. 88–95, 2007.
- [6] H. M. Berman, T. N. Bhat, P. E. Bourne et al., "The Protein Data Bank and the challenge of structural genomics," *Nature Structural Biology*, vol. 7, pp. 957–959, 2000.
- [7] W. C. Wimley, "The versatile β -barrel membrane protein," *Current Opinion in Structural Biology*, vol. 13, no. 4, pp. 404–411, 2003.
- [8] S. Yooseph, G. Sutton, D. B. Rusch et al., "The Sorcerer II Global Ocean Sampling expedition: expanding the universe of protein families," *PLoS Biology*, vol. 5, no. 3, article e16, 2007.
- [9] S. Kelm, J. Shi, and C. M. Deane, "MEDELLER: homology-based coordinate generation for membrane proteins," *Bioinformatics*, vol. 26, no. 22, pp. 2833–2840, 2010.
- [10] K. Arnold, L. Bordoli, J. Kopp, and T. Schwede, "The SWISS-MODEL workspace: a web-based environment for protein structure homology modelling," *Bioinformatics*, vol. 22, no. 2, pp. 195–201, 2006.
- [11] A. A. Canutescu, A. A. Shelenkov, and R. L. Dunbrack Jr., "A graph-theory algorithm for rapid protein side-chain prediction," *Protein Science*, vol. 12, no. 9, pp. 2001–2014, 2003.
- [12] M. Michino, J. Chen, R. C. Stevens, and C. L. Brooks III, "FoldGPCR: structure prediction protocol for the transmembrane domain of G protein-coupled receptors from class A," *Proteins*, vol. 78, no. 10, pp. 2189–2201, 2010.
- [13] M. M. Gromiha and M. Suwa, "A simple statistical method for discriminating outer membrane proteins with better accuracy," *Bioinformatics*, vol. 21, no. 7, pp. 961–968, 2005.
- [14] M. M. Gromiha and M. Suwa, "Discrimination of outer membrane proteins using machine learning algorithms," *Proteins*, vol. 63, no. 4, pp. 1031–1037, 2006.
- [15] Y. Hu, X. Dong, A. Wu, Y. Cao, L. Tian, and T. Jiang, "Incorporation of local structural preference potential improves fold recognition," *PLoS ONE*, vol. 6, no. 2, Article ID e17215, 2011.
- [16] L. R. Forrest, C. L. Tang, and B. Honig, "On the accuracy of homology modeling and sequence alignment methods applied to membrane proteins," *Biophysical Journal*, vol. 91, no. 2, pp. 508–517, 2006.

- [17] S. Liu, C. Zhang, S. Liang, and Y. Zhou, "Fold recognition by concurrent use of solvent accessibility and residue depth," *Proteins*, vol. 68, no. 3, pp. 636–645, 2007.
- [18] K. Ellrott, J.-T. Guo, V. Olman, and Y. Xu, "Improvement in protein sequence-structure alignment using insertion/deletion frequency arrays," *Computational Systems Bioinformatics*, vol. 6, pp. 335–342, 2007.
- [19] J.-Y. Yang and X. Chen, "Improving taxonomy-based protein fold recognition by using global and local features," *Proteins*, vol. 79, no. 7, pp. 2053–2064, 2011.
- [20] D. Mittelman, R. Sadreyev, and N. Grishin, "Probabilistic scoring measures for profile-profile comparison yield more accurate short seed alignments," *Bioinformatics*, vol. 19, no. 12, pp. 1531–1539, 2003.
- [21] G. Wang and R. L. Dunbrack Jr., "Scoring profile-to-profile sequence alignments," *Protein Science*, vol. 13, no. 6, pp. 1612–1626, 2004.
- [22] N. K. Singh, A. Goodman, P. Walter, V. Helms, and S. Hayat, "TMBHMM: a frequency profile based HMM for predicting the topology of transmembrane beta barrel proteins and the exposure status of transmembrane residues," *Biochimica et Biophysica Acta*, vol. 1814, no. 5, pp. 664–670, 2011.
- [23] Y.-Y. Ou, S.-A. Chen, and M. M. Gromiha, "Prediction of membrane spanning segments and topology in β -barrel membrane proteins at better accuracy," *Journal of Computational Chemistry*, vol. 31, no. 1, pp. 217–223, 2010.
- [24] A. Randall, J. Cheng, M. Sweredoski, and P. Baldi, "TMBpro: secondary structure, β -contact and tertiary structure prediction of transmembrane β -barrel proteins," *Bioinformatics*, vol. 24, no. 4, pp. 513–520, 2008.
- [25] M. Punta, L. R. Forrest, H. Bigelow, A. Kernytsky, J. Liu, and B. Rost, "Membrane protein prediction methods," *Methods*, vol. 41, no. 4, pp. 460–474, 2007.
- [26] G. Heijne, "The distribution of positively charged residues in bacterial inner membrane proteins correlates with the transmembrane topology," *The EMBO Journal*, vol. 5, pp. 3021–3027, 1986.
- [27] G. Von Heijne, "Membrane protein structure prediction. Hydrophobicity analysis and the positive-inside rule," *Journal of Molecular Biology*, vol. 225, no. 2, pp. 487–494, 1992.
- [28] M. A. Lomize, A. L. Lomize, I. D. Pogozheva, and H. I. Mosberg, "OPM: orientations of proteins in membranes database," *Bioinformatics*, vol. 22, no. 5, pp. 623–625, 2006.
- [29] G. E. Tusnády, Z. Dosztányi, and I. Simon, "PDB-TM: selection and membrane localization of transmembrane proteins in the protein data bank," *Nucleic Acids Research*, vol. 33, pp. D275–D278, 2005.
- [30] G. E. Tusnády, L. Kalmár, and I. Simon, "TOPDB: topology data bank of transmembrane proteins," *Nucleic Acids Research*, vol. 36, no. 1, pp. D234–D239, 2008.
- [31] A. Andreeva, D. Howorth, J.-M. Chandonia et al., "Data growth and its impact on the SCOP database: new developments," *Nucleic Acids Research*, vol. 36, no. 1, pp. D419–D425, 2008.
- [32] M. H. Saier Jr., M. R. Yen, K. Noto, D. G. Tamang, and C. Elkan, "The transporter classification database: recent advances," *Nucleic Acids Research*, vol. 37, no. 1, pp. D274–D278, 2009.
- [33] J. L. Gardy, C. Spencer, K. Wang et al., "PSORT-B: improving protein subcellular localization prediction for Gram-negative bacteria," *Nucleic Acids Research*, vol. 31, no. 13, pp. 3613–3617, 2003.
- [34] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, "SCOP: a structural classification of proteins database for the investigation of sequences and structures," *Journal of Molecular Biology*, vol. 247, no. 4, pp. 536–540, 1995.
- [35] Y. Park and V. Helms, "On the derivation of propensity scales for predicting exposed transmembrane residues of helical membrane proteins," *Bioinformatics*, vol. 23, no. 6, pp. 701–708, 2007.
- [36] S. F. Altschul, T. L. Madden, A. A. Schäffer et al., "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic Acids Research*, vol. 25, no. 17, pp. 3389–3402, 1997.
- [37] T. Arnold, M. Poynor, S. Nussberger, A. N. Lupas, and D. Linke, "Gene duplication of the eight-stranded β -barrel OmpX produces a functional pore: a scenario for the evolution of transmembrane β -barrels," *Journal of Molecular Biology*, vol. 366, no. 4, pp. 1174–1184, 2007.
- [38] G. Yona and M. Levitt, "Within the twilight zone: a sensitive profile-profile comparison tool based on information theory," *Journal of Molecular Biology*, vol. 315, no. 5, pp. 1257–1275, 2002.
- [39] S. Wu and Y. Zhang, "MUSTER: improving protein sequence profile-profile alignments by using multiple sources of structure information," *Proteins*, vol. 72, no. 2, pp. 547–556, 2008.
- [40] W. Zhang, S. Liu, and Y. Zhou, "SP5: improving protein fold recognition by using torsion angle profiles and profile-based gap penalty model," *PLoS ONE*, vol. 3, no. 6, article e2325, 2008.
- [41] J. Peng and J. Xu, "Low-homology protein threading," *Bioinformatics*, vol. 26, no. 12, pp. i294–i300, 2010.
- [42] H. Zhou and Y. Zhou, "Fold recognition by combining sequence profiles derived from evolution and from depth-dependent structural alignment of fragments," *Proteins*, vol. 58, no. 2, pp. 321–328, 2005.
- [43] Y. Zhang and J. Skolnick, "Scoring function for automated assessment of protein structure template quality," *Proteins*, vol. 57, no. 4, pp. 702–710, 2004.
- [44] R. Giegerich, "A systematic approach to dynamic programming in bioinformatics," *Bioinformatics*, vol. 16, no. 8, pp. 665–677, 2000.
- [45] J. Söding, "Protein homology detection by HMM-HMM comparison," *Bioinformatics*, vol. 21, no. 7, pp. 951–960, 2005.
- [46] A. Zemla, C. Venclovas, J. Moult, and K. Fidelis, "Processing and analysis of CASP3 protein structure predictions," *Proteins*, supplement 3, pp. 22–29, 1999.
- [47] A. Zemla, Č. Venclovas, J. Moult, and K. Fidelis, "Processing and evaluation of predictions in CASP4," *Proteins*, vol. 45, no. 5, pp. 13–21, 2001.
- [48] F. Teichert, J. Minning, U. Bastolla, and M. Porto, "High quality protein sequence alignment by combining structural profile prediction and profile alignment using SABER-TOOTH," *BMC Bioinformatics*, vol. 11, article 251, 2010.
- [49] Y. Zhang and J. Skolnick, "TM-align: a protein structure alignment algorithm based on the TM-score," *Nucleic Acids Research*, vol. 33, no. 7, pp. 2302–2309, 2005.
- [50] A. Godzik, "The structural alignment between two proteins: is there a unique answer?" *Protein Science*, vol. 5, no. 7, pp. 1325–1338, 1996.
- [51] J. Xu and Y. Zhang, "How significant is a protein structure similarity with TM-score = 0.5?" *Bioinformatics*, vol. 26, no. 7, Article ID btq066, pp. 889–895, 2010.
- [52] M. S. Weiss and G. E. Schulz, "Structure of porin refined at 18 Å resolution," *Journal of Molecular Biology*, vol. 227, no. 2, pp. 493–509, 1992.

- [53] B. Van Den Berg, P. N. Black, W. M. Clemons Jr., and T. A. Rapoport, "Crystal structure of the long-chain fatty acid transporter FadL," *Science*, vol. 304, no. 5676, pp. 1506–1509, 2004.
- [54] Ö. Yıldız, K. R. Vinothkumar, P. Goswami, and W. Kühlbrandt, "Structure of the monomeric outer-membrane porin OmpG in the open and closed conformation," *EMBO Journal*, vol. 25, no. 15, pp. 3702–3713, 2006.
- [55] P. Sampathkumar, F. Lu, X. Zhao et al., "Structure of a putative BenF-like porin from *Pseudomonas fluorescens* Pf-5 at 2.6 Å resolution," *Proteins*, vol. 78, no. 14, pp. 3056–3062, 2010.
- [56] J. C. Whitney, I. D. Hay, C. Li et al., "Structural basis for alginate secretion across the bacterial outer membrane," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 32, pp. 13083–13088, 2011.
- [57] J.-D. Pédelacq, L. Maveyraud, G. Prévost et al., "The structure of a *Staphylococcus aureus* leucocidin component (LukF-PV) reveals the fold of the water-soluble species of a family of transmembrane pore-forming toxins," *Structure*, vol. 7, no. 3, pp. 277–287, 1999.
- [58] H. Akama, M. Kanemaki, M. Yoshimura et al., "Crystal structure of the drug discharge outer membrane protein, OprM, of *Pseudomonas aeruginosa*: dual modes of membrane anchoring and occluded cavity end," *Journal of Biological Chemistry*, vol. 279, no. 51, pp. 52816–52819, 2004.
- [59] S. De and R. Olson, "Crystal structure of the *Vibrio cholerae* cytolysin heptamer reveals common features among disparate pore-forming toxins," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 18, pp. 7385–7390, 2011.
- [60] K. A. Bunting, S. M. Roe, and L. H. Pearl, "Structural basis for recruitment of translesion DNA polymerase Pol IV/DinB to the β-clamp," *EMBO Journal*, vol. 22, no. 21, pp. 5883–5892, 2003.
- [61] K.-J. Park, M. M. Gromiha, P. Horton, and M. Suwa, "Discrimination of outer membrane proteins using support vector machines," *Bioinformatics*, vol. 21, no. 23, pp. 4223–4229, 2005.
- [62] P. G. Bagos, T. D. Liakopoulos, I. C. Spyropoulos, and S. J. Hamodrakas, "A Hidden Markov Model method, capable of predicting and discriminating β-barrel outer membrane proteins," *BMC Bioinformatics*, vol. 5, article 29, 2004.

Research Article

Model Predictive Control Based on Kalman Filter for Constrained Hammerstein-Wiener Systems

Man Hong^{1,2} and Shao Cheng¹

¹ Institute of Advanced Control Technology, Dalian University of Technology, Dalian 116024, China

² Dalian Institute of Science and Technology, Dalian 116052, China

Correspondence should be addressed to Man Hong; manhongg@126.com

Received 6 August 2013; Accepted 22 September 2013

Academic Editor: William Guo

Copyright © 2013 M. Hong and S. Cheng. 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.

To precisely track the reactor temperature in the entire working condition, the constrained Hammerstein-Wiener model describing nonlinear chemical processes such as in the continuous stirred tank reactor (CSTR) is proposed. A predictive control algorithm based on the Kalman filter for constrained Hammerstein-Wiener systems is designed. An output feedback control law regarding the linear subsystem is derived by state observation. The size of reaction heat produced and its influence on the output are evaluated by the Kalman filter. The observation and evaluation results are calculated by the multistep predictive approach. Actual control variables are computed while considering the constraints of the optimal control problem in a finite horizon through the receding horizon. The simulation example of the CSTR tester shows the effectiveness and feasibility of the proposed algorithm.

1. Introduction

Temperature control of the batch reactor during chemical processes [1] is the key to producing qualified chemical products. The control requirement is to change the temperature in the reactor with the given temperature process curve set up in advance for the entire batch reaction process. The accuracy of following the temperature change is the scale used to judge the performance of the designed controller. Two factors affect temperature change. First is the change of the interlayer cooling capacity, where the size reflects the input change of the control system. Second is the heat production issue of the chemical reaction in the reactor with a variation in size that cannot be measured generally because of uncertain, strong, and nonlinear characteristics, raising higher requirements for the design of the controller.

The Hammerstein-Wiener model can describe the processes in the continuous stirred tank reactor (CSTR), pH neutralization reaction process, and other nonlinear chemical processes. Many research results were achieved in the nonlinear model predictive control (NMPC) [2–4]. Considering that the state of the actual system is usually unpredictable, the literature [5] presents a new two-step NMPC algorithm that

constrains the output feedback of the Hammerstein system. The literature [6] designs the predictive controller of the batch reaction based on the Kalman filter; the DMC is used as the framework and is combined with the size of the interference evaluated by the Kalman filter. Its impact on the system output for the problems regarding uncertainty and strong nonlinear characteristics is reflected by the system during the heat production of the batch reaction itself. However, no reasonable method was found to resolve the complex relationship among the states of the practical nonlinear system, the constraints of the intermediate mass, and the impact of the interference signal on the output of the system.

Considering the Hammerstein-Wiener system of state, output, and intermediate mass constraints, this paper proposes the predictive control algorithm of the batch reaction based on the Kalman filter for the constrained Hammerstein-Wiener system by combining the literature [5, 6]. The paper designs the output feedback optimal control law of the linear subsystem with the state observer, and it evaluates the size of interference and its impact on the system output with the linear Kalman filter. By predicting the solving error of the nonlinear algebraic equation and by roll optimizing the error performance indicator of the finite horizon, the

relationship between the outputs calculates the actual control variables. The simulation study of the CSTR proves that the control method in this paper overcomes the impact of the heat production in the reaction in terms of the accuracy of temperature tracking and improves the control effect to a certain extent.

2. Description of Batch Reaction Process and the Hammerstein-Wiener System

2.1. Description of Batch Reaction Process [1]. The batch reactor process is generally operated under a closed reactor condition so that the reactants are added to the reactor once to react in the given condition. The intensifying reaction will cause the heat to accumulate and the temperature to rise and further intensify. Obviously, this process, as well as the interlayer temperature, is unsteady. As the control process is complex, the batch reactor control system is generally required to design the cascade structure, in which the main circuit of the outer ring takes the interlayer temperature as the control input and the reaction temperature as the control output, using T_j and T_r , respectively. The secondary circuit takes the cooling water flow rate as the control input and the interlayer temperature as the control output, using Q_c and T_j , respectively. The value of the secondary circuit is set up by the main circuit and the secondary circuit, is generally controlled with a simple PID. The external structure diagram of the reactor is shown in Figure 1.

2.2. Description of the Hammerstein-Wiener System. The Hammerstein-Wiener model consists of two static, nonlinear modules and one dynamic linear module. Its model structure is shown in Figure 2.

Consider the following discrete-time multivariable Hammerstein-Wiener absolute system:

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{Ax}(t) + \mathbf{Bv}(t), & \mathbf{v}(t) &= \mathbf{g}(\mathbf{u}(t), t), \\ \mathbf{w}(t) &= \mathbf{Cx}(t), & \mathbf{y}(t) &= \mathbf{h}(\mathbf{w}(t), t), \end{aligned} \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{v} \in \mathbb{R}^r$, and $\mathbf{u} \in \mathbb{R}^m$ are the state variable, intermediate variable, and the input variable of the system, respectively; \mathbf{w} is the output variable in the linear part; $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ is the constant coefficient matrix with the corresponding dimension; \mathbf{g}, \mathbf{h} are the continuously differentiable nonlinear function vectors, indicating the static nonlinear link between the system input and the intermediate variable and between the linear part and the intermediate variable and meeting $\mathbf{g}(0, t) = 0, \mathbf{h}(0, t) = 0$.

It is assumed that the system controllability, state observability, and the origin as the equilibrium point of the system, the parameters of the system states, intermediate variables, and the input constraints are defined as follows:

$$\begin{aligned} \mathbf{x}(t) &\in M_x \subseteq \mathbb{R}^n, & \mathbf{v}(t) &\in M_v \subseteq \mathbb{R}^r, \\ \mathbf{u}(t) &\in M_u \subseteq \mathbb{R}^m, & \mathbf{w}(t) &\in M_w \subseteq \mathbb{R}^r, \end{aligned} \quad (2)$$

where M_x, M_v, M_u , and M_w represent the compact convex set, including the origin as the interior point.

3. The Constrained Predictive Controller Design Based on the Kalman Filter

3.1. The Constrained NMPC Algorithm [5]. The Hammerstein-Wiener model can be divided into the Hammerstein model [7] and the Wiener model [8]. Different approaches are adopted for these two models to design the controller of the system and to deal with the impact of system output and interference signal on the system. This section mainly describes the designs of the output feedback optimal control law of the Hammerstein linear subsystem based on the state observer.

The considered output signal $y(t)$ is assumed to have no direct control relationship with the input signal $u(t)$ in formula (1). Therefore, in its corresponding linear subsystem (A, B, C) , $\mathbf{x}(t+1) = \mathbf{Ax}(t) + \mathbf{Bv}(t)$, $\mathbf{w}(t) = \mathbf{Cx}(t)$; if the observer status is $\hat{\mathbf{x}}(t)$, the optimal control law of the linear subsystem can be described as $\mathbf{v}(t) \approx \mathbf{K}\hat{\mathbf{x}}(t)$, where $\mathbf{K} = -(\mathbf{R} + \mathbf{B}^T \mathbf{P} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P} \mathbf{A}$, $\mathbf{R} = r\mathbf{I}$ is the positive definite weighting matrix of the intermediate mass, \mathbf{P} is a symmetric positive definite solution to the Riccati equation $\mathbf{A}^T \mathbf{P} \mathbf{A} - \mathbf{P} + \mathbf{A}^T \mathbf{P} \mathbf{B} (\mathbf{R} + \mathbf{B}^T \mathbf{P} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P} \mathbf{A} + \mathbf{Q} = 0$, and $\mathbf{Q} = q\mathbf{I}$ is the semipositive definite state weighting matrix.

The actual control variable $\mathbf{u}(t)$ can be calculated by solving the nonlinear equations $\mathbf{g}(\mathbf{u}, t) - \mathbf{v}(t) = 0$, inevitably introducing the calculation error, and thereby affecting the closed-loop stability and optimality. Next, the actual control is determined through the prediction and the scroll optimization of the calculation error.

Considering the system in formula (1), the calculation error objective function is defined as follows:

$$J_1(t) = \sum_{i=t}^{t+T_p-1} \{ [\mathbf{g}(i) - \mathbf{v}(i)]^T [\mathbf{g}(i) - \mathbf{v}(i)] + \mathbf{u}(i)^T \mathbf{W} \mathbf{u}(i) \}, \quad (3)$$

where the integer $T_p > 0$ predicts the step length and the positive definite matrix \mathbf{W} is the input weighting matrix.

Thus, the NMPC issue of the constrained Hammerstein system is defined as follows:

$$\begin{aligned} \mathbf{u}(t, \hat{\mathbf{x}}(t), T_p)^* &= \min_u J_1, \\ \text{s.t. } \mathbf{x}(i+1) &= \mathbf{Ax}(i) + \mathbf{Bg}(\mathbf{u}(i), i), \\ \mathbf{v}(i) &= \mathbf{K}\hat{\mathbf{x}}(t), \\ \mathbf{x}(t) &\in M_x, & \mathbf{v}(t) &\in M_v, \\ \mathbf{u}(t) &\in M_u, & \mathbf{w}(t) &\in M_w, \\ \mathbf{x}(t) &= \hat{\mathbf{x}}(t), & (i &= t, t+1, \dots, t+T_p-1). \end{aligned} \quad (4)$$

The sequential quadratic programming is applied to calculate online the optimization problem in formula (4) and obtain the optimal control solution:

$$\begin{aligned} \mathbf{u}(t, \hat{\mathbf{x}}(t), T_p)^* \\ = \left\{ \mathbf{u}(t, \hat{\mathbf{x}}(t))^*, \mathbf{u}(t+1, \hat{\mathbf{x}}(t))^*, \dots, \mathbf{u}(t+T_p-1, \hat{\mathbf{x}}(t))^* \right\}. \end{aligned} \quad (5)$$

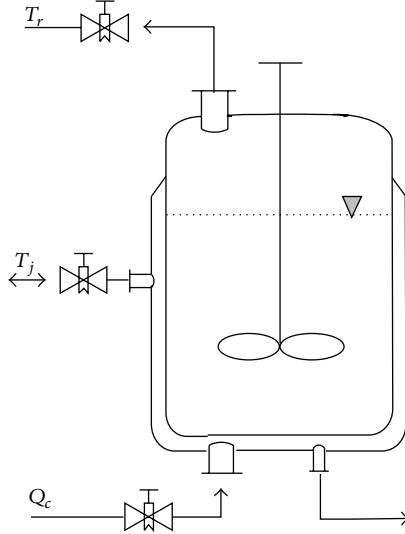


FIGURE 1: CSTR.

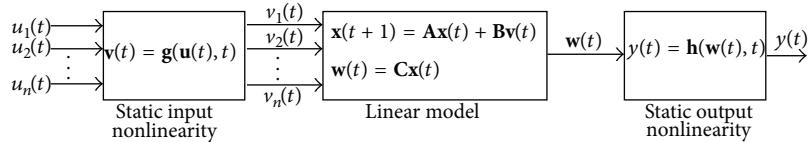


FIGURE 2: Hammerstein-Wiener model structure.

According to the principle of rolling optimization, the NMPC law based on the observation state is defined as follows:

$$\mathbf{u}(t)^{\text{NMPC}} = \mathbf{u}(t, \hat{\mathbf{x}}(t))^*, \quad (t = 0, 1, \dots). \quad (6)$$

In formula (6), the observation state $\hat{\mathbf{x}}(t)$ can be obtained through the following all-dimensional observer:

$$\begin{aligned} \hat{\mathbf{x}}(t+1) &= (\mathbf{A} - \mathbf{L}\mathbf{C})\hat{\mathbf{x}}(t) + \mathbf{B}\mathbf{g}(\mathbf{u}(t)^{\text{NMPC}}, t) \\ &\quad + \mathbf{L}\mathbf{w}(t), \quad (t = 0, 1, \dots), \end{aligned} \quad (7)$$

where \mathbf{L} is the gain matrix of the observer.

Control law (6) observes the system status with the state observer (7) at each sampling time. It then updates the initial conditions of the optimal control problem (4) on this basis and scrolls the optimization to calculate the predictive control variables of the current.

The observation error of the defined state is $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$; thus, the NMPC closed-loop Hammerstein system is

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{g}(\mathbf{u}(t)^{\text{NMPC}}, t) \\ &= (\mathbf{A} + \mathbf{B}\mathbf{K})\mathbf{x}(t) - \mathbf{B}\mathbf{K}\mathbf{e}(t) + \mathbf{B}\mathbf{v}_e(t), \end{aligned} \quad (8)$$

$$\mathbf{e}(t+1) = (\mathbf{A} - \mathbf{L}\mathbf{C})\mathbf{e}(t), \quad (t = 0, 1, 2, \dots),$$

where $\mathbf{v}_e(t) = \mathbf{g}(\mathbf{u}^{\text{NMPC}}, t) - \mathbf{v}(t)$ is the calculation error of the control variables; the controller gain \mathbf{K} is given by $\mathbf{v}(t) \approx \mathbf{K}\hat{\mathbf{x}}(t)$.

The NMPC closed-loop structure of the whole system is shown in Figure 3.

3.2. The Predictive Controller Design Based on the Kalman Filter. This section mainly introduces the interference model [6] based on the Kalman filter and evaluates the size of the interference signal as well as its impact on the system output based on the Wiener model. Meanwhile, with the above constrained NMPC output, the relationship between their outputs solves the error through the predictive nonlinear algebraic equation, and roll optimizes the performance indicators in the finite horizon to calculate the actual control variables.

The input interference model is the key to achieving unbiased control and to improving the tracking performance. To improve the tracking performance and to realize the unbiased control, the system is characterized by introducing the interference model. The model is built in the input interference along with the influence of heat produced by the materials in the chemical reactor on the output. Davison and Smith [8] regarded this method as the standard in eliminating the steady-state error in the design of the quadratic regulator:

$$\mathbf{T}_r(k+1) = \alpha\mathbf{T}_r(k) + \beta[\mathbf{T}_j(k) + \mathbf{d}(k)]. \quad (9)$$

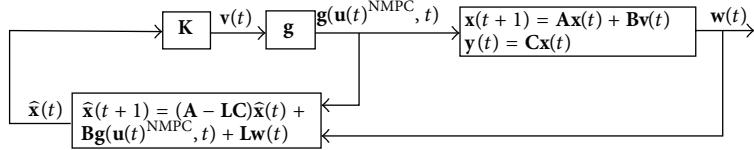


FIGURE 3: NMPC closed-loop system structure of the constrained hammerstein system.

After the establishment of the input interference model, the input interference \mathbf{d} is evaluated. If the interference is the state vector, the following augmented-state space model is obtained:

$$\begin{bmatrix} \mathbf{T}_r(k+1) \\ \mathbf{d}(k+1) \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{T}_r(k) \\ \mathbf{d}(k) \end{bmatrix} + \begin{bmatrix} \eta(k) \\ \xi(k) \end{bmatrix} [\beta \ 0]^T \mathbf{T}_j(k),$$

$$y(k) = [1 \ 0] [\mathbf{T}_r(k) \ \mathbf{d}(k)]^T, \quad (10)$$

where η and ξ are the process noises when the mean value is zero.

According to the measured value of the current output, the system state \mathbf{T} and the interference \mathbf{d} of the current can be determined using the following formula:

$$\begin{bmatrix} T_{\hat{k}|k} \\ d_{\hat{k}|k} \end{bmatrix} = \begin{bmatrix} T_{\hat{k}|k-1} \\ d_{\hat{k}|k-1} \end{bmatrix} + \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} (y_k - T_{\hat{k}|k-1}), \quad (11)$$

where L_1 and L_2 represent the corresponding Kalman gains of the state variable \mathbf{T} and the interference \mathbf{d} , respectively.

Then, the predictive value of the augmented state at the next moment is

$$\begin{bmatrix} T_{\hat{k}+1|k} \\ d_{\hat{k}+1|k} \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} T_{\hat{k}|k} \\ d_{\hat{k}|k} \end{bmatrix} + \begin{bmatrix} \beta \\ 0 \end{bmatrix} T_j(k). \quad (12)$$

For the sake of discussion, the following are set: $\mathbf{A}_1 = \begin{bmatrix} \alpha & \beta \\ 0 & 1 \end{bmatrix}$, $\mathbf{B}_2 = \begin{bmatrix} \beta \\ 0 \end{bmatrix}$, $\mathbf{H} = [1 \ 0]$, and $\mathbf{L} = [\mathbf{L}_1 \ \mathbf{L}_2]$. The Kalman gains of each cycle can then be calculated by $L_k = \mathbf{P}_k \mathbf{H}^T (\mathbf{H} \mathbf{P}_k \mathbf{H}^T + \mathbf{R})^{-1}$, where $\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_{k-1} \mathbf{H}) \mathbf{A} \mathbf{P}_{k-1} \mathbf{A}^T + \mathbf{Q}$ is the covariance of the estimation error of the current; \mathbf{Q}, \mathbf{R} are the process incentive noise covariance matrix and the observation noise covariance matrix, respectively, and their values directly affect the performance of the Kalman filter. Greg Welch and Gary Bishop (2006) discussed it in detail in their writings. Interference d has no direct relationship with input T_j , but the observability of interference d ensures passing the Kalman filter and then evaluates d . When designing the controller, the evaluation value of d is used to remove its influence on the system output. After evaluating the size of d using the Kalman filter in each sampling period, feedforward compensation is made on this part, and its influence on the system output is included in the prediction model. The real-time control variable is

$$\begin{aligned} \Delta T_j(k) &= \mathbf{D}^T [T_{r-sp}(k) - b\Delta d(k) - \tilde{T}_{op}(k)] \\ &= \mathbf{D}^T [T'_{r-sp}(k) - \tilde{T}_{op}(k)], \end{aligned} \quad (13)$$

where $\Delta d(k) = d(k) - d(k-1)$; $\mathbf{D} = (\mathbf{A}^T \mathbf{Q} \mathbf{A} + \mathbf{R})^{-1} \mathbf{A}^T \mathbf{Q}$; and $T'_{r-sp}(k)$ deducts the influence of interference d on the output from the expected value to form the new one.

The predictive control algorithm flowchart of the constrained Hammerstein-Wiener model with the feedforward compensation is shown in Figure 4.

4. Simulation Analysis

The experimental platform is the CSTR tester that can simulate in real time the entire process of the batch reaction. The main parameters selected include the volume of the reactants: 200 L, the spin speed of the stirrer: 90 r/min, the power of the stirrer: 1.5 Kw, the feeding of the reactant m_1 : 50 kg, and the specific heat: 1.2 kJ/kgK. The valve operation of the heating steam comprises manual and automatic modes. The initial stage of the reaction mainly involves the evoked response within the reactor. In other words, the temperature of the reactor is improved to 60°C with steam. The steam valve is manually operated in this process. The temperature in the reactor continues to rise because of the heat production in the reaction. The steam valve is switched to automatic mode to implement the control strategies. The process curve of the reaction temperature is set as follows: rising at 0.2°C/s, remaining constant at 120°C, and the overshoot of the temperature being less than 1.5°C. The result is compared with that of the traditional PID control, as shown in Figure 5.

Compared with the PID control, the PID control method cannot make the reactor temperature follow the process curve well because of the inherent nonlinearity and time lag characteristics of the object. However, the method in this paper achieves a good control effect.

In practical production, different products are produced through different processes, requiring corresponding control tasks to be changed, that is, selecting different process curves while keeping the algorithm unchanged. In the same simulation platform, the process curve is set to rise at 0.3°C/s and be constant at 120°C, with other parameters unchanged. The control result is shown in Figure 6(a), whereas the traditional PID algorithm control result is shown in Figure 6(b).

For batch production, the different natures of the reactants result in different exothermic characteristics in the entire reaction process, requiring the adopted control algorithm to have good robustness. In other words, the original control result remains even when the material property changes. Now, the reactant m_2 is used with the specific heat of 1.5 kJ/kgK. The reaction temperature process curve is set as follows: rising at 0.2°C/s and remaining constant at 120°C. The result is compared with that of the traditional PID control, as shown in Figure 7.

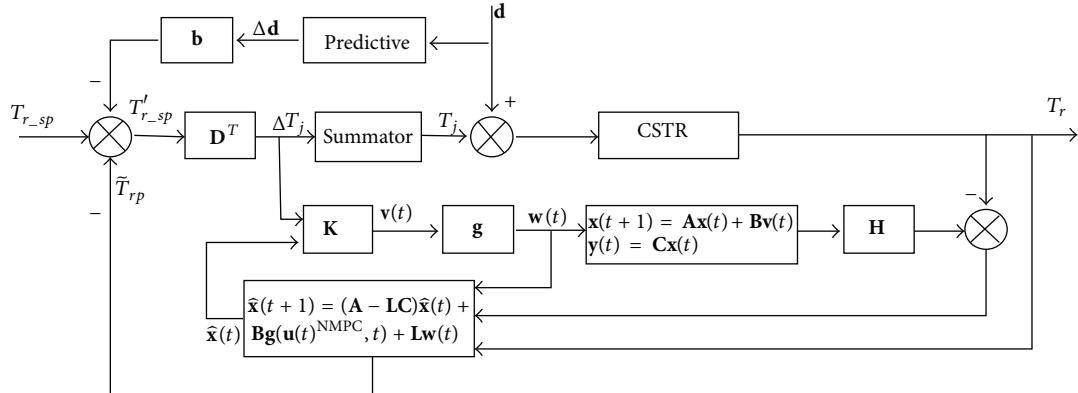


FIGURE 4: The predictive control algorithm of the constrained Hammerstein-Wiener model with the feedforward compensation.

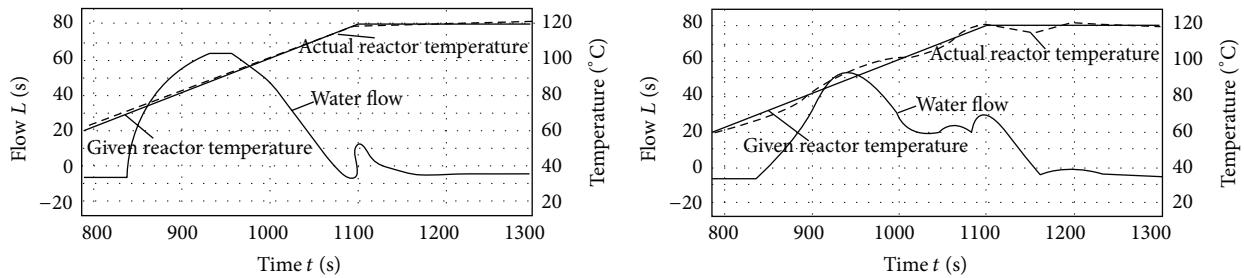


FIGURE 5: The process temperature curve rising at 0.2°C/s and remaining constant at 120°C.

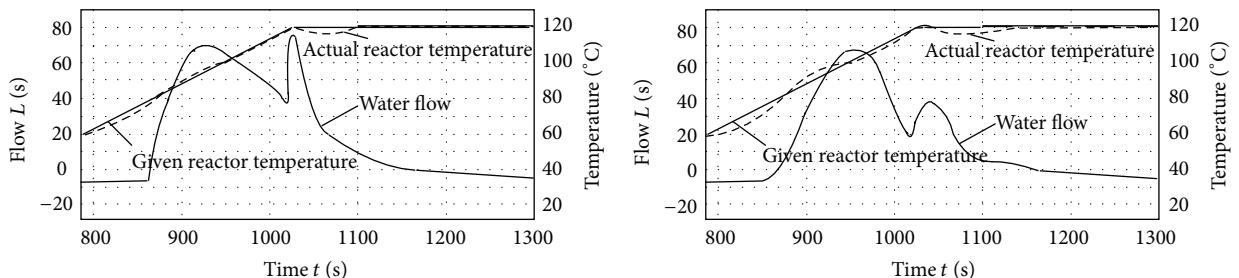


FIGURE 6: The process temperature curve rising at 0.3°C/s and remaining constant at 120°C.

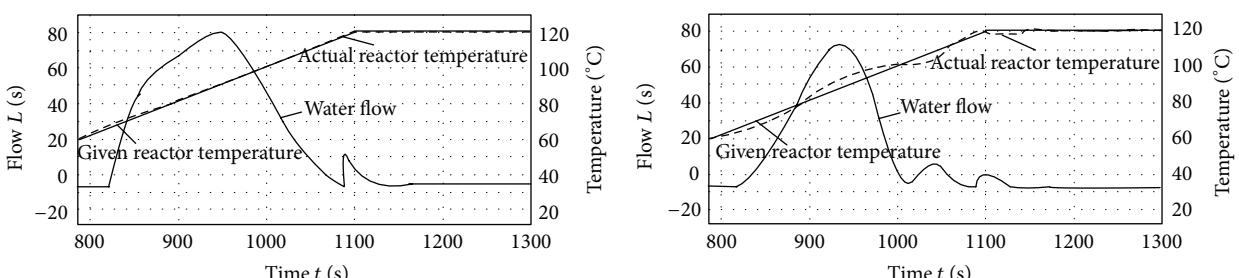


FIGURE 7: Comparison of the result after changing the special heat of the material.

The traditional PID control algorithm is difficult to use in controlling the reaction temperature in a satisfactory range because of the specific heat change of the reactant. However, the method proposed in this paper shows good robustness, indicating practical significance to the complex and changeable strong, nonlinear, chemical process system.

5. Conclusions

With the temperature tracking issue in the batch reaction process as the study object, as well as with the condition that the value of the heat production cannot be measured and its influence law on the output is unknown in the whole process of the reactant reaction, this paper proposes the predictive control algorithm of the batch reaction based on the Kalman filter for the constrained Hammerstein-Wiener system. Combined with state observer, the paper designs the output feedback optimal control law of the linear subsystem and characterizes the unmodeled dynamic system with the output interference model, which compensates for the impact of the heat production from the reaction on the system output and improves the accuracy of temperature tracking. The simulation experiment is conducted to verify the control results of this algorithm on different process curves and determine its robustness. The proposed algorithm, when compared with the traditional PID algorithm, fully demonstrates effectiveness and feasibility.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (no. 61074020). The authors also wish to thank the associate editor and the anonymous reviewers for their valuable and constructive criticisms and suggestions.

References

- [1] A. Jutan and A. Uppal, "Combined feedforward-feedback servo control scheme for an exothermic batch reactor," *Industrial & Engineering Chemistry Process Design and Development*, vol. 23, no. 3, pp. 597–602, 1984.
- [2] H. Song, W. Gui, and C. Yang, "Identification of Hammerstein-Wiener model with least squares support vector machine," in *Proceedings of the 26th Chinese Control Conference (CCC '07)*, pp. 260–264, July 2007.
- [3] M. Hong and S. Cheng, "Nonlinear model predictive control based on LS-SVM Hammerstein-wiener model," *Journal of Computational Information Systems*, vol. 8, no. 4, pp. 1373–1381, 2012.
- [4] J. C. Gómez, A. Jutan, and E. Baeyens, "Wiener model identification and predictive control of a pH neutralisation process," *IEE Proceedings*, vol. 151, no. 3, pp. 329–338, 2004.
- [5] H. E. De-feng, Y. U. Li, and Z. O. U. Tao, "Output feedback nonlinear predictive control for constrained Hammerstein systems," *Control Engineering of China*, vol. 16, no. 4, pp. 416–418, 2009.
- [6] X. F. Zhu and D. E. Seborg, "Nonlinear predictive control based on Hammerstein models," *Control Theory & Applications*, vol. 11, no. 5, pp. 564–575, 1994.
- [7] A. L. Cervantes, O. E. Agamennoni, and J. L. Figueroa, "A nonlinear model predictive control system based on Wiener piecewise linear models," *Journal of Process Control*, vol. 13, no. 7, pp. 655–666, 2003.
- [8] E. J. Davison and H. W. Smith, "Pole assignment in linear time-invariant multivariable systems with constant disturbances," *Automatica*, vol. 7, pp. 489–498, 1971.

Research Article

Cross-Domain Personalized Learning Resources Recommendation Method

Long Wang,¹ Zhiyong Zeng,² Ruizhi Li,² and Hua Pang³

¹ College of Information, Liaoning University, Shenyang 110036, China

² School of Computer Science and Information Technology, Northeast Normal University, Changchun 130024, China

³ College of Education Technology, Shenyang Normal University, Shenyang 110034, China

Correspondence should be addressed to Hua Pang; 603665868@qq.com

Received 26 June 2013; Accepted 3 September 2013

Academic Editor: William Guo

Copyright © 2013 Long 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.

According to cross-domain personalized learning resources recommendation, a new personalized learning resources recommendation method is presented in this paper. Firstly, the cross-domain learning resources recommendation model is given. Then, a method of personalized information extraction from web logs is designed by making use of mixed interest measure which is presented in this paper. Finally, a learning resources recommendation algorithm based on transfer learning technology is presented. A time function and the weight constraint of wrong classified samples can be added to the classic TrAdaBoost algorithm. Through the time function, the importance of samples date can be distinguished. The weight constraint can be used to avoid the samples having too big or too small weight. So the Accuracy and the efficiency of algorithm are improved. Experiments on the real world dataset show that the proposed method could improve the quality and efficiency of learning resources recommendation services effectively.

1. Introduction

With the continuous development of network technology and educational informationization, the network learning system is widely applied in every stage of education. How to improve the intelligence of the network learning system and the users' efficiency has become a common concern for all researchers. It is more and more difficult for students to look for their interested resources from the increasing learning resources in the network learning system. So adding learning resources recommendation service to the network learning system can save the students' time and effort on information searching and can free the students from enormous network information resources. Under these conditions and needs, the personalized learning resources recommendation service technique has been gradually developed.

The personalized learning resources recommendation service refers to predicting whether the new learning resources can meet the students' personalized demand which is extracted from the previous learning information and recommending the interested resources to the students.

In order to obtain good recommendation results, a certain amounts of personalized information should be accumulated. The accumulation time is proportional to the student's learning time in the network learning system. Shortening the accumulation time is an effective method to improve learning resources recommendation service efficiency. The cross-domain learning resources recommendation is an effective method to shorten the process of accumulating personalized information.

In this paper, a cross-domain personalized learning resources recommendation method is presented. First, a cross-domain personalized learning resources recommendation service model is introduced. Next, a personalized information mining method is designed. Then, a cross-domain learning resources recommendation algorithm is presented. Finally, the simulation results and the experiment analysis are given.

2. Service Model

There are large amount of learning resources in network learning system. Generally, the student does not know if

it contains the interested contents when he studies new resource. So, he can only check the content or the resource description. If it does not contain the interested content, the checking is useless. In order to reduce the useless checking, the recommendation service should be added to the system. For improving the service efficiency, a cross-domain personalized recommendation service model is presented in this paper. The service model is shown in Figure 1.

As shown in Figure 1, when students use the network learning system, some web logs data are generated. These data include user name, access date, access times, access time, and requested URL. Through mining the data, the personalized information of students are obtained. Put the personalized information as a basis for personalized learning resources recommendation into the student personalized information database. When a student learns some subject, the service can extract learning information from current subject and the subjects learned before based on his personalized information, and train the recommendation model. Through the recommendation model, the service can find the learning resources which the student is interested in the current subject and recommend them to the student.

3. Personalized Information Mining Based on Mixed Interest Measure

For a student, what learning resources they are interested in is the most easy and effective way to reflect his personalized demands. From the web logs, the service can find out the learning resources that the student has browsed, analyze what they are interested in, what they are uninterested in, and then compose the personalized information of the student.

Now, browsing interest measure [1] is widely used to estimate whether the users are interested in the browsed contents or not. The browsing interest measure P is computed as:

$$P = \frac{C \times T}{B}, \quad (1)$$

where C is the browsing times, T is the each browsing time, and B is the number of bytes of the browsing content.

In network learning system, the file types of learning resources are not only text but also image, animation, audio, and video. In formula (1), the difference from different file types is not considered. In addition, the learning resources not only include the resource content but also include the resource description which is called the brief of resource. According to the characteristics of learning resources, the mixed interest measure I is computed as follow.

First, the content browsing interest measure P^c is computed as the following:

$$P^c = \eta \times \alpha \times \sum_{i=1}^n \frac{T_i^c}{B^c}, \quad (2)$$

where n is the browsing times, T^c is the each content browsing time, B^c is the number of bytes of the resource content, α is the coefficient of resource content, and η is the coefficient of resource type.

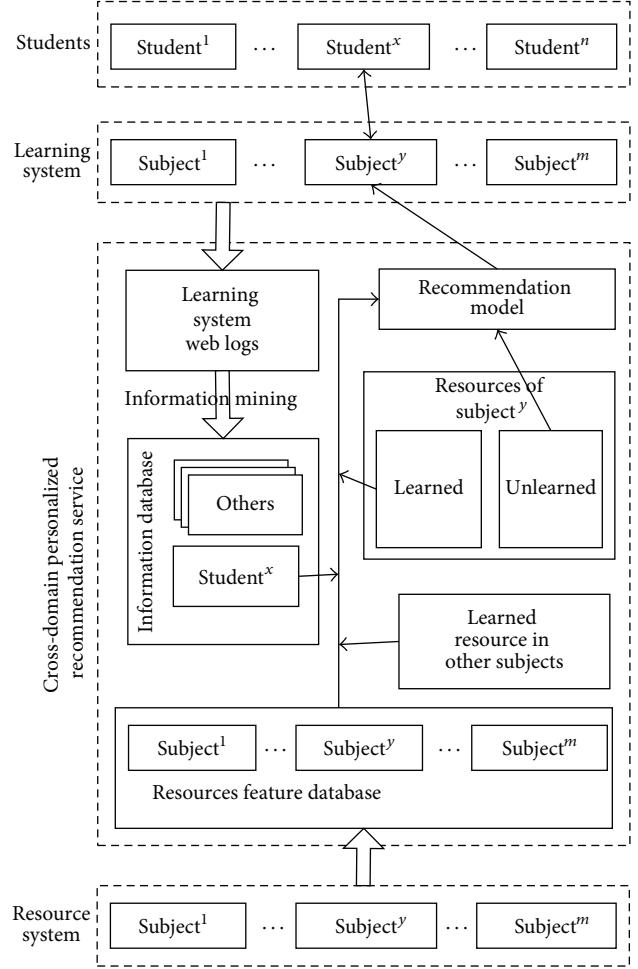


FIGURE 1: Service Model.

Then, the brief browsing interest measure P^b is computed as the following:

$$P^b = \beta \times \sum_{i=1}^m \frac{T_i^b}{B^b}, \quad (3)$$

where m is the browsing times, T^b is the each brief browsing time, B^b is the number of bytes of the resource brief, and β is the coefficient of resource brief.

Next, the browsing interest measure P is computed as the following:

$$P = P^c + P^b. \quad (4)$$

Finally, the mixed interest measure I is computed as the following:

$$I = \frac{P / \sum_{j=1}^N P_j}{1/N}, \quad (5)$$

where N is the total number of the browsed resources.

If $I \geq 1$, the learning resource is interested; otherwise, the learning resource is uninterested.

4. Cross-Domain Learning Resources Recommendation Algorithm

The basic theory of resources recommendation is analyzing new learning resources for students based on their personalized information and finding their interested resources. At present, traditional machine learning technology is used in most of resources recommendation service, and most of traditional machine learning is based on statistic learning, in which training data and testing data should be under the same distribution.

But in the cross-domain recommendation field, traditional machine learning cannot solve this problem, and the transfer learning is an effective method to solve it.

4.1. Transfer Learning. Now, transfer learning includes instance-based transfer learning and feature-based transfer learning [2–7]. Source domain data and target domain data may have different distribution, yet similar content, in which case, instance-based transfer learning has more obvious effect. The instance-based transfer learning refers to estimate instances from source domain data by some method and applies the instances evaluated better to the learning task of the target domain. Researchers have proposed many instance-based transfer learning algorithms [8–12]. Dai and others [13] have expanded traditional Boosting ensemble learning algorithm and proposed TrAdaBoost algorithm that has ability to transfer. The basic idea of TrAdaBoost is that the source domain data and the target domain data are mixed, and then they are trained. And after each iteration training, adapt the weight of the instance which is classified to be wrong. If the instance belongs to the target domain, the weight of it which is considered important should be increased and the effect of the sample should be improved at the next iteration training. If the instance belongs to the source domain, the weight of it which is considered unimportant should be decrease and the effect of sample should be reduced at next iteration training.

4.2. Cross-Domain Resources Recommendation Algorithm Design. In TrAdaBoost, the instances are treated in the same way during the initial weight assignment stage, and the weights of source domain instances and the weights of target domain instances are assigned averagely. For learning resources recommendation, the learning resources that the students browsed recently are more likely to reflect the current personalized information, so a time function can be added to weight assignment stage of the algorithm. In this way, the importance of learning resources can be distinguished.

In addition, the instance's weight which is classified wrong in target domain in TrAdaBoost algorithm will be increased continuously. On the contrary, the instance's weight which is classified wrong in source domain will be decreased continuously. When the number of instances which have too big or too small weight reaches a certain degree, it will decline the classifier ability of the algorithm. In order to solve this

problem, the weight of the wrong classified samples will be constrained.

The detailed description of the cross-domain personalized resource recommendation algorithm is shown below.

Step 1. Let $D = D_s \cup D_o$, where $D_o = \{(d_{m+1}, C(d_{m+1})), (d_{m+2}, C(d_{m+2})), \dots, (d_n, C(d_n))\}$ is the learned resources set in the current subject, $D_s = \{(d_1, C(d_1)), (d_2, C(d_2)), \dots, (d_m, C(d_m))\}$ is the learned resources set in other subjects, and $C(d_i) \in \{0, 1\}$ is the category label of d_i .

Step 2. Set R ($R < m$) and initialize $r = 0$; R is the times of iteration, and r is the count of iteration.

Step 3. Initialize the weight of instances:

$$w_i^0 = \begin{cases} \left(\frac{1}{m}\right) \times f(t_i), & 1 \leq i \leq m, \\ \left(\frac{1}{n} - \frac{1}{m}\right) \times f(t_i), & m + 1 \leq i \leq n. \end{cases} \quad (6)$$

The time function $f(t_i)$ is as the following:

$$f(t_i) = \begin{cases} 1, & t_i = 0 \\ 1 - \lambda \times \ln\left(1 + \frac{1}{t_i}\right)^{t_i}, & t_i > 0. \end{cases} \quad (7)$$

Step 4. $r = r + 1$.

Step 5. The weight of sample should be normalized, and it should be adjusted as the following:

$$w_i^r = \frac{w_i^{r-1}}{\sum_{i=1}^n w_i^{r-1}},$$

$$w_i^r = \begin{cases} u, & w_i^r > u \\ l, & w_i^r < l \\ w_i^r, & \text{others.} \end{cases} \quad (8)$$

Step 6. Call weak learner; then get back a hypothesis $h^r(d) : d \rightarrow C(d)$.

Step 7. Calculate the error of $h^r(d)$:

$$\varepsilon^r = \sum_{i=m+1}^n \frac{w_i^r \times |h^r(d_i) - c(d_i)|}{\sum_{i=m+1}^n w_i^r}. \quad (9)$$

Let

$$\varepsilon^r = \begin{cases} \varepsilon^r & \varepsilon^r < \frac{1}{2} \\ \frac{1}{2} & \varepsilon^r \geq \frac{1}{2} \end{cases} \quad (10)$$

Step 8. Update the weights of the instances:

$$w_i^r = w_i^r$$

$$\times \begin{cases} 1, & h^r(d_i) = c(d_i) \text{ or } w_i^r \in \{u, l\} \\ \beta_s, & h^r(d_i) \neq c(d_i), \quad 1 \leq i \leq m, w_i^r > l \\ (\beta_o^r)^{-1}, & h^r(d_i) \neq c(d_i), \quad m + 1 \leq i \leq n, w_i^r < u, \end{cases} \quad (11)$$

TABLE 1: Experimental data.

Student (data set)	Literature		History	
	Interested	Uninterested	Interested	Uninterested
$S_0 (U_0)$	812	614	789	543
$S_1 (U_1)$	785	586	856	654
$S_2 (U_2)$	914	689	758	542
$S_3 (U_3)$	1021	754	699	488
$S_4 (U_4)$	654	562	951	623
\vdots	\vdots	\vdots	\vdots	\vdots
$S_{59} (U_{59})$	948	643	751	523
$S_{60} (U_{60})$	987	763	928	616
$S_{61} (U_{61})$	1012	670	789	563
$S_{62} (U_{62})$	965	543	851	652

where

$$\beta_s = \frac{1}{1 + \sqrt{2 \times \ln(m/R)}}, \quad (12)$$

$$\beta_o^r = \frac{\varepsilon^r}{1 - \varepsilon^r}.$$

Step 9. If $r < R$, go to Step 4; otherwise, get back the final hypothesis:

$$h^R(d) = \begin{cases} 1, & \sum_{r=R/2}^R \ln\left(\frac{1}{\beta_o^r}\right) \times h^r(d) \geq \frac{1}{2} \times \sum_{r=R/2}^R \ln\left(\frac{1}{\beta_o^r}\right). \\ 0, & \text{others.} \end{cases} \quad (13)$$

In this algorithm, the formula (7) has the following properties: the function $f(t_i)$ on $[0, +\infty)$ is monotonically decreasing; $f(0) = 1$; $\lim_{t_i \rightarrow +\infty} f(t_i) = 0$; the decrease of function $f(t_i)$ is from fast to slow. Its properties satisfy the time characteristics of the learning resources. When $1 \leq i \leq m$, t_i is the time interval between the browsed time of the current instance and the recent browsed time of the instances in D_s . when $m + 1 \leq i \leq n$, t_i is the time interval between the browsed time of the current instance and the recent browsed time of the instances in D_o . $\lambda \in [0, 1]$ reflects the descending rate of the instance's importance. When $\lambda = 0$, the time characteristics of instances are not considered. When $0 < \lambda \leq 1$, the bigger the λ , the faster the descending rate of the instance's importance, and the value of $f(t_i)/f(0)$ is bigger than $1 - \lambda$.

In formula (11), u is the upper threshold of the weight, l is the limit threshold of the weight, $u = \max(w_i^1) \times \theta$, and $l = \min(w_i^1)/\theta$. The $\max(w_i^1)$ is the maximum value of w_i^1 , $\min(w_i^1)$ is the minimum value of w_i^1 , $\theta \geq 1$ is a real and can be adjusted. When θ is large enough, it means that the weights of the wrong classified instances are not constrained.

5. Experimental Evaluation

This paper chooses the Literature subject and the History subject to verify the effectiveness of the recommendation method and select the learning logs of a class with 62 students as the experimental data from a network learning system without recommendation service. The learning resources which are placed in "favorite" by students are used as the interested resources, and the learning resources which are placed in "recovery" by students are used as the uninterested resources. The experimental data sets are composed of the resources feature expression and their browsed time. Part of the information is shown in Table 1.

Firstly, test the effectiveness of the personalized information mining based on mixed interest measure; the specific experimental process of the personalized information mining for each student is shown as follows.

- (1) Extract the records of student S_i from web logs.
- (2) Extract the records which are corresponding to the resources in U_i .
- (3) According to the proposed method, calculate the student's interest measure of each learning resources in U_i .
- (4) Classify the learning resources according to whether the interest measure is bigger than 1, compare the results with the actual categorical attribute, and calculate the accuracy f of the method.

The experimental results are shown in Figure 2.

In 62 groups experimental data, 8 groups are between 84 and 85%, 25 groups are between 85 and 86%, 27 groups are between 86 and 87%, 2 groups are between 87 and 88%, and the average accuracy rate can reach 86.48%, and these meet the demands of personalized learning resources recommendation.

Then, apply the data set that was achieved from personalized information mining to test cross-domain resources recommendation algorithm, and the specific experimental process is as follows.

- (1) For each data set, randomly select 10% instances from History subject and all instances from Literature subject, use this data set as training set, and use the rest instances of History subject as test set.
- (2) The values of θ are 5 and 10^6 , the values of λ are 0.5 and 0.
- (3) Train and test, and the iteration times are 1 to 50. We should compute the accuracy, precision, and recall.
- (4) Repeat 10 times, calculate the average of accuracy, precision and recall.
- (5) Calculate the average of accuracy, precision, and recall of all data sets.

The final comparison results of the average of accuracy, precision, and recall are shown in Table 2.

Using the same data set, the experimental results of the traditional methods SVM and TrSVM [12] are as in Table 3.

TABLE 2: Final results comparison.

θ	Coefficient λ	Accuracy	Precision	Recall	Comment
10^6	0	80.98%	81.63%	81.17%	TrAdaBoost
10^6	0.5	82.11%	82.15%	82.06%	Add time function only
5	0	85.59%	86.10%	86.06%	Add weight constraints only
5	0.5	86.54%	86.63%	86.42%	In this paper

TABLE 3: Experimental results.

Accuracy	SVM	TrSVM			In this paper
	Precision	Recall	Accuracy	Precision	Recall
55.13%	55.89%	56.03%	82.3%	82.5%	82.4%
					86.54%
					86.63%
					86.42%

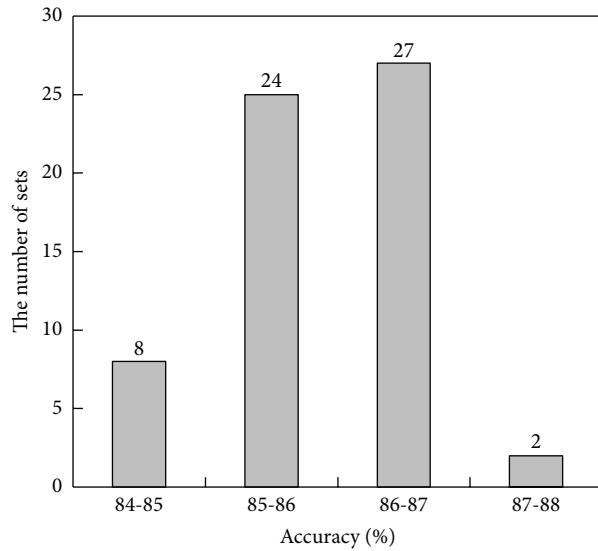


FIGURE 2: Personalized information mining experimental results.

Using the transfer learning recommendation method, the accuracy, precision and recall are all much higher than the traditional SVM method and higher than the TrSVM method.

It is shown in Table 2 that the accuracy, precision, and recall of TrAdaBoost algorithm are 80.98%, 81.63%, and 80.98%. After adding the time function to the weight assignments stage, the accuracy, precision and recall are 82.11%, 82.15% and 82.06%, just a slight improvement, this is because the algorithm uses Boosting integration learning framework, and so the initial weight assignment has a small effect on the result of final recommendation. When adding weight constraints only, the accuracy, precision, and recall are 85.59%, 86.10%, and 85.59%, better than TrAdaBoost algorithm; this is because there are error data in the data set obtained from personalized information mining. Without weight constraint, the error data will greatly affect the quality of recommendation service, and under the help of the weight constraint, the quality of recommend service is greatly improved. The accuracy, precision, and recall in this paper

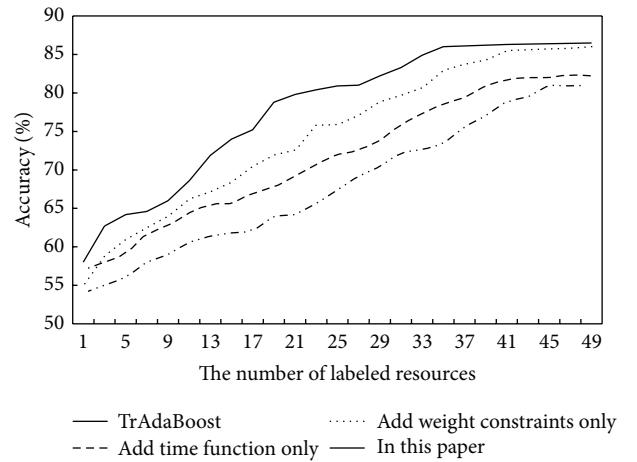


FIGURE 3: Convergence comparison.

are 86.54%, 86.63%, and 86.42%, apparently higher than the TrAdaBoost algorithm; the recommendation algorithm in this paper is more effective.

The convergence comparison is shown in Figure 3.

It is shown in Figure 3 that the TrAdaBoost algorithm needs about 45 times of iteration to achieve convergence. While adding time functions or weight constraints, the algorithm needs about 40 times to achieve convergence, and the proposed algorithm in this paper only needs about 35 times of iteration. The proposed algorithm is more effective.

The experimental results show that the recommendation service based on the proposed algorithm in this paper is superior to the recommendation service based on TrAdaBoost algorithm both in quality and efficiency and meets the demands of personalized learning resources recommendation service.

6. Conclusion

In the personalized learning resources recommendation service, the learning resources recommendation method requires a certain amount of personalized information to

achieve a satisfactory effect. How to achieve the information accurately and conveniently and make full use of the information to recommend is an urgent puzzle to be solved. We extract students' personalized information from web log based on the mix interest measure, put forward a kind of cross-domain learning resources recommendation method, and this method not only can take advantage of the students' personalized information in current subjects but also can use the personalized information in other subjects they have learned. The experimental results show that the proposed learning resources recommendation method can effectively improve the quality and efficiency of learning resources recommendation.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grants nos. 60973136 and 61073164) and the Youth Research Foundation of Liaoning University (Grant no. 2011LDQN14).

References

- [1] S. D. Pierrako and G. Paliouras, *Web Usage Mining as a Tool for Personalization: A Survey*, Kluwer Academic Publishers, 2003.
- [2] S. J. Pan, V. W. Zheng, Q. Yang, and D. H. Hu, "Transfer learning for wifi-based indoor localization," in *Proceedings of the Workshop on Transfer Learning for Complex Task of the 23rd AAAI Conference on Artificial Intelligence*, Chicago, Ill, USA, July 2008.
- [3] J. Ramon, K. Driessens, and T. Croonenborghs, "Transfer learning in reinforcement learning problems through partial policy recycling," in *Proceedings of the 18th European Conference on Machine Learning*, pp. 699–707, Springer, 2007.
- [4] M. E. Taylor and P. Stone, "Cross-domain transfer for reinforcement learning," in *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*, pp. 879–886, ACM, June 2007.
- [5] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*, pp. 759–766, Corvalis, Ore, USA, June 2007.
- [6] H. Daumé III and D. Marcu, "Domain adaptation for statistical classifiers," *Journal of Artificial Intelligence Research*, vol. 26, pp. 101–126, 2006.
- [7] Z. Wang, Y. Song, and C. Zhang, "Transferred dimensionality reduction," in *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD '08)*, pp. 550–565, Springer, September 2008.
- [8] J. Huang, A. Smola, A. Gretton et al., "Correcting sample selection bias by unlabeled data," in *Proceedings of the 19th Annual Conference on Neural Information Processing Systems*, pp. 601–608, MIT Press, 2007.
- [9] M. Sugiyama, S. Nakajima, H. Kashima, P. Von Bünnau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," in *Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS '07)*, pp. 1433–1440, MIT Press, December 2007.
- [10] S. Bickel, C. Sawade, and T. Scheffer, "Transfer learning by distribution matching for targeted advertising," in *Proceedings of the 21th Annual Conference on Neural Information Processing Systems*, pp. 145–152, MIT Press, 2009.
- [11] A. Storkey and M. Sugiyama, "Mixture regression for covariate shift," in *Proceedings of the 19th Annual Conference on Neural Information Processing Systems*, pp. 1337–1344, MIT Press, 2007.
- [12] J. Hong, J. Yin, Y. Huang, Y. Liu, and J. Wang, "TrSVM: a transfer learning algorithm using domain similarity," *Journal of Computer Research and Development*, vol. 48, no. 10, pp. 1823–1830, 2011.
- [13] W. Dai, Q. Yang, G. Xue et al., "Boosting for transfer learning," in *Proceedings of the 24th International Conference on Machine Learning*, pp. 193–200, ACM, 2007.

Research Article

The Prediction of Calpain Cleavage Sites with the mRMR and IFS Approaches

Wenyi Zhang, Xin Xu, Longjia Jia, Zhiqiang Ma, Na Luo, and Jianan Wang

School of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, China

Correspondence should be addressed to Na Luo; luon110@nenu.edu.cn and Jianan Wang; wangjn@nenu.edu.cn

Received 11 July 2013; Accepted 12 August 2013

Academic Editor: William Guo

Copyright © 2013 Wenyi Zhang 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.

Calpains are an important family of the Ca^{2+} -dependent cysteine proteases which catalyze the limited proteolysis of many specific substrates. Calpains play crucial roles in basic physiological and pathological processes, and identification of the calpain cleavage sites may facilitate the understanding of the molecular mechanisms and biological function. But traditional experiment approaches to predict the sites are accurate, and are always labor-intensive and time-consuming. Thus, it is common to see that computational methods receive increasing attention due to their convenience and fast speed in recent years. In this study, we develop a new predictor based on the support vector machine (SVM) with the maximum relevance minimum redundancy (mRMR) method followed by incremental feature selection (IFS). And we concern the feature of physicochemical/biochemical properties, sequence conservation, residual disorder, secondary structure, and solvent accessibility to represent the calpain cleavage sites. Experimental results show that the performance of our predictor is better than several other state-of-the-art predictors, whose average prediction accuracy is 79.49%, sensitivity is 62.31%, and specificity is 88.12%. Since user-friendly and publicly accessible web servers represent the future direction for developing practically more useful predictors, here we have provided a web-server for the method presented in this paper.

1. Introduction

Calpains are an important family of the Ca^{2+} -dependent cysteine proteases which catalyze the limited proteolysis of many specific substrates [1, 2]. Probably, 16 known calpain isoform genes are founded in humans. Then, 14 genes encoded proteins have cysteine protease domains, and the other 2 genes that encode some regulatory proteins are associated with some catalytic subunits forming heterodimeric proteases [3, 4]. Calpains play crucial roles in basic physiological and pathological processes, including the regulation of gene expression, signal transduction, cell death and apoptosis, remodeling cytoskeletal attachments during cell fusion or motility, and cell cycle progression [3–5]. Moreover, calpain aberrancies frequently lead to a variety of diseases and cancers [6]. As we know, traditional experimental identification and characterization of calpain cleavage sites are labor-intensive and expensive. Recently, calpain cleavage sites prediction attracts more and more attention, and more and more

studies have understood its regulatory roles and molecular mechanisms of calpain cleavage.

In recent years, many computational methods were developed to predict calpain cleavage sites. In the paper [7], Tompa et al. selected 49 calpain substrates with a total of 106 sequentially identified cleavage sites from the literature. They determined the amino acid preferences around the cleavage bond with 11-mer peptide, and they synthesized a short peptide of TPLKSPPPSPR to be a superior substrate of calpain. Then, Boyd et al. developed PoPS online tool to predict protease specificity [8, 9]. And the site prediction based on the frequency and substitution matrix scoring strategy predicted Calpain 1 and 2 specific cleavage sites [10]. Recently, Liu et al. developed a new computational program for the prediction of calpain cleavage sites. With the previously released algorithm of GPS (Group-based Prediction System), they designed a novel software package of GPS-CCD (Calpain Cleavage Detector) for prediction of calpain cleavage sites [6]. Although aforementioned predictors were effective, we

should make more efforts to improve the performance of calpain cleavage sites prediction.

In this study, we developed a new predictor based on the support vector machine (SVM) with the maximum relevance minimum redundancy (mRMR) method followed by incremental feature selection (IFS). And we concerned the features of physicochemical/biochemical properties, sequence conservation, residual disorder, secondary structure, and solvent accessibility to represent the calpain cleavage sites. Experimental results showed that the performance of our predictor was better than several other state-of-the-art predictors, whose average prediction accuracy was 79.49%, sensitivity was 62.31%, and specificity was 88.12%. Since user-friendly and publicly accessible web servers represented the future direction for developing practically more useful predictors [11], here we have provided a web server for the method presented in this paper at http://202.198.129.219:8080/calpain_cleavage/.

2. Materials and Method

2.1. Data Sets. Here, we selected 130 unique substrates for calpain cleavage sites. And all the proteins were extracted from Uniprot/Swiss-Prot (Jul 20, 2012), by searching the “calpain” in the field “Sequence annotation” with experimental verification. We defined a calpain cleavage peptide (m, n) as the cleavage bond flanked by m residues upstream and n residues downstream, where m and n were equal to 10. Similar to [6], all experimentally verified cleavage sites were regarded as positive samples, and the other noncleavage sites in the same substrates were taken as the negative samples. With the threshold of 40% identity by CD-HIT, the training dataset contained 368 positive samples.

2.2. Protein Features and Vector Encoding. The first feature we select is the position specific scoring matrix (PSSM) of each calpain cleavage peptide. All biological species have developed starting out from a very limited number of ancestral species. Their evolution involves changes of single residues, insertions and deletions of several residues [12], gene doubling, and gene fusion. With these changes accumulated for a long time, many similarities between initial and resultant amino acid sequences are gradually eliminated, but the corresponding proteins may still share some equal functions and action mechanisms. Accordingly, evolutionary conservation may play important roles in biological analysis [13]. We used PSI-BLAST [14] to generate the scoring of specific residues. PSSM profile for each peptide can be represented as a matrix of $M \times 20$ dimensions, and M is the length of peptide; 20 dimensions mean a measure of residue conservation of 20 different standard amino acids [15].

The second feature we exploit is the feature of amino acid factors derived from AAIndex [16], which is a famous database including various physicochemical and biochemical properties. Due to native 20 amino acids having their own specific properties, the composition of these properties of different residues will affect the structure and function of the protein [13]. Atchley et al. [17] performed multivariate

statistical analyses on AAIndex and replaced amino acid properties with five pattern scores (polarity, secondary structure, molecular volume, codon diversity, and electrostatic charge) [13]. We use the five pattern scores to represent each amino acid.

We also consider other features to make full use of protein sequence and prior knowledge, including disorder score, secondary structure, and solvent accessibility. Therefore, the information of disorder score is involved with protein structure and function. In this study, we use VSL2 [18] to calculate the disorder score of each amino acid peptide. Moreover, we use SSPro4 [19] to predict the secondary structural property of each amino acid of a given protein sequence as “helix,” “stand,” or “other” which are encoded with “100,” “010,” and “001”. So, we construct a series of $K \times 3$ matrix, K is the length of the chain peptide. The predictor SSPro4 also can predict solvent accessibility of each amino acid as “buried” or “exposed,” which is encoded with “10” and “01;” then, $K \times 2$ matrix is formed; K is also the length of chain peptide.

2.3. The Feature Space. As mentioned previously, for each amino acid of a given peptide, the following 31 features are needed: 20 PSSM conservation score features, 5 amino acid factors features, 1 disorder feature, 3 secondary structure features, and 2 solvent accessibility features. The length of given peptide is 21; there are total of $31 \times 21 = 651$ features. According to (6) of [20], the feature vector for any protein, peptide, or biological sequence is none but a general form of pseudo amino acid composition or PseAAC [21, 22] that can be formulated as

$$P = [\psi_\psi \psi_2 \cdots \psi_\mu \cdots \psi_\Omega]^T, \quad (1)$$

where $\Omega = 651$, T is the transpose operator, and ψ_μ ($\mu = 1, 2, \dots, 651$) represents the μ feature.

2.4. The mRMR Method. We use the mRMR method to rank the importance of the 651 features based on minimal redundancy and maximal relevance [23]. The ranked feature with the smaller index indicates that it has a better trade-off between the maximum relevance and minimum redundancy. The mutual information is used for reflecting the dependence of vector x and vector y :

$$I(x, y) = \iint p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dx dy, \quad (2)$$

where x and y are two random vectors, $p(x, y)$ is the joint probabilistic density, and $p(x)$ and $p(y)$ are the marginal probabilistic densities.

Suppose that the set M is the already-selected feature set containing m features, and the set N is the to-be-selected feature set containing n features. D denotes the relevance between the feature f in N and the class c :

$$D = I(f, c). \quad (3)$$

And R denotes the relevance between the feature f in N and all features in M , and R can be calculated by

$$R = \frac{1}{m} \sum_{f_i \in M} I(f, f_i). \quad (4)$$

So the feature f_j in the set N with the maximum relevance and minimum redundancy can be calculated by

$$\max_{f_j \in N} \left[I(f_j, c) - \frac{1}{m} \sum_{f_i \in M} I(f_j, f_i) \right], \quad (j = 1, 2, \dots, n). \quad (5)$$

We can use the mRMR method to find the feature set S , and each feature in S has the index indicating its importance; the more important the feature is, the smaller the index is.

2.5. Support Vector Machine. SVM belongs to the family of margin-based classifier and is very powerful to deal with prediction, classification, regression problems [24, 25]. Therefore, SVM is widely used for all kinds of problems. SVM looks for optimal hyperplane which maximizes the distance between the hyperplane and the nearest sample from each of the two classes. Formally, given a training vector $x_i \in R^n$ and their class values $y_i \in (-1, 1)$, $i = 1, \dots, N$, SVM solves the following optimization problems:

$$\begin{aligned} & \text{Minimize} \quad \frac{1}{2} \omega^T \cdot \omega + C \sum_{i=1}^N \xi_i, \\ & \text{subject to} \quad y_i (\omega^T \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \end{aligned} \quad (6)$$

where ω is a normal vector perpendicular to the hyperplane and ξ_i is slack variables for allowing misclassifications. Here $C (>0)$ is the penalty parameter which balances the trade-off between the margin and training error. In the work, LIBSVM package [26, 27] with radial basis kernel function is used. Two parameters, the regularization parameter C and kernel width parameter γ , are optimized based on 5-fold cross validation using a grid search strategy.

2.6. Evaluation. In statistical prediction, three cross validation tests are often used to evaluate the performance of predictors: subsampling test, independent dataset test, and jackknife test [28, 29]. However, of the three test methods, the jackknife test is deemed the least arbitrary that can always yield a unique result for a given benchmark dataset as elaborated in [30] and demonstrated by (28)–(30) in [21]. Accordingly, the jackknife test has been increasingly and widely used by investigators to examine the quality of various predictors (see, e.g., [31–34]). However, to reduce the computational time, we adopted the 5-fold cross validation in this study as done by many investigators with SVM as the prediction engine.

5-fold cross validation [29] is used in this work. The dataset is randomly divided into five equal sets, out of which four sets are used for training and the remaining one for testing. This procedure is repeated five times, and the final

prediction result is the average accuracy of the five testing sets.

Four parameters, sensitivity (Sn), specificity (Sp), accuracy (Ac), and matthews' correlation coefficient (MCC) are used to measure the performance of our model. They are defined by the following formulas:

$$\begin{aligned} \text{Sn} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ \text{Sp} &= \frac{\text{TN}}{\text{TN} + \text{FP}}, \\ \text{Ac} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \\ \text{MCC} &= \frac{(\text{TP} \times \text{TN}) - (\text{FN} \times \text{FP})}{\sqrt{(\text{TP} + \text{FN}) \times (\text{TN} + \text{FP}) \times (\text{TP} + \text{FP}) \times (\text{TN} + \text{FN})}}, \end{aligned} \quad (7)$$

where TP, TN, FP, and FN are the number of true positive, true negative, false positive, and false negative, respectively. For a given dataset, all these values can be obtained from the decision function with fixed cutoff.

2.7. Incremental Feature Selection (IFS). With the mRMR method, we can rank the importance of the 651 features, and then, we can use Incremental Feature Selection (IFS) [35–38] to determine the optimal number of features. We can create the features set by the features importance rank, such as

$$S_i = \{f_1, f_2, \dots, f_i\}, \quad (1 \leq i \leq N). \quad (8)$$

We can use SVM to predict the performances of each feature set and evaluate the set with the 5-fold cross validation; thus, the optimal feature set can be yielded [39–41].

3. Result and Discussion

3.1. The mRMR Result and IFS Result. In the Supporting Information S5, the mRMR feature table listed the ranked 651 features with the maximum relevance and minimum redundancy to the class of samples. The list of ranked feature was to be used in the following IFS procedure for the optimal feature set selection.

In IFS test (Support Information S5, see Supplementary Material available online at <http://dx.doi.org/10.1155/2013/861269>), we added the feature one by one and built about 651 predictors. In Figure 1, the MCC reached their maximum value when 284 features were used. The accuracy, sensitivity, specificity, and MCC were 0.7949, 0.6231, 0.8812, and 0.5249. Figure 1 showed the MCC plot based on Supporting Information S5.

3.2. Analysis of the Optimal Feature Set. In the IFS procedure, we selected 284 optimal features (Supporting Information S5). In the result, 154 belonged to the PSSM conservation score, 43 to the amino acid factors, 21 to the disorder, 63 to the

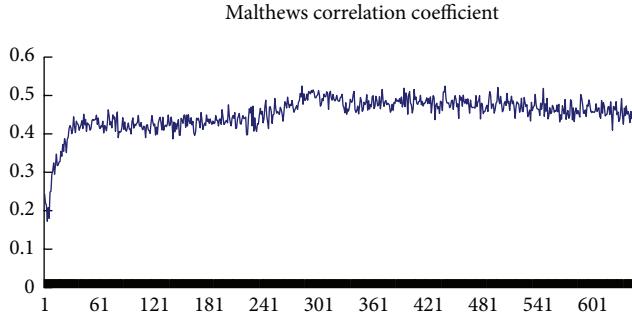


FIGURE 1: Plot of the MCC of different number features.

secondary structure, and 3 to the solvent accessibility. It indicated that PSSM conservation score, amino acid factors and secondary structure played important roles to predict calpain cleavage sites. The optimal feature distribution revealed that site 1, site 7 to site 8, site 10, and site 12 to site 14 played the most important roles in prediction of calpain cleavage sites. Moreover, the features close to calpain cleavage site were more important than central site and the site far from the calpain cleavage site.

3.2.1. PSSM Conservation Score Feature Analysis. As mentioned previously, there were 154 PSSM conservation features, and we found that the conservation against mutations of the 20 amino acids had different impacts on the prediction of calpain cleavage sites. We measured the number of each kind of amino acids for the PSSM features (Figure 2(a)) and found that different mutations of amino acid had different roles in prediction of calpain cleavage sites. Mutation of the amino acid Valine (V), Leucine (L) and Phenylalanine (P) were important in predicting the calpain cleavage sites. In the mRMR feature list, the 2 to 5 ranked features were PSSM features at site 11, site 8, site 6, and site 4 against transition to amino acid Leucine (L). This indicated that the conservation of Valine, Leucine, and Phenylalanine were the keys to determining whether or not it was calpain cleavage site. We also measured the PSSM feature number of each amino acid site (Figure 2(b)). The result revealed that site 5, site 8, site 12, and site 13 were more important in predicting calpain cleavage sites than other sites (shown in Figure 2(b)).

3.2.2. Amino Acid Factor Analysis. We investigated the number of each type of amino acid factor features (Figure 3(a)) and the number of amino acid factors at each site (Figure 3(b)). As a result, the secondary structure was the most important features in predicting the calpain cleavage sites. And the codon diversity was the second important feature to predict calpain cleavage sites. In Figure 3(b), site 10, site 11, and site 21 had relatively more effects on the calpain cleavage sites. Moreover, in the Supporting Information S5, the first feature was the polarity feature, and the polarity feature at site 10, site 12, and site 15 played more roles in the calpain cleavage sites prediction. This indicated that the polarity of the residues located more close to the calpain cleavage site has a critical role in predicting the calpain cleavage site.

3.2.3. Disorder, Secondary Structure, and Solvent Accessibility Feature Analysis.

With the final optimal feature set, there were 63 secondary structure features; 21 disorder features and a reasonable explanation was that the feature of secondary structure and disorder encoding were sensitive for predicting calpain cleavage sites. And in the Supporting information S5, we could show that first index of secondary structure was 189 in site 1, and it was the “stand” feature; also the first index of disorder was 143 at site 1. There were 3 solvent accessibility features in the optimal feature set; they were in sites 1 and 2; however the index was 281, 282, and 284.

3.3. Comparison with Existing Method. According to the mRMR and IFS procedure, the performance of predictor was the best when we selected 284 features. And the accuracy, specificity, sensitivity, Matthews' correlation coefficient was 0.7949, 0.8812, 0.6231, and 0.5249. And we made a comparison with GPS 2.0, GPS 1.0, PoPS, site prediction 1, and site prediction 2. For MCC, our predictor was obviously improved than all other predictors. It indicated that our predictor has excellent performance in predicting positive samples. And in the same value of specificity (~0.90), our sensitivity was 0.6231, higher than the other predictor. But the accuracy of our predictor was slightly worse than those of the others. Since we did not know the ratio of positive and negative in their training set, we built the predictor based on a training set in which the negative samples were two times than positive samples. The sensitivity of our method was 0.6231, and the sensitivity GPS 2.0 was 0.6087, GPS1.0 was 0.5000, PoPS was 0.5245, site prediction 1 was 0.4130, and site prediction 2 was 0.3967, when the threshold was medium.

More importantly, the reasonably good performance of our method reflects that the physicochemical/biochemical properties can effectively capture the information of around calpain cleavage sites. But there still exists some limits in our method. For example, our predictor only considers the amino acid sequence information but does not consider the protein structure features. Therefore, in the future, we should focus on development of amino acid encoding schema and development of a predictor to maximize the prediction performance of calpain cleavage sites.

4. Conclusion

Calpains are an important family of the Ca^{2+} -dependent cysteine proteases which catalyze the limited proteolysis of many specific substrates. Calpains play crucial roles in basic physiological and pathological processes. In the paper, we developed a new predictor based on the support vector machine (SVM) with the maximum relevance minimum redundancy (mRMR) method followed by incremental feature selection (IFS). And we concerned the feature of physicochemical/biochemical properties, sequence conservation, residual disorder, secondary structure, and solvent accessibility to represent the calpain cleavage sites. And we selected 284 optimal features; these features were central to predict the calpain cleavage sites, and with the optimal features set, the accuracy of our predictor was 0.7949, and the sensitivity and

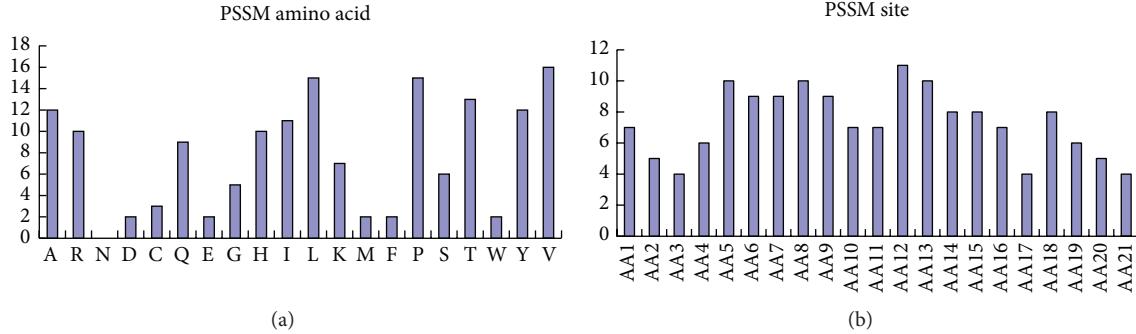


FIGURE 2: Bar plots to show the distribution in the final optimal feature set for (a) the PSSM score and (b) the corresponding specific site score. It was shown from panel (a) that mutations of amino acid Valine (V) played most important role in prediction of calpain cleavage sites; followed by Leucine (L) and Phenylalanine (P). And it was shown from panel (b) that conservation in site 5, site 8, site 12, and site 13 was more important in determining the calpain cleavage sites.

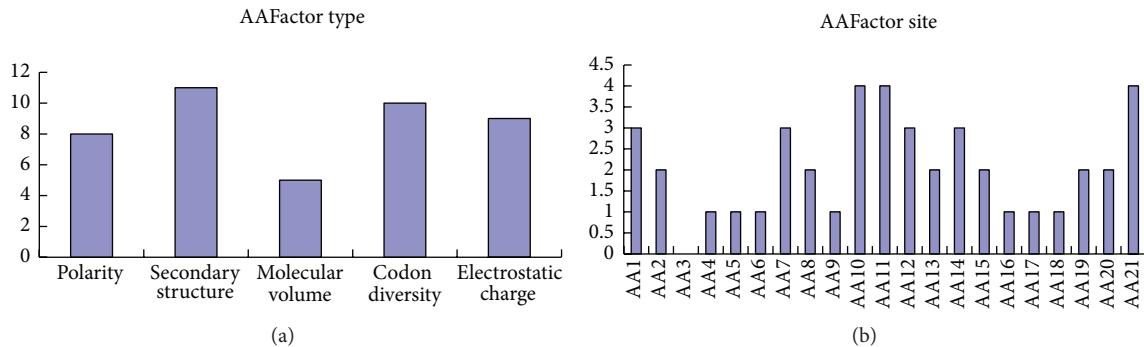


FIGURE 3: Bar plots to show the distribution in the final optimal feature set for (a) the amino acid factor features, and (b) the corresponding specific site score. It can be seen from panel (a) that the secondary structure and codon diversity were the most important one for predicting the calpain cleavage sites. It can be seen from panel (b) that the residues at site 10, site 11, and site 21 played more roles in the calpain cleavage sites prediction.

MCC were especially higher than other predictor. Further, remaining feature would contain more information of calpain cleavage and are needed more analysis in the feature.

Acknowledgments

The work is fully supported by the National Natural Science Foundation of China under Grant no. 60803102, Natural Science Foundation of Jilin Province under Grant nos. 201215006 and 201101004, Fundamental Research Funds for the Central Universities under Grant no. 11CXYPY010, and Opening Fund of Top Key Discipline of Computer Software and Theory in Zhejiang Provincial Colleges at Zhejiang Normal University of China under Grant no. ZSDZZZZXK37.

References

- [1] S. J. Storr, N. O. Carragher, M. C. Frame, T. Parr, and S. G. Martin, “The calpain system and cancer,” *Nature Reviews Cancer*, vol. 11, no. 5, pp. 364–374, 2011.
- [2] I. Bertipaglia and E. Carafoli, “Calpains and human disease,” *Subcellular Biochemistry*, vol. 45, pp. 29–53, 2007.
- [3] S. J. Franco and A. Huttenlocher, “Regulating cell migration: calpains make the cut,” *Journal of Cell Science*, vol. 118, no. 17, pp. 3829–3838, 2005.
- [4] Y. X. Fan, Y. Zhang, and H. B. Shen, “LabCaS: labeling calpain substrate cleavage sites from amino acid sequence using conditional random fields,” *Proteins*, vol. 81, no. 4, pp. 622–634, 2013.
- [5] D. E. Croall and K. Ersfeld, “The calpains: modular designs and functional diversity,” *Genome Biology*, vol. 8, no. 6, article 218, 2007.
- [6] Z. Liu, J. Cao, X. Gao, Q. Ma, J. Ren, and Y. Xue, “GPS-CCD: a novel computational program for the prediction of calpain cleavage sites,” *PLoS One*, vol. 6, no. 4, Article ID e19001, 2011.
- [7] P. Tompa, P. Buzder-Lantos, A. Tantos et al., “On the sequential determinants of calpain cleavage,” *Journal of Biological Chemistry*, vol. 279, no. 20, pp. 20775–20785, 2004.
- [8] S. E. Boyd, M. G. De La Banda, R. N. Pike, J. C. Whisstock, and G. B. Rudy, “PoPS: a computational tool for modeling and predicting protease specificity,” in *Proceedings of the IEEE Computational Systems Bioinformatics Conference (CSB '04)*, pp. 372–381, August 2004.
- [9] S. E. Boyd, R. N. Pike, G. B. Rudy, J. C. Whisstock, and M. G. De La Banda, “PoPS: a computational tool for modeling and predicting protease specificity,” *Journal of Bioinformatics and Computational Biology*, vol. 3, no. 3, pp. 551–585, 2005.
- [10] J. Versperuten, K. Gevaert, W. Declercq, and P. Vandenebeele, “SitePredicting the cleavage of proteinase substrates,” *Trends in Biochemical Sciences*, vol. 34, no. 7, pp. 319–323, 2009.

- [11] K. C. Chou and H. B. Shen, "Review: recent advances in developing web-servers for predicting protein attributes," *Natural Science*, vol. 2, pp. 63–92, 2009.
- [12] K.-C. Chou, "The convergence-divergence duality in lectin domains of selectin family and its implications," *FEBS Letters*, vol. 363, no. 1-2, pp. 123–126, 1995.
- [13] B.-Q. Li, L.-L. Hu, S. Niu, Y.-D. Cai, and K.-C. Chou, "Predict and analyze S-nitrosylation modification sites with the mRMR and IFS approaches," *Journal of Proteomics*, vol. 75, no. 5, pp. 1654–1665, 2012.
- [14] S. F. Altschul, T. L. Madden, A. A. Schäffer et al., "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic Acids Research*, vol. 25, no. 17, pp. 3389–3402, 1997.
- [15] Y. N. Zhang, D. J. Yu, S. S. Li, Y. X. Fan, Y. Huang, and H. B. Shen, "Predicting protein-ATP binding sites from primary sequence through fusing bi-profile sampling of multi-view features," *BMC Bioinformatics*, vol. 13, article 118, 2012.
- [16] S. Kawashima and M. Kanehisa, "AAindex: amino acid index database," *Nucleic Acids Research*, vol. 28, no. 1, article 374, 2000.
- [17] W. R. Atchley, J. Zhao, A. D. Fernandes, and T. Dröge, "Solving the protein sequence metric problem," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 18, pp. 6395–6400, 2005.
- [18] K. Peng, P. Radivojac, S. Vucetic, A. K. Dunker, and Z. Obradovic, "Length-dependent prediction of protein in intrinsic disorder," *BMC Bioinformatics*, vol. 7, article 208, 2006.
- [19] J. Cheng, A. Z. Randall, M. J. Sweredoski, and P. Baldi, "SCRATCH: a protein structure and structural feature prediction server," *Nucleic Acids Research*, vol. 33, no. 2, pp. 72–76, 2005.
- [20] K.-C. Chou, "Some remarks on protein attribute prediction and pseudo amino acid composition," *Journal of Theoretical Biology*, vol. 273, pp. 236–247, 2011.
- [21] K.-C. Chou, "Prediction of protein cellular attributes using pseudo amino acid composition," *Proteins*, vol. 43, pp. 246–255, 2001, Erratum in *Proteins*, vol. 44, pp. 60, 2001.
- [22] K.-C. Chou, "Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes," *Bioinformatics*, vol. 21, no. 1, pp. 10–19, 2005.
- [23] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: criteria of Max-Dependency, Max-Relevance, and Min-Redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.
- [24] X. Zhao, Z. Ma, and M. Yin, "Using support vector machine and evolutionary profiles to predict antifreeze protein sequences," *International Journal of Molecular Sciences*, vol. 13, no. 2, pp. 2196–2207, 2012.
- [25] X.-W. Zhao, Z.-Q. Ma, and M.-H. Yin, "Predicting protein-protein interactions by combining various sequence-derived features into the general form of chou's pseudo amino acid composition," *Protein and Peptide Letters*, vol. 19, no. 5, pp. 492–500, 2012.
- [26] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machine," *CM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article 27, 2001.
- [27] LIBSVM, 2012, <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.
- [28] X. Zhao, X. Li, Z. Ma, and M. Yin, "Prediction of lysine ubiquitylation with ensemble classifier and feature selection," *International Journal of Molecular Sciences*, vol. 12, no. 12, pp. 8347–8361, 2011.
- [29] K.-C. Chou and C.-T. Zhang, "Prediction of protein structural classes," *Critical Reviews in Biochemistry and Molecular Biology*, vol. 30, no. 4, pp. 275–349, 1995.
- [30] K.-C. Chou and H.-B. Shen, "Cell-PLoc: a package of Web servers for predicting subcellular localization of proteins in various organisms," *Nature Protocols*, vol. 3, no. 2, pp. 153–162, 2008.
- [31] H. Mohabatkar, "Prediction of cyclin proteins using chou's pseudo amino acid composition," *Protein and Peptide Letters*, vol. 17, no. 10, pp. 1207–1214, 2010.
- [32] X. T. Li and M. H. Yin, "Application of differential evolution algorithm on self-potential data," *Plos One*, vol. 7, no. 12, Article ID E51199, 2012.
- [33] X. W. Zhao, W. Y. Zhang, X. Xu, Z. Q. Ma, and M. H. Yin, "Prediction of protein phosphorylation sites by using the composition of k-spaced amino acid pairs," *Plos One*, vol. 7, no. 10, Article ID E46302, 2012.
- [34] X. T. Li, J. Zhang, and M. H. Yin, "Animal migration optimization: an optimization algorithm inspired by animal migration behavior," *Neural Computing and Applications*, 2013.
- [35] Z. He, J. Zhang, X.-H. Shi et al., "Predicting drug-target interaction networks based on functional groups and biological features," *PLoS One*, vol. 5, no. 3, Article ID e9603, 2010.
- [36] T. Huang, W. Cui, L. Hu, K. Feng, Y.-X. Li, and Y.-D. Cai, "Prediction of pharmacological and xenobiotic responses to drugs based on time course gene expression profiles," *PLoS One*, vol. 4, no. 12, Article ID e8126, 2009.
- [37] X. T. Li and M. H. Yin, "An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure," *Advances in Engineering Software*, vol. 55, pp. 10–31, 2013.
- [38] X. Li, J. Wang, J. Zhou, and M. Yin, "A perturb biogeography based optimization with mutation for global numerical optimization," *Applied Mathematics and Computation*, vol. 218, no. 2, pp. 598–609, 2011.
- [39] D. B. Cai and M. H. Yin, "On the utility of landmarks in SAT based planning," *Knowledge-Based Systems*, vol. 36, pp. 146–154, 2012.
- [40] J. P. Zhou, P. Huang, M. H. Yin, and C. G. Zhou, "Phase transitions of EXPSPACE-complete problems," *International Journal of Foundations of Computer Science*, vol. 21, no. 6, pp. 1073–1088, 2010.
- [41] J. P. Zhou, M. H. Yin, X. T. Li, and J. Y. Wang, "Phase transitions of EXPSPACE-complete problems: a further step," *International Journal of Foundations of Computer Science*, vol. 23, no. 1, pp. 173–184, 2012.

Research Article

A New Skeleton Feature Extraction Method for Terrain Model Using Profile Recognition and Morphological Simplification

Huijie Zhang,^{1,2} Zhiqiang Ma,^{1,2} Yaxin Liu,^{1,2} Xinting He,^{1,2} and Yun Ma^{1,2}

¹ School of Computer Science and Information Technology Northeast Normal University, Changchun 130117, China

² Key Laboratory of Intelligent Information Processing of Jilin Universities, Changchun 130117, China

Correspondence should be addressed to Zhiqiang Ma; mazq@nenu.edu.cn

Received 10 July 2013; Accepted 12 August 2013

Academic Editor: William Guo

Copyright © 2013 Huijie Zhang 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.

It is always difficult to reserve rings and main truck lines in the real engineering of feature extraction for terrain model. In this paper, a new skeleton feature extraction method is proposed to solve these problems, which put forward a simplification algorithm based on morphological theory to eliminate the noise points of the target points produced by classical profile recognition. As we all know, noise point is the key factor to influence the accuracy and efficiency of feature extraction. Our method connected the optimized feature points subset after morphological simplification; therefore, the efficiency of ring process and pruning has been improved markedly, and the accuracy has been enhanced without the negative effect of noisy points. An outbranching concept is defined, and the related algorithms are proposed to extract sufficient long trucks, which is capable of being consistent with real terrain skeleton. All of algorithms are conducted on many real experimental data, including GTOPO30 and benchmark data provided by PPA to verify the performance and accuracy of our method. The results showed that our method precedes PPA as a whole.

1. Introduction

Terrain model is an important component in the actual engineering about 3D geographic information system and is applied widely in the field of spatial information visualization, including battleground simulation, flight simulation training, and terrain deformation simulation. However, it is a big challenge to construct a terrain model, because terrain model usually contains massive data, and its feature is rather complex. A valid approach of building terrain model is to extract the key feature of terrain instead of using all of the terrain data. The aim is not only to guarantee the accuracy of terrain model but also to decrease the data scale.

Over the years, many skeleton feature extraction methods have been proposed, which are mainly based on terrain flow simulation [1–3] or geometrical gradient analysis [4–6]. The moving-window algorithm is an early classical algorithm, which takes elevation map as 2D image [4]. Although this algorithm can extract the basic outline feature of terrain, it does not consider the morphological information but just connects simply the extreme value of elevation in 2×2 window into feature lines. Therefore, the accuracy is very low.

Another classical algorithm is deterministic eight nodes (D8) proposed by O'Callaghan and Mark, which obtains the ridge lines by simulating the procedure of the surface water flow from 8 directions [7]. A series of improved algorithms are proposed based on D8 [8–11]. However, since drainage accumulation matrixes and flow direction matrixes are complex so as to take a lot of time overhead to calculate, the efficiency is not acceptable in many application fields.

Furthermore, for some special terrain, such as depression and flat, flow simulation algorithms are not capable of extracting the feature lines correctly [12]. The main reason is that flow simulation algorithm depends on the outlet to establish the flow direction matrixes. Nevertheless, it is possible that there is no outlet for depression and flat. Hence, depression filling algorithm is a relative valid approach to solve this problem [13–15]. However, it may increase the area of flat and thus suffer from the new problems [16]. In addition, depression filling also leads to extra time consumption. Barnes proposed the simplified algorithm to improve the efficiency [15]. For terrain feature of flat, researchers also proposed the related strategies, including graph structure [17],

elevation increments method [18], and radial basis function [19].

Because of the above limitations, most flow simulation algorithms cannot guarantee both the precision of the feature extraction and the global morphology of the feature lines. Chang et al. raise the Profile recognition and the Polygon-breaking Algorithm (PPA) [20]. Its advantage is not to be influenced by depression and flat and to be capable of extracting the global skeleton feature in most instances.

In recent years, PPA is applied in various terrain modeling, including terrain synthesis [21] and terrain deformation [22–28], because of its fine characteristics mentioned. Zhou et al. utilized PPA to extract the skeleton feature of terrain and realized the terrain synthesis [21]. Gain improved the terrain synthesis and proposed the method based on user's sketch map [29]. In addition, there are other applications of PPA in terrain visualization, including multiresolution terrain deformation [22] and enhanced terrain synthesis based on GPU [30]. In order to deal with global large scale data, the related work, such as artificial intelligence planning [31] and the differential evolution algorithm [32, 33], has also referential significance.

Nevertheless, PPA cannot reserve the ring feature, and the accuracy and stability of algorithm need to be improved further. In this paper, we proposed a new method, Morphological Profile recognition and Polygon breaking (MPPA). Its primary advantage is that MPPA is capable of recognizing the stable rings and reserving them. Moreover, MPPA defined the outbranching in order to remain the sufficient long truck lines, which is consistent with real terrain feature. Furthermore, the noise points have been eliminated by our simplification algorithm based on morphology; therefore, the whole performance of MPPA has been improved greatly.

2. Target Recognition and Feature Simplification

The original data of terrain model is usually stored as the form of Digital Elevation Model (DEM). It is a large scale data set. It is necessary to extract the significant data which contains the skeleton structure information of the terrain from the massive data set. In our work, profile recognition strategy is adopted to recognize the initial feature points as candidate target points, which are similar to the target in PPA. Its advantage is that it can almost cover most of the feature points.

PPA connected all of the candidate target points into polygon areas and then disassembled those closed polygons in order to extract the ridge axes. Owing to there are a lot of noisy points, the efficiency of algorithm is really low. Moreover, its principle of removal of noisy points is the inexistence of the closed polygon, which did not accord with the original ridge morphology. Different from PPA, our method proposed a new strategy to refine the initial feature points by morphological simplification algorithm, which can eliminate a lot of noisy points in large quantity. And then we utilized the simplified feature points, that is, subset of initial target points, to exact the skeleton feature lines.

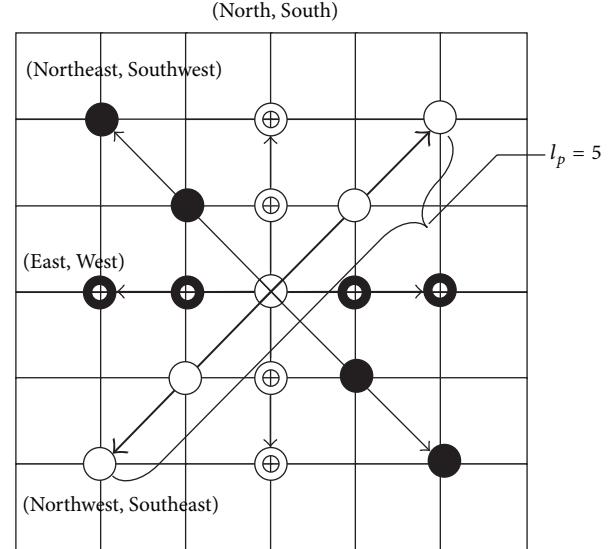


FIGURE 1: Profile recognition.

2.1. Profile Recognition. The profile recognition is a classical method that extracts the candidate target points from the original DEM data set [20]. For the adjacent points on the eight directions of the current central point, they are divided into four couples as (North, South), (East, West), (Northeast, Southwest), (Northwest, Southeast), as shown in Figure 1. Each couple represents a profile of terrain model, which has been identified by mean of different shapes in Figure 1.

Let the number of points on each profile be the profile length as l_p , which is closely related to recognition result of feature points. In this paper, we set l_p as 5, that is to say, there are two adjacent points at each side of the current point in each profile. Afterwards, the current point is determined whether it is a candidate target points according to the relationship between the elevation data of the current point with that of its adjacent points on different profiles. The definition of the candidate feature points set is as follows.

Definition 1. Let V be an original set of elevation points, let $v_{r,c} \in V$ be current point, let $m = \lfloor l_p/2 \rfloor$ be the number of adjacent points for each profile, and let $P_{r,c}^n = \{v_{r\pm\Delta,c\pm\Delta} \mid 0 \leq \Delta \leq m\}$ be the set of adjacent points on the n th profile; then $v_{r,c} \in V$ is a candidate feature point, if the following constraints are satisfied:

- (1) on all the profiles, there exist at least two points $p_1 \in P_{r,c}^n$ and $p_2 \in P_{r,c}^n$; their elevations are lower than that of $v_{r,c}$;
- (2) p_1 and p_2 should be at the different side on the same profile,

where l_p is profile length, (r, c) are coordinates of current points, and $n = 1, 2, 3, 4$ is the sequence number of profile.

By using Definition 1, a set of candidate target points will be extracted, which almost covered the whole ridge feature of the terrain. It is important to note that the constraints are

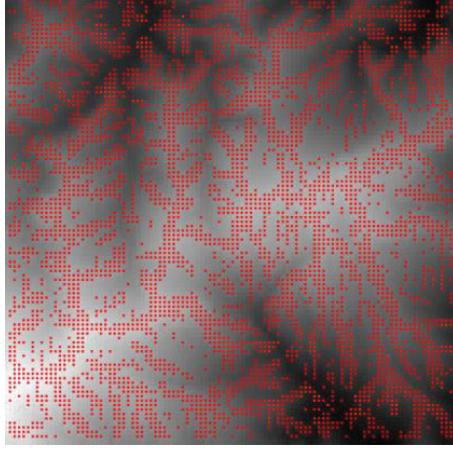


FIGURE 2: Target points.

loose enough so as to prevent from missing the real target points. As a result, it is a data set with a lot of redundant noise points, but its whole feature information has been reserved. Figure 2 shows the set of candidate target points which we have obtained by using profile recognition.

2.2. Morphological Simplification. Morphology has presented its superior properties in the field of image processing. In our work, we adopted the morphological idea to solve the simplification problem of candidate target points. In this section, we defined a suit of binary coding sequence and then proposed a new simplification strategy for candidate target points according to the terrain morphology features. It is noteworthy that efficiency of our skeleton extraction method MPPA will be improved greatly, because the feature points are connected into polygon strips after the candidate target points have been simplified by our strategy.

Let $S = (b_i = 0 \text{ or } 1 \mid i = 0, 1, 2, \dots, 8)$ be a binary coding sequence, and the mapping relationship between the bit code and the position of elevation point is presented by the following:

$$\begin{aligned} r &= \left\lfloor \frac{i}{3} \right\rfloor + 1, \\ c &= \mod(i, 3) + 1, \\ i &= (r - 1) * 3 + c - 1, \end{aligned} \tag{1}$$

where (r, c) is the spatial position of the elevation point $v_{r,c}$.

Obviously, it is easy to implement Formula (1). Nevertheless, we can determine the corresponding spatial position of each bit in S according to the mapping relationship. Moreover, there exists a corresponding decimal value $d_S = \sum_{i=0}^8 2^{b_i}$ for each S , and thus each d_S can represent a kind of spatial morphology of terrain, named morphological coding. Since S consists of nine bits, the value range of d_S is $2^0 \sim 2^9$. Hence, morphological coding d_S can stand for 512 types of morphology of terrain at most. In our work, we defined 48 kinds of morphologies to simplify the candidate target points, shown in Figure 3. Figure 3(a) described the mapping

relationship between (r, c) and d_S . In Figure 3(b), each morphological coding stands for a kind of morphology, which has been denoted as d_S . However there are only 12 types of morphologies illustrating in Figure 3(b). And the other ones can be deduced by rotating these instances.

For each candidate target point, we detect the terrain morphology feature according to the relationship between it with its 8 adjacent points. If the morphology feature matched any morphological coding above-mentioned, then the current point should be removed from the set of candidate target points. The morphology feature $MF_{(r,c)}$ of the current point is calculated by

$$MF_{(r,c)} = \sum_{\Delta r=0,-1,1, \Delta c=0,-1,1} v_{r+\Delta r, c+\Delta c} * 2^i, \tag{2}$$

where the value range of $v_{r+\Delta r, c+\Delta c}$ is 0 or 1 and $v_{r,c} = 1$ if $v_{r,c}$ is a candidate target point; otherwise $v_{r,c} = 0$.

The detail steps of simplification algorithm about candidate target points are described in Algorithm 1.

According to Algorithm 1, it is not hard to see that we take the morphology feature $MF_{(r,c)}$ of the current point as the index in the contiguous space of MTC. Therefore, searching procedure is not necessary to recognize the morphology of terrain and then determine whether the current point should be eliminated from the set of candidate target points. As a result, the efficiency of algorithm is very high. In our experiment for real data, the outermost iteration is approximate 6 times to complete the simplification procedure.

2.3. Connection of Feature Points. Morphology simplification is significant to improve the efficiency of MPPA. The main reason is that the number of optimized feature points is less than that of the initial candidate target points. And then we connect the optimized set of feature points into the polygon strips by the following rules.

- (1) For each feature point, connect with 4 adjacent feature points, shown in Figure 4, and 1, 2, 3, and 4 represent the direction coefficients.
- (2) Eliminate the edge, which the average of elevation of its two end points is lower, if there are two edges to be cross.

If there is edge from current point $v_{r,c}$ to the point in direction 2, at the same time there is edge from $v_{r,c+1}$ to the point in its direction 3, the cross phenomenon will appear, shown in Figure 4. The Rule (2) is a valid method to solve the above problems. Finally, the optimized feature segments composed by triangles and edges are obtained.

3. Ring Process

The most skeleton features of terrain are presented by tree topology. Hence, PPA breaks the polygon segments until there are not the closed polygons. Nevertheless, the real terrain contains a lot of atoll textures. As a result, tree topology is not enough to cover all of the skeleton features of real terrain. Therefore, we utilize graphic structure to store the

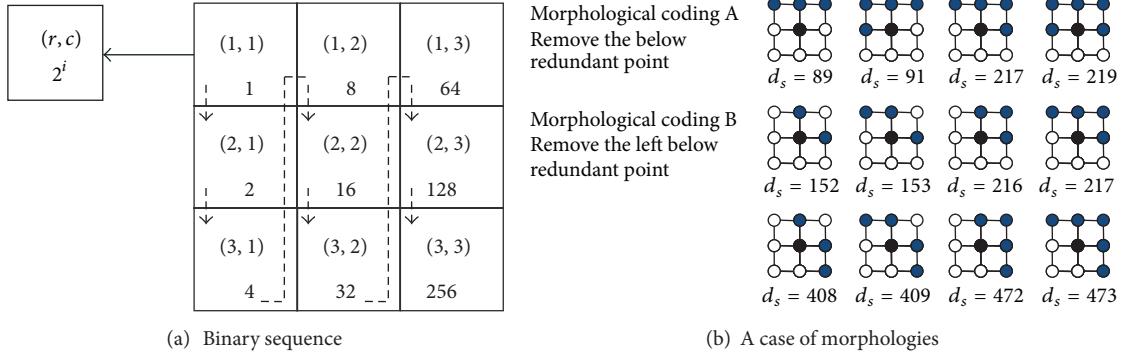


FIGURE 3: Morphological coding.

Input: The set V_f of candidate target points.
Output: The optimized set V_f of feature points.
1. Assign the 512 bits of contiguous space MTC to store the state of morphological coding.
2. Initialize the designed 48 kinds of morphologies in MTC into 1, other bits should be set into 0.
3. If there is no change in V_f , then go to step 4; otherwise do the following loop:
3.1 For each $v_{r,c} \in V_f$, calculate its morphology feature $MF_{(r,c)}$ using formula (2);
3.2 Evaluate whether $MF_{(r,c)}$ matches one of 48 kinds of morphologies;
3.3 Eliminate $v_{r,c}$ from set V_f , if $MTC[MF_{(r,c)}]$ equals to 1;
4. Algorithm end.

ALGORITHM 1: Simplification of target points.

Input: Feature segments set FS with triangles and single edges.
Output: Feature segments set FS without any little triangles.
1. Traverse all of the feature segments:
1.1 If there is not any case of 4 kinds of triangles of Figures 5(a)–5(d) into set T , go to step 5; otherwise go to step 1.2;
1.2 For $t \in T$, evaluate whether its three edges are shared;
1.3 Denote the edge as 2, if it has been shared by two triangles; otherwise denote it as 1 and put it into set E .
2. Sort edges in set E in ascending order.
3. Remove the edge $e \in E$ from FS and E , if e satisfies the conditions as shown in Figures 5(a)–5(d).
4. Go to step 1.
5. Algorithm end.

ALGORITHM 2: Polygon disassembling.

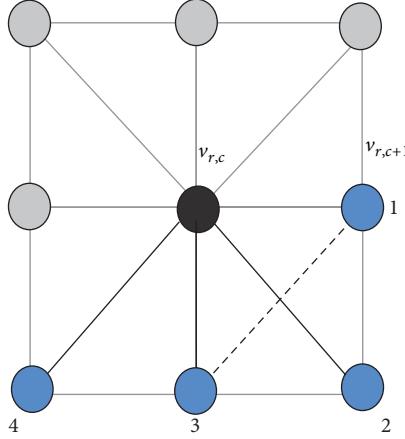


FIGURE 4: Direction coefficient.

skeleton features. However, not all of the rings are expected in the final result. Those single polygons and those sufficiently small rings should be disassembled, while big rings that can stand for the certain terrain feature are reserved by our method MPPA.

3.1. Polygon Disassembling. In MPPA, we take those rings that do not contain any nontarget points as single polygons. Figures 5(a)–5(d) described the 4 kinds of triangles that need to be disassembled. Let $v_{r,c}$ be the current point; we just consider the triangles in the 4 directions as shown in Figure 4, since those in other directions can be processed by its left-top points.

For those triangles that accord with morphologies in Figure 5, we will break the edge with lower elevation. However, the undesirable parallelograms will appear, when two triangles shared a common edge, shown as dot lines in Figures 5(e)–5(h). Hence, the edge shared by triangles will be remained (denoted by 2 in Figure 5), while the outer edges will be removed (denoted by 1 in Figure 5) in that case. The detailed procedure is described by Algorithm 2.

3.2. Judgment of Connected Domain. Besides breaking polygons, MPPA focuses on the ring process. In fact, there are few small rings in real terrain. It is reasonable that MPPA breaks the small rings but remains the big rings. Hence, it is important to evaluate the size of ring. The filling intensity is proposed in this paper to resolve this problem.

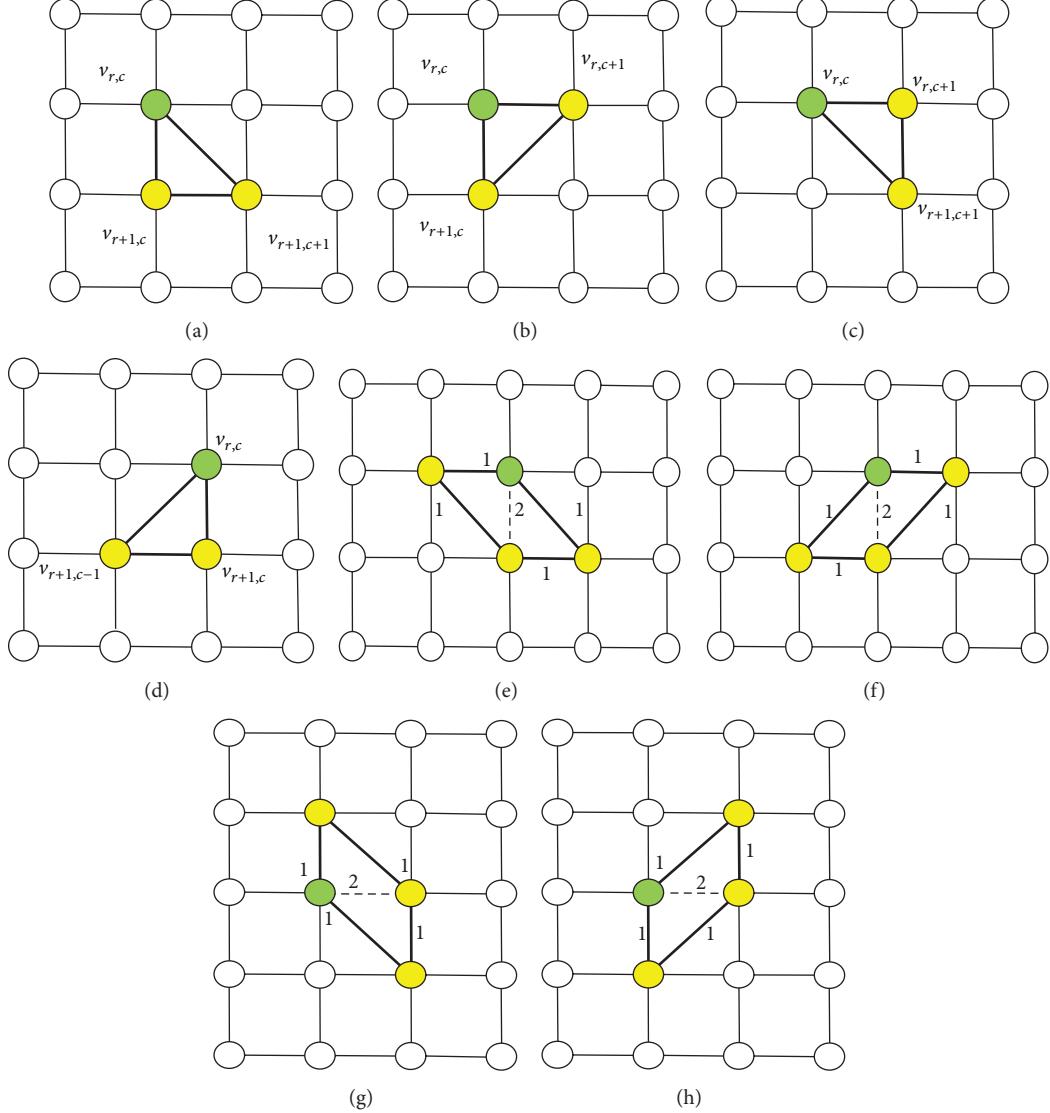


FIGURE 5: Triangles breaking.

Definition 2. Let V_f and V_n be feature points set and nonfeature points set, respectively, and denote $v_n \in V_n$ as a common label, if v_n is enclosed in a closed area by the feature points of V_f . The count of v_n with the same label is called filling intensity.

By Definition 2, it is not difficult to find that filling intensity is the count of the interconnected points. Afterwards, we proposed the algorithm to judge the connected domain and calculate the filling intensity.

In Algorithm 3, each point is traversed once at most; therefore it does not bring big time overhead. Nevertheless, it is necessary to process the rings and reserve the long branches of skeleton features. As you can see, all of the nonfeature points have been assigned in the certain connected

domain, parts of which have closed into rings, shown in Figure 6.

3.3. Eliminate the Small Rings. For these small rings whose filling intensity is smaller than the criterion, MPPA consider it as the inauthentic terrain features. Hence, it is an important task to eliminate the sufficiently small rings. In Section 3.2, we have denoted the signs and calculated the filling intensity for each connected domain, which made it relative easy to remove the insignificant small rings.

By polygon breaking in Section 3.1, the basic skeleton features of terrain have come into being in substance. Afterwards, the main task is to denote those segments, as borders of small rings, and then delete them. The following rules

```

Input: Non-feature points set  $V_n$ .
Output: The connected domain matrix DOM, and its
filling intensity  $\text{Ins}_i, i = 1, 2, 3, \dots$ 
1. Set  $\text{Ins}_i = 0, i = 1, 2, 3, \dots$ 
2. Set  $i = 0$  and  $\text{DOM}(r, c) = 0, 0 \leq r, c \leq m$ ,
where  $m$  is the scale of original data.
3. Find one point that is neither target point, nor
marked point from  $V_n$ , let  $v_{r,c} \in V_n$  as the
current non-feature point,  $i = i + 1$  and mark it, go to
step 3.1; there is no such point go to step 4:
3.1 Initialize array  $A$ ;
3.2 Set  $k = 1$  and  $A[k] = v_{r,c}$ ;
3.3 Set  $\text{Ins}_i = \text{Ins}_i + 1$  and  $\text{DOM}(r, c) = i$ ;
3.4 Find one point that is neither target point, nor
the marked point in eight adjacent points of
current point, let  $v_{rr,cc}$  be the point which is
found at this time, if find the point go to
step 3.6; if there is no such point,  $k = k - 1$ ,
 $v_{r,c} = A[k]$ , go to step 3.5.
3.5 If  $k = 1$ , go to step 4.
3.6 Mark  $v_{rr,cc}, k = k + 1, A[k] = v_{rr,cc}, v_{r,c} = v_{rr,cc}$ ,
go to step 3.3.
4. Algorithm end.

```

ALGORITHM 3: Connected domain judgment.

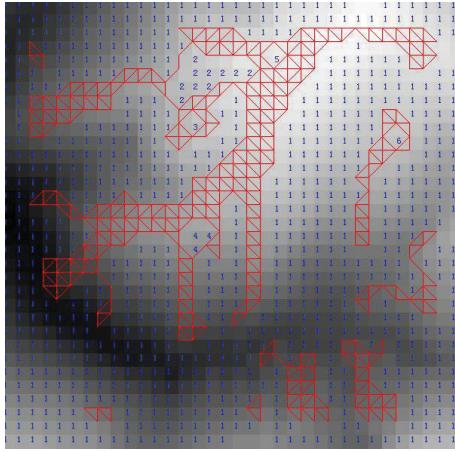


FIGURE 6: Connected domain.

are proposed to evaluate whether those segments should be eliminated.

- (1) Filling intensity of area where any adjacent point on both sides of the current segment belongs is less than the criterion.
- (2) Points on both sides of the current segment belong to different domains according to DOM matrix.

Figure 7 described the distributing instances of points on both sides of the current segment, which bold line shows the current segment detected, green point indicates the start point of segment, and the yellow point indicates the end point of segment. The other points denote the detected points on both sides of the current segment. In order to distinguish

the connected domain, we use $\text{DOM}(r, c) = n_i$ to mark the different domains. If the above rules are satisfied, the current segment should be denoted and put them into a set. In terms of this strategy, those segments are traversed and processed in the same way. Finally, the segment with the lowest average elevation of its two end points should be eliminated. As a result, the undesirable small rings are disassembled.

3.4. Evaluate and Reserve Stable Big Rings. It is important to recognize accurate skeleton features with ring and prevent extracting wrong features; therefore not all the big rings should be reserved. It is a key problem to recognize the stable rings and remain them in the skeleton features. At the same time, unstable big rings still need to be disassembled.

By using the same method as in Section 3.2, we find out the big rings and determine the segments which belong to those big rings. By contrast with Rule (1) in Section 3.3, the opposite rule is adopted as the evaluation principle of big rings, that is to say, the filling intensity is bigger than the criterion. However, Rule (2) in Section 3.3 is also suitable for the judgment of big rings.

Nevertheless, these rules are not sufficient to judge the stability of big rings. An additional and significant rule is that the segments that comprised the big rings should be composed by the thick points. It is important to note that the above judgment is completed by original target points before the morphology simplification is performed. If the big ring contained some single segments composed by sparse target points, there is reason to believe that the big ring is unstable and this phenomenon is aroused by noise points.

In addition, the reasons for using the original target points to judge the density of segments on big rings are in order to avoid the effect of omitted noise points during the procedure of morphology simplification. Algorithm 4 described the detailed steps to evaluate and disassemble the unstable big rings. It is noteworthy that reduction of single branches around rings should have been done before Algorithm 4.

4. Customizable Pruning of Skeleton Features

Although the skeleton features have been extracted by the above stages, there are some redundant little branches, shown in Figure 8. Moreover, the skeleton features that only contain the trunk or the branches with specified step length are usually required in many application fields. However, there are few extraction methods that can provide the personalized strategy of pruning.

4.1. Recognize Outbranch Points. In our MPPA, we proposed the customizable pruning strategy to solve the above problems. Since the branches whose step lengths are equal to one are bound to be undesirable, they should be pruned. In Figure 8, the blue lines described the little branches, whereas, those long branches, especially for main lines, should be remained, even by pruning for many times. Therefore, we proposed the method to recognize the outbranches and reserve them.

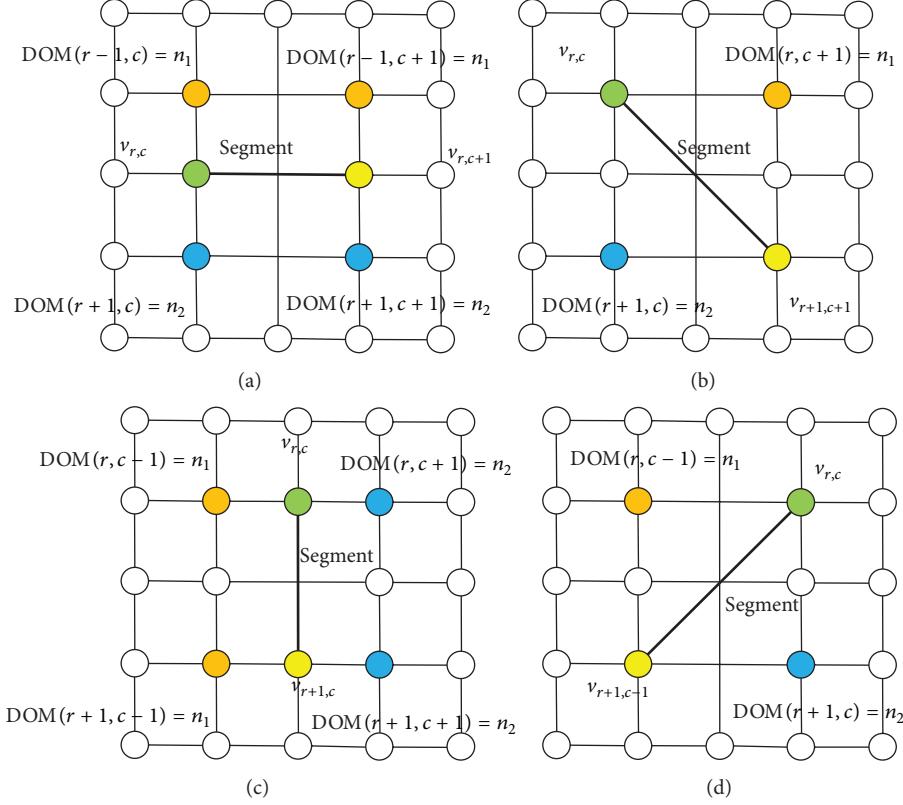


FIGURE 7: Evaluation of segment eliminating.

The premise of recognizing the outbranches is to find the branch points. In order to clearly express the concept of branch points, the formalized describing is as follows.

Definition 3. Let $v_{r,c}$ be a feature point, and let n be the number of segments that are connected directly with $v_{r,c}$; then $v_{r,c}$ is also a branch point, if $n > 2$.

Based on Definition 3, the outbranch point is defined as follows.

Definition 4. Let $v_{r,c}$ be a branch point, and let n be the number of branch points that are connected directly with $v_{r,c}$; then $v_{r,c}$ is also an outbranch point, if and only if $n = 1$.

As shown in Figure 9, the branches that are denoted by red squares are outbranches. It is very important to reserve the main feature lines. According to Definition 4, the outbranches must be located at the end of the skeleton features. If this kind of branches can be preserved, then the main feature lines will not be influenced during the period of the repeated pruning. Algorithm 5 described the procedure to recognize the outbranches.

4.2. Preremove Outbranches. In fact, the outbranching points are the border of feature lines. Moreover, the skeleton feature of real terrain is expected to be as long as possible and be consecutive. The main aim of recognizing outbranching points is to reserve the long and continuous feature lines. If

all of branches are treated with by the uniform principle, then the feature lines will become shorter. Hence, we proposed the concept of the outbranch based on the outbranching point.

Definition 5. Let S_b be an outbranch points set, and let e_b be a branch in direction of end point of $b \in S_b$; then e_b is an outbranch, if e_b is the longest branch in the end segments that connected with b .

The red lines in Figure 10 showed the longest branch of each outbranching point. Those longest outbranches are preserved, so that these significant branches will not be influenced during the pruning, that is to say, the main trunk of skeleton features are still long enough, although most of insignificant branches have been eliminated.

4.3. Process of Pruning. By the process of rings and the preservation of outbranch mentioned before, our MPPA method can extract more precise skeleton features of real terrain. At the same time, it is a customizable method, which can determine the pruning length according to user's different requirements. The detailed steps are presented in Algorithm 6.

By Algorithm 6, the final personalized skeleton features have been extracted, as shown in Figure 11. As we can see, the final skeleton features have reserved the longer main ridge lines, and those insignificant little branches have been eliminated, as shown in Figure 11(a). Moreover, it is noteworthy that a stable ring has been remained in the final skeleton features, which are in accordance with the real terrain

Input: Segments set seg_{res} without polygons and small rings.
Output: Segments set seg_{br} and segments set seg_{res} that has been disassembled.

1. Traverse segments set seg_{res} . If all the segments have been traversed, then go to step 5.
2. Let $s \in \text{seg}_{\text{res}}$ as the current segment:
 - 2.1 Find the adjacent points of s as Figure 7, and evaluate the filling intensities ins of these points, if $\text{ins} < \text{criterion}$, then go to step 1;
 - 2.2 Evaluate whether s satisfies the Rule (2) in Section 3.3.
3. If the points (Figure 7) on the both sides of s are non-target points according to the original target points set, and then:
 - 3.1 Disassemble the rings and remove s from seg_{res} ;
 - 3.2 Otherwise, denote s as the segment of the stable ring and put it into seg_{br} .
4. If there is no any change for the segments set seg_{res} , then go to step 5.
5. Algorithm end.

ALGORITHM 4: Evaluate and process the big rings.

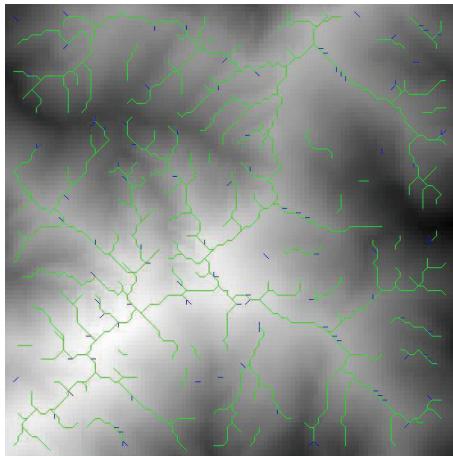


FIGURE 8: Little branches.

features in the natural world. It is not hard to find that the screenshot in Figure 11(b) is the known Crater Lake. Obviously, the ring skeleton feature has been extracted clearly.

5. Experimental Results and Discussion

To test validity of the results extracted by our MPPA, many experiments have been conducted. Since PPA is applied widely in synthesis and deformation of terrain [22, 27, 30], we compare our method MPPA with the classical method PPA. The experimental platform is a PC with a 3.0 GHz CPU, DDR3 1600 MHz 4 GB main memory, and Nvidia Geforce GTX 560 (1 GB) graphic card, and the experiment program is coded in VS2008.

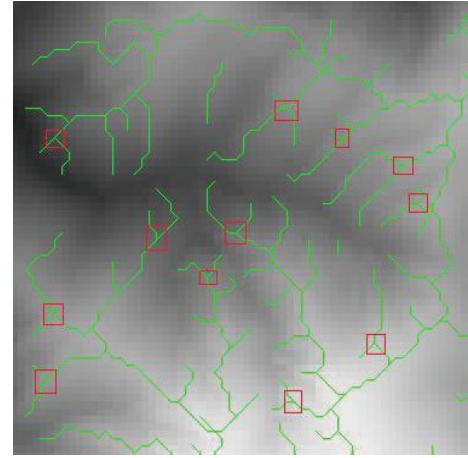


FIGURE 9: Branch points.

Input: Array A_b with branch points signs.
Output: Array A_o with out-branching points signs.

1. Let n as the count of the adjacent branch points of the current point.
2. For each branch point p_b in A_b , do:
 - 2.1 Set $n = 0$;
 - 2.2 For each branch connected with p_b , do:
 - 2.2.1 Find the end point p_a of each branch connected with p_b , according to the direction coefficients of p_b ;
 - 2.2.2 If p_a is an end point that has not any direction coefficient, then go to 2.2; otherwise, if p_a is an out-branching point then set $n = n + 1$, and go to 2.2.
 - 2.3 If n is equal to 1, denote p_b as out-branch point in A_o .
3. Algorithm end.

ALGORITHM 5: Recognize the outbranches.

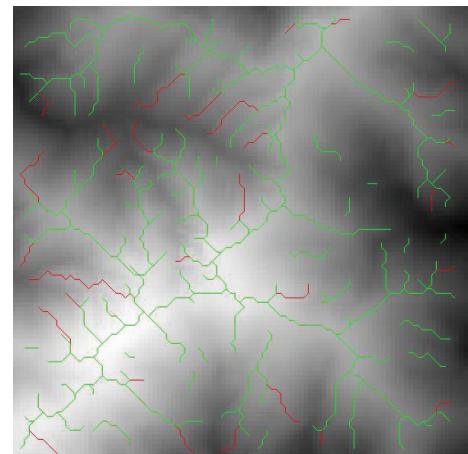


FIGURE 10: Outbranching.

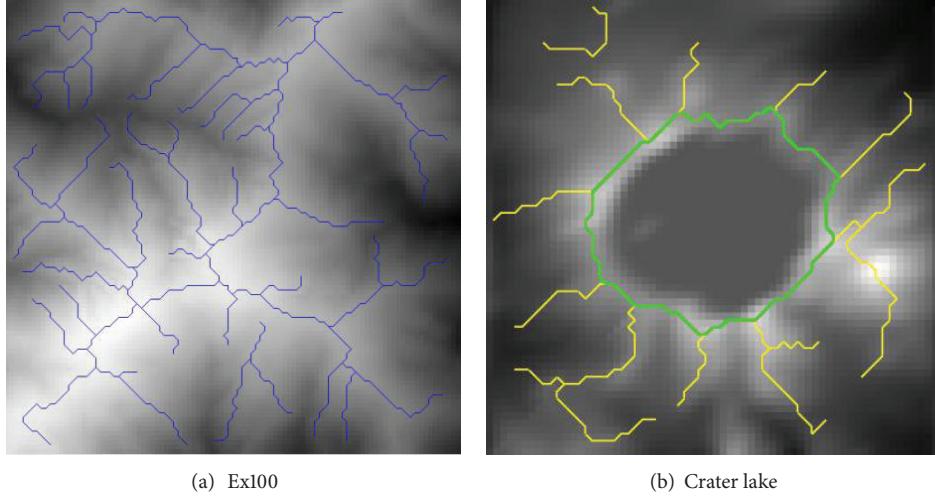


FIGURE 11: Results of skeleton features.

Input: The current segments set seg_{res} , ring segments set seg_{br} , and out branch sign array A_o .
Output: The result segments set seg_{res} with the final skeleton features.
1. Let A_b be branch points array produced in Section 4.1 and let loop = 1 as loop times.
2. Call Algorithm 5 to update the out-branching set A_o of the current segments set seg_{res} .
3. Let $\text{length}_{\text{criterion}}$ be criterion of branch length.
4. Search the unchecked directions of adjacent points of $p \in A_b$, until reach the end points of this branch.
5. For each branch e_b at the beginning of p :
5.1 Let $\text{length}_{\text{criterion}} = \text{length}_{\text{criterion}} + \text{loop}$;
5.2 Calculate the length n of e_b ;
5.3 If $n < \text{length}_{\text{criterion}}$, then eliminate p and the branch e_b ;
5.4 Check direction coefficient of p , if there is the direction that has not been traversed, then go to step 4;
5.5 $\text{loop} = \text{loop} + 1$.
6. Retrieval the stable ring according to ring segments set seg_{br} .
7. Retrieval the out branch according to out-branching sign array A_o .
8. Algorithm end.

ALGORITHM 6: Repeated pruning of MPPA.

Our MPPA is conducted based on GTOPO30, which contains the global elevation data. Its resolution is 30 arc second, that is, approximately 1km. In order to compare MPPA with PPA, we download the benchmark data set ex100 and ex200 that is provided by PPA. There are 10000 points in ex100 and 40000 points in ex200.

Without loss of generality, we select the data L1T7, L1T15, and L1T16 randomly by our another roaming system of

TABLE 1: Contrast of segments between PPA and MPPA.

DEM data	Number of segments		Simplification ratio
	PPA	MPPA	
ex100	8032	1968	75.50%
ex200	41374	11876	71.30%
L1T7	16208	6138	62.13%
L1T15	22928	12405	45.90%
L1T16	23608	8687	63.20%

terrain visualization for GTOPO30. Their longitude and latitude are (97.458, 0.258), (41.458, 33.858), and (99.458, 31.458), respectively. Furthermore, the scale of data is about 16900 points. In addition, we also select G1, G2, and G3 to verify the effect of skeleton features extraction of MPPA. These three blocks of data will be described in Section 5.3.

In addition, Crater Lake has been used as our experimental data, which is of 10 m resolution with 4123×4207 elevation points. In our experiment, Crater Lake is simplified as data set with 128×128 points. The aim is to verify the effect of ring process of MPPA.

5.1. Performance Contrast Analysis. Whether for PPA or MPPA, profile recognition method is utilized to detect the target points. The main advantage is that it can cover all of terrain features because of its super loose judgment condition. Nevertheless, a lot of noise points will come into being.

The advantage of MPPA is that it first simplifies the points to obtain the subset of the points with more precise features, before those target points are connected into segments. However, PPA connected all of the target points to construct the polygon segments. Hence, the number of original segments of MPPA is less than that of PPA greatly. Table 1 described the number contrast of original segments between PPA and MPPA. This means that subsequent algorithms of MPPA are conducted on the little subset of segments, compared with the

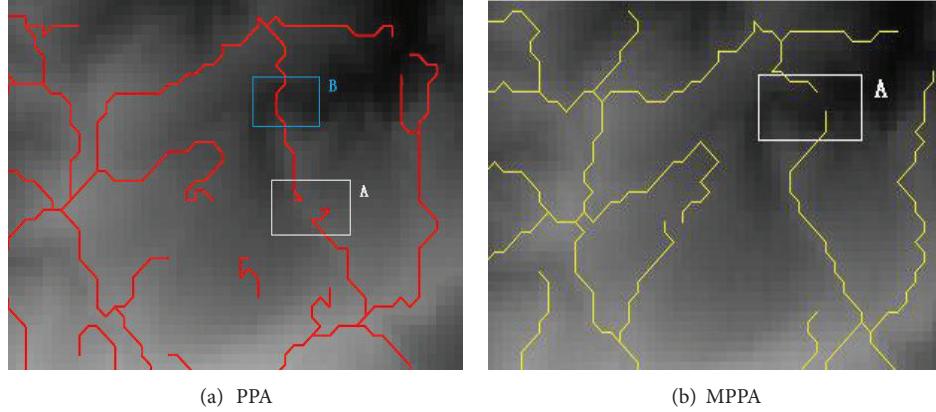


FIGURE 12: Contrast of accuracy (the top right corner of ex200B3).

segments set of PPA. There is reason to believe that MPPA is of relatively high efficiency.

Certainly, the fine efficiency of MPPA is at the price of the time overhead of simplification algorithm. In fact, the simplification algorithm of MPPA replaced with the polygon-breaking algorithm of PPA. Afterwards, we analyze and contrast the two methods.

MPPA proposed the simplification method of noise points based on the morphology. By defining the 48 kinds of morphological coding, the noise points are eliminated largely. From the perspective of simplification algorithm performance, each target point in MPPA need to be traversed only once to judge whether it should be reserved. If the feature morphology composed by the target points and its adjacent points satisfied anyone of the 48 kinds of morphologies, the target point will be eliminated.

Assume that the scale of elevation data is n , and then on the surface, the simplification algorithm of MPPA needs to calculate the $48 \times n$ times. Nevertheless, our method does not need to match the 48 kinds of morphologies but calculate a morphological coding value and directly index the designed morphological signs. The signs have been built in advance by creating a space with 512 bits, equal to 64 bytes. It is really little space overhead. So, the calculating times of the simplification algorithm of MPPA is less than the original data scale n . Obviously, the time complexity of the simplification method is $O(n)$.

Nevertheless, PPA starts with an end point of each segment s to traverse all the branches that connected with it directly or indirectly, until return its end point of s , or there is no path to go. It is noteworthy that the segments set of PPA is so big because it is constructed by fully connecting all of the original target points with a lot of noise points. And a mass of target points have been traversed for many times repeatedly. At worst, the time complexity of the polygon breaking method is $O(n^2)$. Even though the less points need to be traversed with process of the polygon breaking, its average time complexity is approximate $O(n^{3/2})$.

Obviously, our simplification method can eliminate the noise points rapidly. The following chat presents the change of

the noise point number with the increasing of iteration times, as shown in Figure 23.

5.2. Accuracy Analysis and Contrast. Except for the difference mentioned in Section 5.1, the effects of feature extraction by MPPA differ from those by PPA, which can be presented in the following three aspects.

First, since the principle of polygon breaking of PPA is that there is no closed polygon, it is impossible to extract the skeleton feature with rings, such as crater. Nevertheless, our method MPPA is accomplished in ring process. It can recognize the stable rings, reserve them, and at the same time eliminate unstable rings and small rings.

Furthermore, during the procedure of polygon breaking of PPA, it traverses all of the branches blindly and then reaches the other end point or backdates the jump-off point. Obviously, it does not consider any morphological information about skeleton features, which might lead to some error phenomenon. Figure 12 described the renderings contrast between PPA and MPPA. In order to present the visualized effect of feature extraction, we render the terrain model as the form of gray scale image, in which the light area shows the higher elevation, and the dark area shows the lower elevation.

As we can see, some area with lower elevation has been taken as skeleton feature by PPA, shown as the part B denoted by blue rectangle in Figure 12(a). It is not hard to see that the both sides of the part B are bright; therefore, this area is a valley, which should not be contained in the skeleton features. By MPPA, this area has been removed from the skeleton feature segments, shown in Figure 12(b). At the same time, the area A, denoted by white rectangle in Figure 12(a), ought to be reserved; however, PPA breaks the key feature segments.

The last but the most important, for some terrain elevation data, the pruning effect of MPPA is different from that of PPA. Since PPA connected all of points including a lot of noise points into polygon segments, there are so many insignificant little branches, after polygon breaking. Figure 13(a) showed the case of little branches, which might influence the effect of pruning. Whereas Figure 13(b) is the effect of feature segments connected by MPPA after target

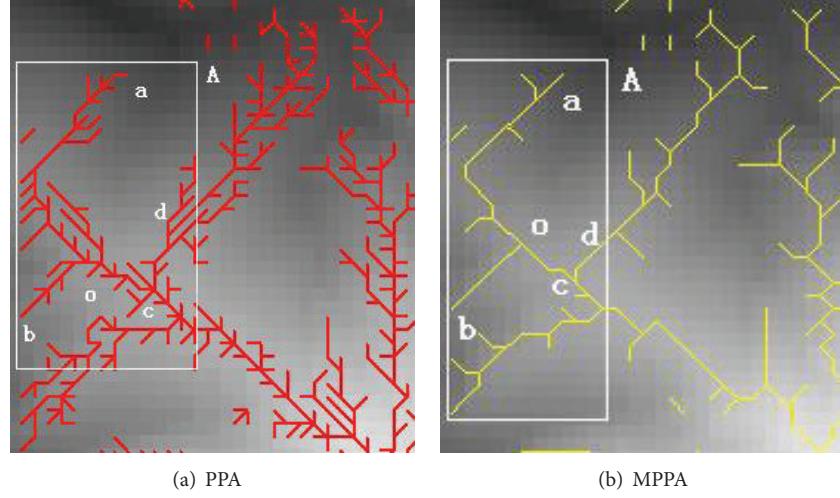


FIGURE 13: Contrast of segments before pruning (the left of ex100).

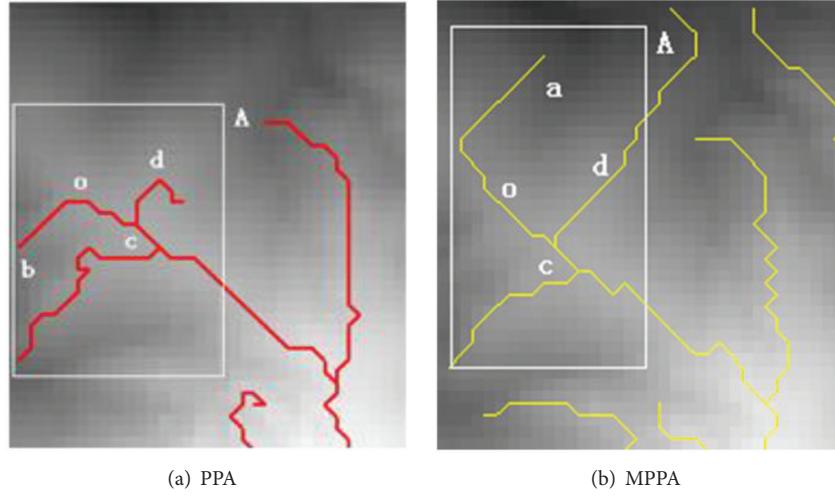


FIGURE 14: Contrast of segments after pruning (the left of ex100).

points simplification. It is not difficult to find that the trunk line extracted by the two methods is consistent, which also verified the validity of MPPA.

It is noteworthy that the feature segments of MPPA have few little branches. It is beneficial to prune accurately. Figure 14(b) showed the effect of MPPA after feature segments have been pruned, which is consistent with the effect of Figure 13(b). Nevertheless, as shown in Figure 14(a), PPA has omitted the trunk lines, which is inconsistent with its results before pruning, shown in Figure 13(a).

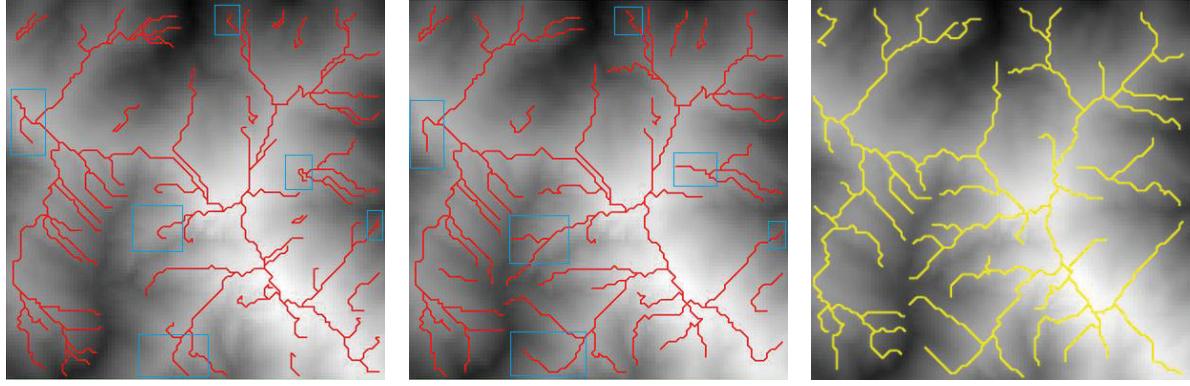
In our experiment, the step length of segment oa is 15, and that of segment ob is 5. Since the short branch should be pruned preferentially, ob ought to be eliminated. However, PPA has removed oa , which lost the accuracy of skeleton features. MPPA reserved oa and then connected it with segment oc ; therefore, a long truck line was formed. Obviously, these skeleton features that extracted by our method are more close to the real terrain features. Compared with PPA, MPPA reserved the long trunk lines and is of more precise.

5.3. Stability of Algorithm and Rendering Results. By executing the program of PPA, we find the various results for the same data. However, the results of MPPA are consistent though the experiment is executed for many times repeatedly. Three real elevation data are adopted to conduct the experiments, including ex100, ex200B0, and ex200B1. The results are shown in Figures 15, 16, and 17, where the two left screenshots are produced by PPA, and the right results are extracted by MPPA.

Apparently, these results from the three data sources demonstrate that many details of feature lines extracted by PPA are different even for the same data, which have been denoted by blue rectangle in Figures 15–17. Hence, MPPA is of good stability, compared with PPA.

In order to further verify the effects of MPPA, we conduct MPPA on many real elevation data, including ex200B2, ex200B3, and other three blocks from GTOPO30.

Figure 18 described the three screenshots, including the original terrain, skeleton features, and longest truck line.

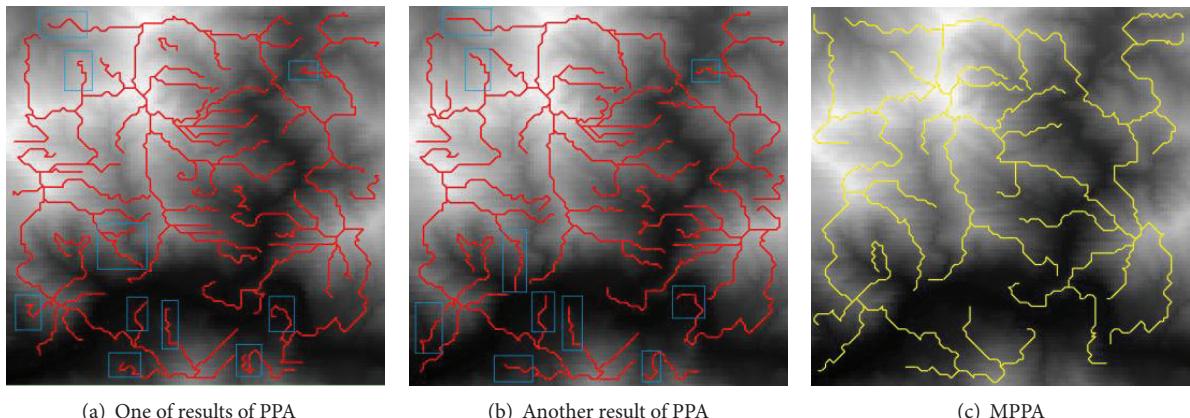


(a) One of results of PPA

(b) Another result of PPA

(c) MPPA

FIGURE 15: Contrast of stability (ex100).

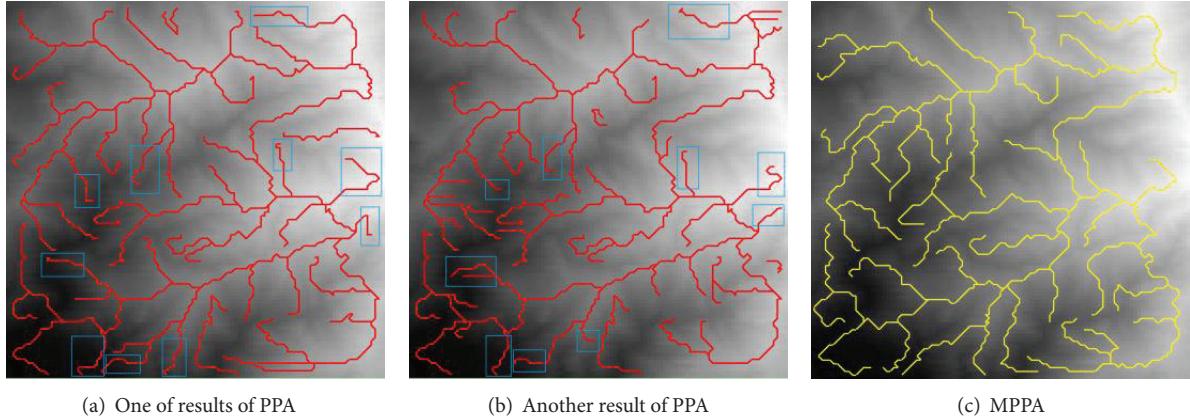


(a) One of results of PPA

(b) Another result of PPA

(c) MPPA

FIGURE 16: Contrast of stability (ex200B0).



(a) One of results of PPA

(b) Another result of PPA

(c) MPPA

FIGURE 17: Contrast of stability (ex200B1).

About 10000 points are contained, and the longest truck feature has been denoted by blue line whose total length of steps is 202. Similarly, the results from data source ex200B3 have been presented in Figure 19. The length of the longest truck line is 210.

Other data sources are from GTOPO30, whose resolution is 1 km and the scale is to contain $93 * 75 = 6975$ points. Their longitude and latitude are about G1(137.502, 69.995), G2(138.278, 70.620), and G3(140.628, 67.495), and their screenshots are shown in Figures 20, 21, and 22. Obviously,

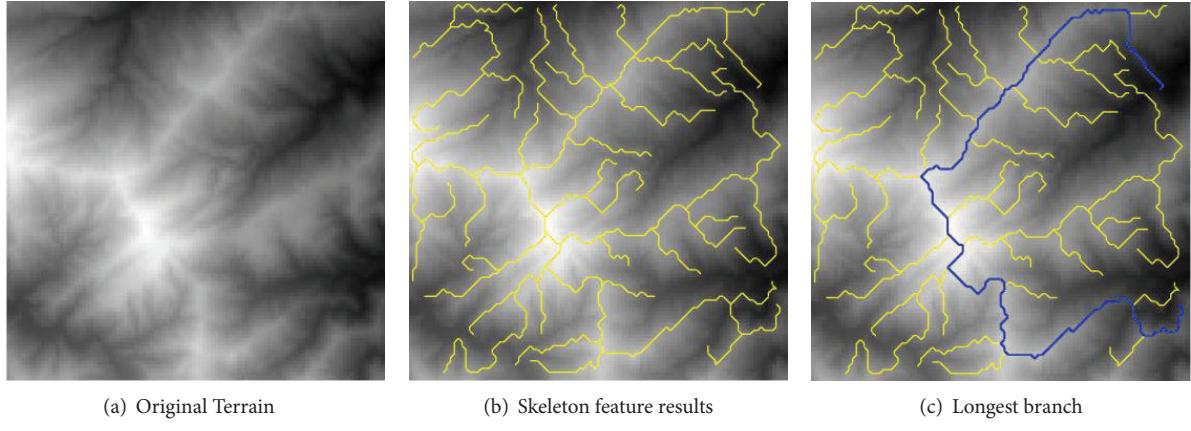


FIGURE 18: Screenshots extracted by MPPA (right below of ex200B2).

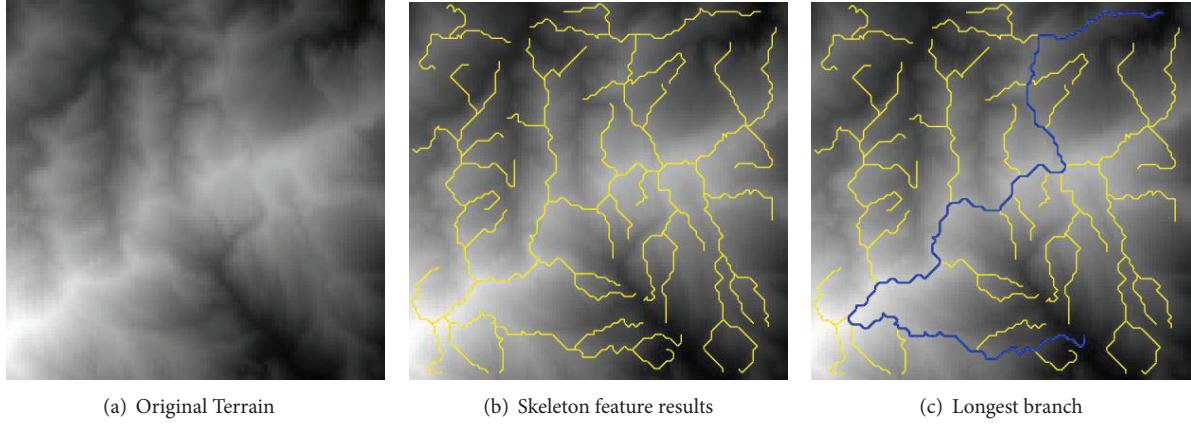


FIGURE 19: Screenshots extracted by MPPA (left below of ex200B3).

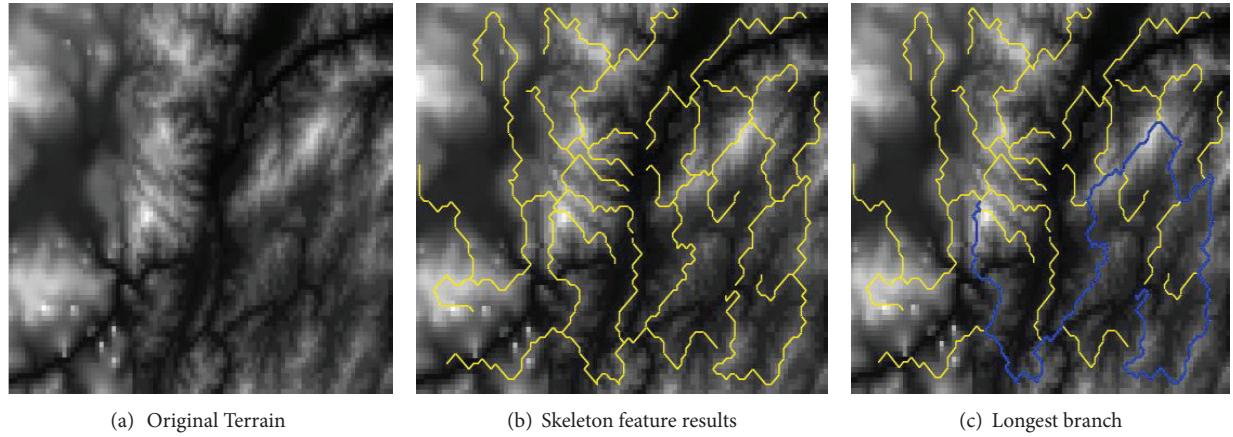


FIGURE 20: Screenshots extracted by MPPA (G1).

the skeleton features with long truck lines are extracted well. The length of the longest branch is 201, 231, and 264, respectively, as shown in Figures 20–22. By the above experiments and analysis for the experimental results, MPPA is of fine performance and accuracy of feature extraction.

6. Conclusions

At present, most features extraction methods of terrain skeleton are not competent to extract the ring features. Furthermore, the trunk line is not long enough to present the real

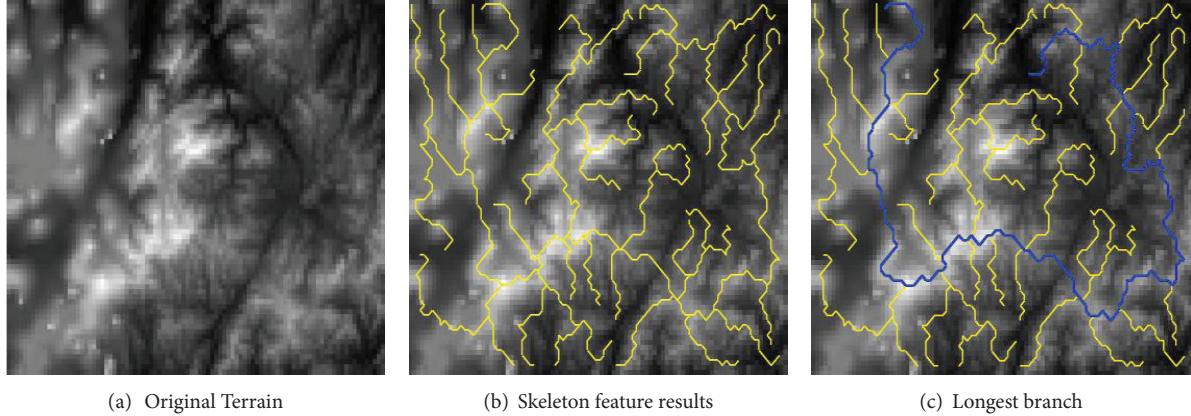


FIGURE 21: Screenshots extracted by MPPA (G2).

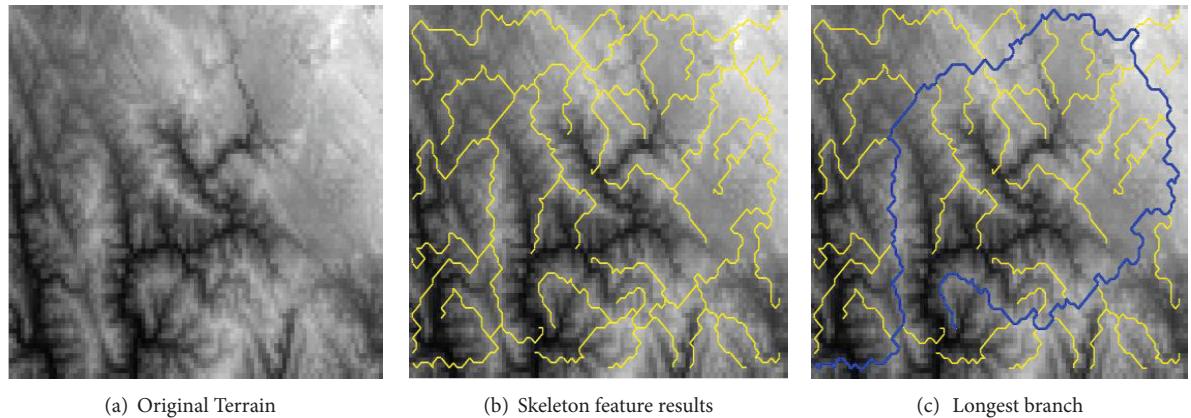


FIGURE 22: Screenshots extracted by MPPA (G3).

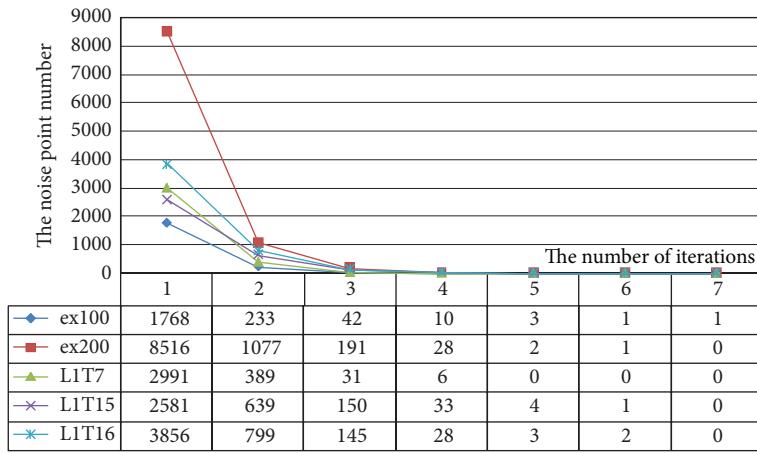


FIGURE 23: Eliminating rate of noise points of MPPA.

terrain. Moreover, the accuracy of skeleton features for some terrain is not very high because of the negative effect of noisy points. Aimed at the above problems, a new method of skeleton feature extraction is proposed to eliminate the noise points, to reserve the truck line, and to recognize and remain

the stable ring, which is significant in terrain modeling engineering.

In this paper, we defined the candidate feature points and proposed a morphological simplification algorithm to eliminate the noise points rapidly, which is benefit to the

subsequent ring process and pruning. The feature point subset that has been simplified contains less data, compared with the original target points; therefore, the efficiency of ring process and pruning algorithms has been improved. Moreover, the accuracy of extraction has also been enhanced without the negative effect of noise points.

In addition, a concept of filling intensity is defined to determine a big stable ring. For small and unstable rings, we proposed the disassembling algorithm. In order to keep the long truck line, a concept of outbranch is defined, and the pruning algorithm is proposed to reserve the outbranch feature lines and eliminate the little branches.

Finally, all algorithms have been tested on many real elevation data and benchmark data. The experimental results show that our MPPA outweighs PPA in the aspects of performance and accuracy. Moreover, we proposed the strategy of ring process and customized pruning rules.

In the future, we will extract the skeleton features on the huge set of terrain elevation and expand these theories and algorithms of MPPA in the field of 3D visualization, such as 3D terrain synthesis and terrain deformation.

Acknowledgments

The authors are grateful for the helpful comments and suggestions of the reviewers, which have improved the presentation. This work is supported by Research Fund for the Doctoral Program of Higher Education of China (no. 20100043120012) and by National Natural Science Foundation of China for Young Scholars (no. 41101434).

References

- [1] D. Bavera, C. de Michele, and M. Pepe, “Melted snow volume control in the snowmelt runoff model using a snow water equivalent statistically based model,” *Hydrological Processes*, vol. 26, no. 22, pp. 3405–3415, 2012.
- [2] I. C. Fuller and M. Marden, “Slope-channel coupling in steep-land terrain: a field-based conceptual model from the Tarndale gully and fan, Waipaoa catchment, New Zealand,” *Geomorphology*, vol. 128, no. 3–4, pp. 105–115, 2011.
- [3] M. Cavalli, S. Trevisani, F. Comiti, and L. Marchi, “Geomorphometric assessment of spatial sediment connectivity in small Alpine catchments,” *Geomorphology*, vol. 188, pp. 31–41, 2013.
- [4] T. K. Peucker and D. H. Douglas, “Detection of surface-specific points by local parallel processing of discrete terrain elevation data,” *Computer Graphics and Image Processing*, vol. 4, no. 4, pp. 375–387, 1975.
- [5] J. Wood, *The Geomorphological Characterisation of Digital Elevation Models*, University of Leicester, Leicester, UK, 1996.
- [6] L. E. Band, “Topographic partition of watersheds with digital elevation models,” *Water Resources Research*, vol. 22, no. 1, pp. 15–24, 1986.
- [7] J. F. O’Callaghan and D. M. Mark, “The extraction of drainage networks from digital elevation data,” *Computer Vision, Graphics, & Image Processing*, vol. 28, no. 3, pp. 323–344, 1984.
- [8] J. Fairfield and P. Leymarie, “Drainage networks from grid digital elevation models,” *Water Resources Research*, vol. 27, no. 5, pp. 709–717, 1991.
- [9] M. C. Costa-Cabral and S. J. Burges, “Digital elevation model networks (DEMON): a model of flow over hillslopes for computation of contributing and dispersal areas,” *Water Resources Research*, vol. 30, no. 6, pp. 1681–1692, 1994.
- [10] D. G. Tarboton, “A new method for the determination of flow directions and upslope areas in grid digital elevation models,” *Water Resources Research*, vol. 33, no. 2, pp. 309–319, 1997.
- [11] A. Meisels, S. Raizman, and A. Karnieli, “Skeletonizing a DEM into a drainage network,” *Computers and Geosciences*, vol. 21, no. 1, pp. 187–196, 1995.
- [12] R. Moussa, M. Voltz, and P. Andrieux, “Effects of the spatial organization of agricultural management on the hydrological behaviour of a farmed catchment during flood events,” *Hydrological Processes*, vol. 16, no. 2, pp. 393–412, 2002.
- [13] C. Gascuel-Odoux, P. Auroousseau, T. Doray et al., “Incorporating landscape features to obtain an object-oriented landscape drainage network representing the connectivity of surface flow pathways over rural catchments,” *Hydrological Processes*, vol. 25, no. 23, pp. 3625–3636, 2011.
- [14] F. Cazorzi, G. D. Fontana, A. D. Luca, G. Sofia, and P. Tarolli, “Drainage network detection and assessment of network storage capacity in agrarian landscape,” *Hydrological Processes*, vol. 27, no. 4, pp. 541–553, 2013.
- [15] R. Barnes, C. Lehman, and D. Mulla, “An efficient assignment of drainage direction over flat surfaces in raster digital elevation models,” *Computers & Geosciences*, 2013.
- [16] F. Nardi, S. Grimaldi, and M. Santini, “Hydrogeomorphic properties of simulated drainage patterns using digital elevation models: the flat area issue/Propriétés hydrogéomorphologiques de réseaux de drainage simulés à partir de modèles numériques de terrain: la question des zones planes,” *Hydrological Sciences Journal*, vol. 53, no. 6, pp. 1176–1193, 2008.
- [17] R. Jones, “Algorithms for using a DEM for mapping catchment areas of stream sediment samples,” *Computers and Geosciences*, vol. 28, no. 9, pp. 1051–1060, 2002.
- [18] R. Jana, T. V. Reshma Devi, P. S. Arun, and T. I. Eldho, “An enhanced technique in construction of the discrete drainage network from low-resolution spatial database,” *Computers and Geosciences*, vol. 33, no. 6, pp. 717–727, 2007.
- [19] H. Zhang and G. Huang, “Building channel networks for flat regions in digital elevation models,” *Hydrological Processes*, vol. 23, no. 20, pp. 2879–2887, 2009.
- [20] Y. Chang, G. Song, and S. Hsu, “Automatic extraction of ridge and valley axes using the profile recognition and polygon-breaking algorithm,” *Computers and Geosciences*, vol. 24, no. 1, pp. 83–93, 1998.
- [21] H. Zhou, J. Sun, G. Turk, and J. M. Rehg, “Terrain synthesis from digital elevation models,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 4, pp. 834–848, 2007.
- [22] H. Hnaidi, E. Guérin, S. Akkouche, A. Peytavie, and E. Galin, “Feature based terrain generation using diffusion equation,” *Computer Graphics Forum*, vol. 29, no. 7, pp. 2179–2186, 2010.
- [23] O. Mora, J. J. Mallorqui, and A. Broquetas, “Linear and non-linear terrain deformation maps from a reduced set of interferometric SAR images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 10, pp. 2243–2253, 2003.
- [24] J. Crause, A. Flower, and P. Marais, “A system for real-time deformable terrain,” in *Proceedings of the 2011 ACM the South African Institute of Computer Scientists and Information Technologists Conference on Knowledge, Innovation and Leadership in a Diverse, Multidisciplinary Environment*, pp. 77–86, October 2011.

- [25] R. Westerteger, T. Compton, and T. Bemadin, “Interactive retro-deformation of terrain for reconstructing 3D fault displacements,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2208–2215, 2012.
- [26] R. Xu, F. Li, and L. Liao, “A 3D non-regular military symbol deformation method along the battlefield terrain,” in *Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT ’10)*, pp. 207–211, July 2010.
- [27] R. Kooima, J. Leigh, A. Johnson, D. Roberts, M. Subbarao, and T. A. Defanti, “Planetary-scale terrain composition,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 5, pp. 719–733, 2009.
- [28] O. Šťava, B. Beneš, M. Brisbin, and J. Křivánek, “Interactive terrain modeling using hydraulic erosion,” in *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 201–210, 2008.
- [29] J. Gain, P. Marais, and W. Straßer, “Terrain sketching,” in *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games*, pp. 31–38, March 2009.
- [30] F. P. Tasse, J. Gain, and P. Marais, “Enhanced texture-based terrain synthesis on graphics Hardware,” *Computer Graphics Forum*, vol. 31, no. 6, pp. 1959–1972, 2012.
- [31] X. Li, J. Wang, J. Zhou, and M. Yin, “A perturb biogeography based optimization with mutation for global numerical optimization,” *Applied Mathematics and Computation*, vol. 218, no. 2, pp. 598–609, 2011.
- [32] X. Li and M. Yin, “Application of differential evolution algorithm on self-potential data,” *PloS ONE*, vol. 7, no. 12, Article ID e51199, 2012.
- [33] X. Li and M. Yin, “An opposition-based differential evolution Algorithm for permutation flow shop scheduling based on diversity measure,” *Advances in Engineering Software*, vol. 55, pp. 10–31, 2013.

Research Article

Behavior-Based Fuzzy Control for Mobile Robot Navigation

Hongwei Mo,¹ Qirong Tang,² and Longlong Meng¹

¹ Automation College, Harbin Engineering University, No. 145 Nantong Street, Nangang District, Harbin 150001, China

² University of Stuttgart, Pfaffenwaldring 9, 70569 Stuttgart, Germany

Correspondence should be addressed to Hongwei Mo; honwei2004@126.com

Received 10 July 2013; Accepted 13 August 2013

Academic Editor: William Guo

Copyright © 2013 Hongwei Mo 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 behavior-based fuzzy control method for mobile robot navigation is presented. It is based on behavioral architecture which can deal with uncertainties in unknown environments and has the ability to accommodate different behaviors. Basic behaviors are controlled by specific fuzzy logic controllers, respectively. The proposed approach qualifies for driving a robot to reach a target while avoiding obstacles in the environment. Simulation and experiments are performed to verify the correctness and feasibility of the proposed method.

1. Introduction

One way to accomplish robot navigation is using behavior arbitration and behavior control [1, 2]. Since the behavior control architecture was proposed in 1986, it has been adopted to solve the robot navigation problem frequently. The behavior control is a special form of decentralized switching control in which each behavior is autonomous and can drive the robot on its own without dependencies on other behaviors. Under the standard behavior paradigm, each behavior triggers its own specific control command according to the behavior. The behavior control architecture can handle the navigation problem in an online manner and it does not require environment model; thus, it has been widely used for mobile robot navigation [3–5]. Fuzzy logic control (FLC) [6] is the most important method for robot behavior-based control and has been investigated for mobile robot navigation and obstacle avoidance by many researchers, see [7–14].

In [9], Selekwa et al. presented the design of a preference-based fuzzy behavior system for navigating robotic vehicles using the multivalued logic framework. The proposed method allows the robot to smoothly and effectively navigate through cluttered environments. In [11], a new fuzzy logic algorithm is developed for mobile robot navigation in local environments. A robot perceives its environment through sensors, while the fuzzy logic algorithm

performs the main tasks of obstacle avoidance and target seeking. In [13], Wang et al. proposed behavior-based hierarchical fuzzy control method for mobile robot navigation in dynamic environment.

Fuzzy logic controllers can be used for driving mobile robot to perform specific motions with good robustness. However, a fuzzy mobile robot controller should be self-adaptive since it must deal with very different control situations such as following a wall or a corridor, or avoiding an obstacle. Rules must be designed so that they can handle these different situations. A way of solving this problem is to create a fundamental or basic controller which can be incrementally updated and optimized by tuning both labels and rules adequately. Another way of solving this problem is to decompose the complex behavior into several subbehaviors or motions which are controlled by separate schema. These control schemas can be merged by combining the corresponding actions. Within each control schema, it is possible to define the most appropriate semantics for the linguistic variables which are manipulated in the rules, and thus each behavior can be tuned independently to be more effective in its own context. Complex behaviors can be obtained based upon the composition of simpler behaviors.

This paper will address the mobile robot navigation issue by combining the ideas of behavior-based control and fuzzy logic control. The rest of this paper is organized as follows. Section 2 presents the framework of behavior-based

robot navigation. Section 3 gives the robot kinematic model. Section 4 presents the detailed structure of the proposed control method which contains three elementary behaviors. Simulation results that demonstrate the performance of the proposed approach are given in Section 5. Experiments are carried out in Section 6 while concluding remarks are given in Section 7 to close this paper.

2. Framework of Behavior-Based Robot Navigation

This section proposes a framework for a behavior-based navigation strategy for a mobile robot in an unknown environment. The general structure of a fuzzy behavior system consists of some independent fuzzy behaviors and a component of command fusion, see Figure 1. The framework totally includes four components, which are preprocessing, goal determination, behavior arbitration, and command fusion. In each robot control cycle, the robot reasoning system provides a set of motor control commands via an inference process I. This inference process can be considered as a relationship between the input U and the output Y . The input U is represented by a multidimensional vector corresponding to a particular set of input data, for example, distance to obstacle, direction to the goal. Similarly, the output Y corresponds to motor speed and steering angle, and so on. Thus, the relation between Y and U can be expressed by

$$I : U(U_1, U_2, \dots, U_n) \rightarrow Y(Y_1, Y_2, \dots, Y_m). \quad (1)$$

2.1. Preprocessing. Input data from the robot sensors is performed with a simple preprocessing to reduce noises. Specifically, the sonar and infrared sensor measurements are preprocessed. Meanwhile, in the preprocessing module, if the input U is too large, the computational complexity also will be reduced. The dimensions of input U can be reduced by introducing a limited number of intermediate variables. These variables classify different perceptual situations which are relevant to the robot's current behavior and status. Some intermediate variables are statements such as front-obstacle distance, right-obstacle distance, or left-obstacle distance. These variables are also used for afterwards behavior design.

2.2. Goal Determination. In this investigation, we classify the goal determination to two types which are a determined goal and a undetermined goal. When the goal is determined, its exact coordinates are given. The robot must move to a given target while autonomously avoids the blocking obstacles on the robot path. When the goal is undetermined, it does not have the exact coordinates of the goal.

2.3. Behaviors and Fuzzy Logic Controllers. In general, in our case there are two types of robot behaviors, which are goal seeking (GS) and obstacle avoidance (OA). In order to realize the GS behavior, a controller is designed to generate motor commands to reach the goal as soon as possible. The OA behavior is actually sensor-based behavior, which implements a control strategy based on external sensing.

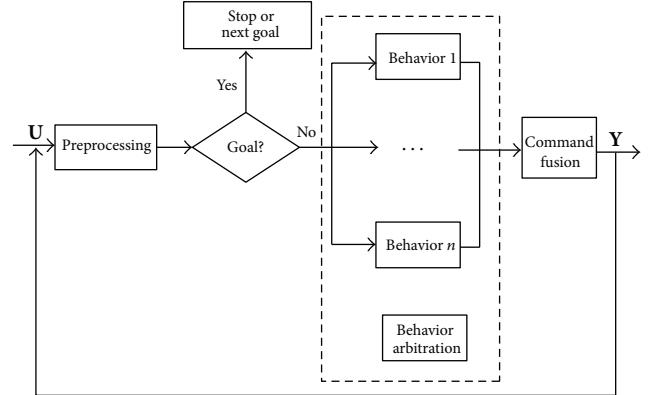


FIGURE 1: Framework of behavior-based robot navigation.

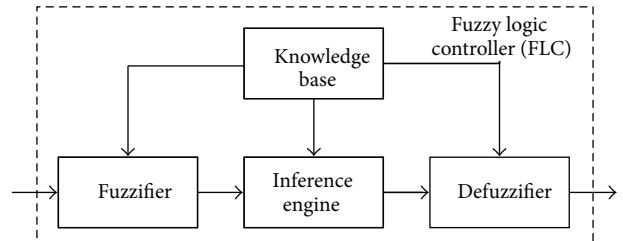


FIGURE 2: Architecture of a basic fuzzy logic controller.

As we designed, each behavior will be controlled by a specific fuzzy controller. In those controllers, reasoning is contained in the rules operating on linguistic inputs and outputs, such as

$$\text{If } u_1 \text{ is } A_1 \text{ and } u_2 \text{ is } A_2 \text{ and } \dots \text{ and } u_n \text{ is } A_n, \quad (2)$$

$$\text{Then } y_1 \text{ is } B_1 \text{ and } y_2 \text{ is } B_2 \text{ and } \dots \text{ and } y_m \text{ is } B_m,$$

where u_i is the input linguistic variable taking linguistic value A_i . Each linguistic value is defined by a membership function $\mu_{A_i}(u_i)$. The y_i is the output linguistic variable taking linguistic value B_i . Each linguistic value of output is defined by a membership function $\mu_{B_i}(y_i)$.

Given two linguistic values A and B defined on the same universe of discourse, the AND and OR operations are defined by (3) and (4), respectively, which are

$$\mu_{A \cap B} = \min_{u \in U} (\mu_A(u), \mu_B(u)), \quad (3)$$

$$\mu_{A \cup B} = \max_{u \in U} (\mu_A(u), \mu_B(u)). \quad (4)$$

Figure 2 shows the principle of a basic fuzzy logic controller. In this investigation, we use a Takagi-Sugeno-Kang model as the fuzzy inference engine and the Centroid method for defuzzification.

The fuzzy inference engine is defined as in Box 1.

2.4. Behavior Arbitration (Coordination). Behavior arbitration can be taken as two conceptually different problems. The first one is the behaviors of OA and GS. In our case,

```

Input:  $u_i$ —crisp numerical value.
Output:  $y_i$ —crisp numerical value.
BEGIN:
Step 1. Fuzzification of the input  $u_i$ ,
Step 2. Application of the fuzzy operators (AND or OR)
in the antecedent of the rules according to (3) and (4),
Step 3. Implication from the antecedent to the consequent
using of the AND operation according to (3),
Step 4. Aggregation of the consequents across
the rules using the OR operation according to (4),
Step 5. Defuzzification to output variable  $y_i$ .
END

```

Box 1

they are activated during the whole motion process. The second problem is combining the results from different behaviors into one command sent to the robot's actuators. These problems are expressed in Figure 3.

One strategy for behavior arbitration is context-dependent blending (CDB) in which fuzzy logic is applied so that a decision between behaviors can be made in a prevailing situation. Our behavior arbitration strategy is similar to the CDB method. It uses fuzzy context rules to express the behavior arbitration strategy. When the obstacle is close, both OA and GS behaviors are activated and each behavior is assigned a weighting factor. These factors are adjusted dynamically according to the fuzzy weight rules. The weighting factors determine the degree of influence of each behavior on the final motion command. The weight rules continuously update the behavior weighting factors during robot motions.

3. Kinematic Model of Mobile Robot

In this investigation, a differentially driven mobile robot is used, see its kinematic illustration in Figure 4.

The robot has two drive wheels mounted on the same axis. Kinematic equations of this two-wheeled mobile robot are

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta}_m \end{bmatrix} = \begin{bmatrix} \cos \theta_m & 0 \\ \sin \theta_m & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (5)$$

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2}r & \frac{1}{2}r \\ \frac{r}{D} & \frac{-r}{D} \end{bmatrix} \cdot \begin{bmatrix} V_R \\ V_L \end{bmatrix}, \quad (6)$$

where x and y are coordinates of the mass center of the mobile robot, θ_m is the angle that represents the current orientation of the robot, v and ω are linear and angular velocities of the robot, V_R and V_L are angular velocities of right and left wheels, r is the wheels radius, and D is the distance between the two wheels centers.

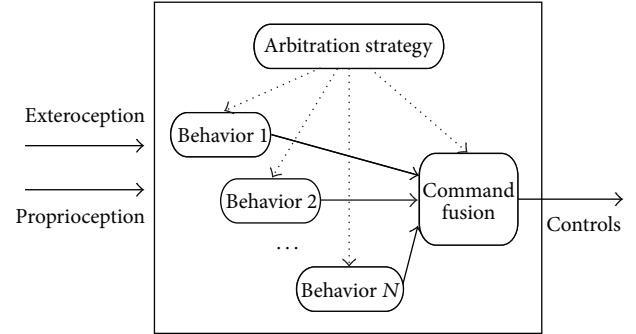


FIGURE 3: Behavior arbitration.

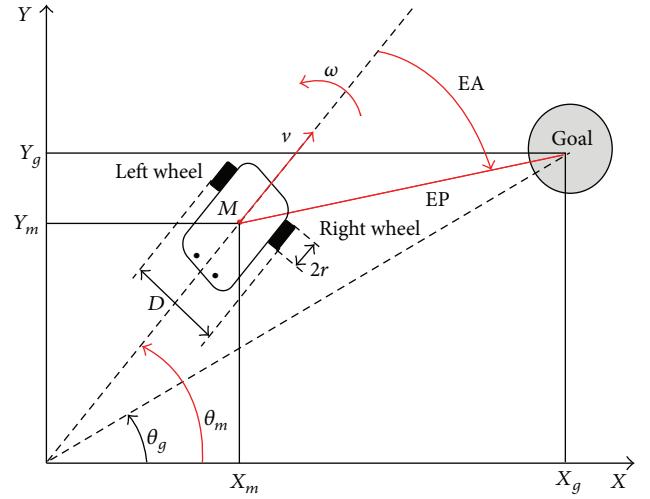


FIGURE 4: The kinematic illustration of the used mobile robot.

Combining (5) and (6) yields

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta}_m \end{bmatrix} = \begin{bmatrix} \frac{1}{2}r \cos \theta_m & \frac{1}{2}r \cos \theta_m \\ \frac{1}{2}r \sin \theta_m & \frac{1}{2}r \sin \theta_m \\ \frac{r}{D} & \frac{-r}{D} \end{bmatrix} \cdot \begin{bmatrix} V_R \\ V_L \end{bmatrix}. \quad (7)$$

Equation (7) is the kinematic model of the used robot for both simulation and experiments. Right wheel angular velocity V_R and left wheel angular velocity V_L are used as motion commands to the motors for realizing different behaviors.

4. Behavior-Based Fuzzy Control for Mobile Robot Navigation

We decompose the task of robot navigation into three elementary behaviors: goal seeking (GS), obstacle avoidance (OA), and behavior fusion (BF).

4.1. Global Goal-Seeking Behavior. The GS behavior discussed here is a kind of global behavior which does not rely on

external sensed data but seeks for the global and exact goal. The inputs of the fuzzy logic controller include distance from the robot to the goal (EP) and the angular between robot orientation and goal orientation (EA). They are shown in Figure 4. The expressions of these two inputs are

$$\begin{aligned} EP &= \sqrt{D_x^2 + D_y^2}, \\ EA &= \theta_m - \theta, \\ D_x &= x_g - x, \\ D_y &= y_g - y, \\ \theta &= \arctan(D_y, D_x), \end{aligned} \quad (8)$$

where (x, y, θ_m) is the robot position and orientation and (x_g, y_g, θ_g) is the position and orientation of the goal.

The position deviation is represented by a five-variable linguistic fuzzy set $\{\text{ZE}, \text{S}, \text{M}, \text{B}, \text{VB}\}$ with its entries for distance of zero, small, medium, big, and very big, respectively. Their corresponding membership functions are shown in Figure 5(a). Similarly, the angular error is represented by $\{\text{NB}, \text{NM}, \text{NS}, \text{ZE}, \text{PS}, \text{PM}, \text{PB}\}$ with linguistic members to indicate negative big, negative medium, negative small, zero, positive small, positive medium, and positive big, respectively, see their membership functions in Figure 5(b). The positive and negative imply that the robot turns to the left and right, respectively.

The motion control variables of the mobile robot are the angular velocities of the right and left wheels. Similarly, the two velocities are represented by a seven linguistic variables' fuzzy set $\{\text{NB}, \text{NM}, \text{NS}, \text{ZE}, \text{PS}, \text{PM}, \text{PB}\}$ with their membership functions shown in Figure 5(c).

The rule base of the GS behavior is summarized in Table 1. For instance, the (1, 1) entry in Table 1 can be written as

$$\text{If EP is ZE and EA is NB, then } V_L \text{ is NM and } V_R \text{ is PM.} \quad (9)$$

We can use this behavior-based fuzzy controller alone in an environment without obstacles. However, usually the environment contains obstacles.

4.2. The Obstacle Avoidance Behavior. Obstacle avoidance is actually a sensor-based behavior which implements a control strategy based on external sensing. We reduce the dimensions of inputs by grouping the robot's sonar reading into three options which are left, front, and right. For example, our robot has 24 ultrasonic sonars that produce a set of obstacle distances by the following equations:

$$\begin{aligned} d_{\text{right}} &= \min(d_2, d_3, d_4, d_5), \\ d_{\text{front}} &= \min(d_0, d_1, d_{21}, d_{22}, d_{23}), \\ d_{\text{left}} &= \min(d_{17}, d_{18}, d_{19}, d_{20}). \end{aligned} \quad (10)$$

This is due to the layout of the ultrasonic sensors. The obstacle distance of each option is represented by three-member fuzzy set $\{\text{VERYNEAR}, \text{NEAR}, \text{FAR}\}$ with their

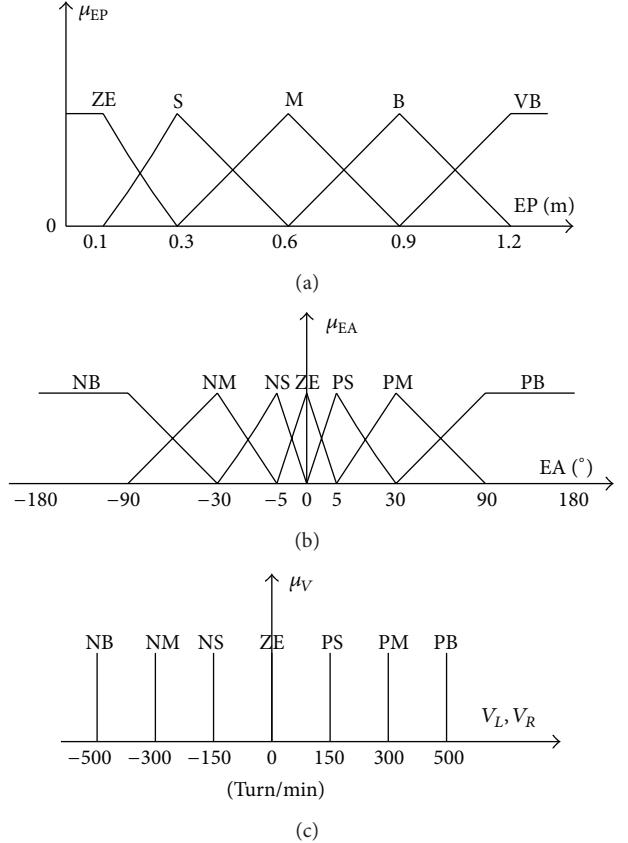


FIGURE 5: Membership functions (a) for position deviation (b) for angular deviation (c) for angular velocities of right and left wheels.

TABLE 1: Rule base of the GS behavior.

		Angular error (EA)						
		NB	NM	NS	ZE	PS	PM	PB
Position error (EP)	ZE							
	V _L	NM	NS	ZE	ZE	ZE	PS	PM
S	V _R	PM	PS	ZE	ZE	ZE	NS	NM
	V _L	NB	NB	NM	PS	PS	PM	PM
M	V _R	PM	PM	PS	PS	NM	NB	NB
	V _L	NM	NM	NB	PM	PM	PS	PS
B	V _R	PS	PS	PM	PM	NB	NM	NM
	V _L	PS	PS	PM	PB	PB	PM	PB
VB	V _R	PB	PM	PB	PB	PM	PS	PS
	V _L	PS	PS	PM	PB	PB	PM	PM
	V _R	PM	PM	PB	PB	PM	PS	PS

membership functions shown in Figure 6(a) while behavior weight member functions are shown in Figure 6(b).

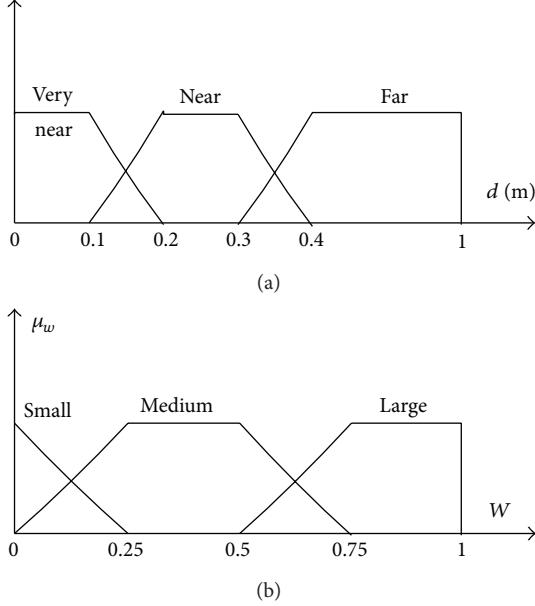


FIGURE 6: Membership functions (a) for distance and (b) for behavior weight.

The velocity variables V_R and V_L (here for obstacle avoidance) are represented by fuzzy set {NF, NM, PM, PF} which means negative fast, negative medium, positive medium, and positive fast. The velocity rules of left and right wheels for the OA behavior are summarized in Figures 7 and 8, respectively. The rules exhibit a behavior characteristic that, if the obstacle distance in any situation is very near, the robot should turn away to find a safer direction.

For instance, the two (3, 3) elements of the top layers in Figures 7 and 8 have the rules as

$$\begin{aligned} & \text{IF } d_{\text{front}} \text{ is very near AND } d_{\text{left}} \text{ is far AND} \\ & d_{\text{right}} \text{ is far, THEN } V_L \text{ is NM;} \\ & \text{IF } d_{\text{front}} \text{ is very near AND } d_{\text{left}} \text{ is far AND} \\ & d_{\text{right}} \text{ is far, THEN } V_R \text{ is PM,} \end{aligned} \quad (11)$$

respectively.

When the three parts of the robot body corresponding to (10) have similar obstacle distances as shown in the (1, 1) element of the two top layers in Figures 7 and 8, the robot has to escape from its current embarrassed situation, so V_L is NF and V_R is PF which let the robot make a large left turn angle and decrease its speed since

$$\omega = \frac{r}{D} (V_R - V_L), \quad v = \frac{r}{2} (V_R + V_L). \quad (12)$$

4.3. Behavior Fusion. The behavior fusion is based on the weight assigned for each of the behaviors. The weight of OA behavior τ is represented by three-member linguistic fuzzy set {SMALL, MEDIUM, LARGE} with the membership

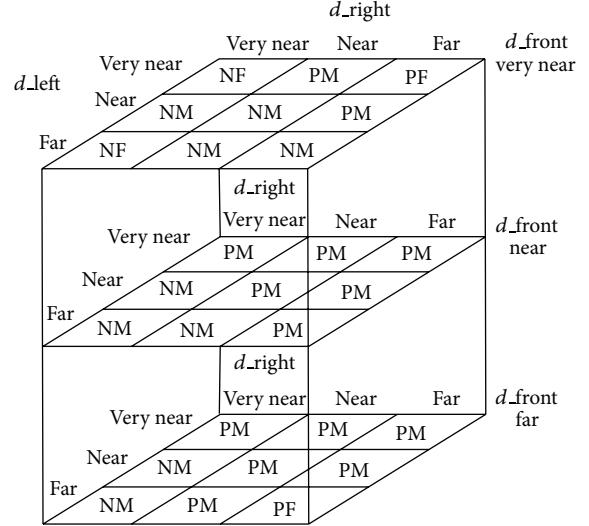


FIGURE 7: The V_L rule base for OA behavior.

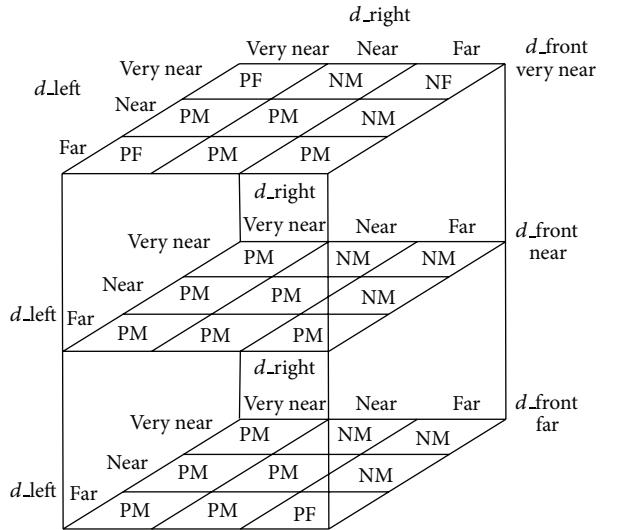


FIGURE 8: The V_R rule base for OA behavior.

functions shown in Figure 6(b) and rules in Figure 9. At first, we use a behavior arbitration module to calculate the defuzzified weight factors of all behaviors and then carry out command fusion by using these weight factors via

$$\begin{aligned} V_R &= \frac{\sum w_i V_{R_i}}{\sum w_i}, \\ V_L &= \frac{\sum w_i V_{L_i}}{\sum w_i}, \end{aligned} \quad (13)$$

where V_R and V_L are angular velocities of right and left wheels as motion commands, while V_{R_i} and V_{L_i} are angular velocities of right and left wheels preference values suggested by each specific behavior.

Here, w_i is the defuzzified weight factor. The implementation of the behavior fusion is depicted in Figure 10.

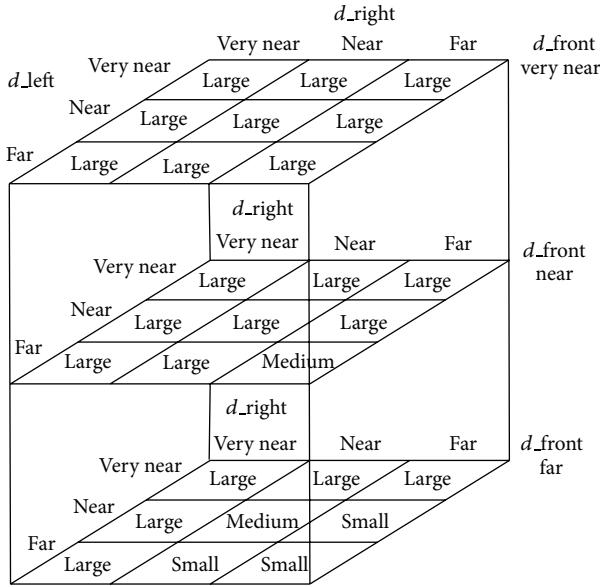


FIGURE 9: Weight rules for OA behavior.

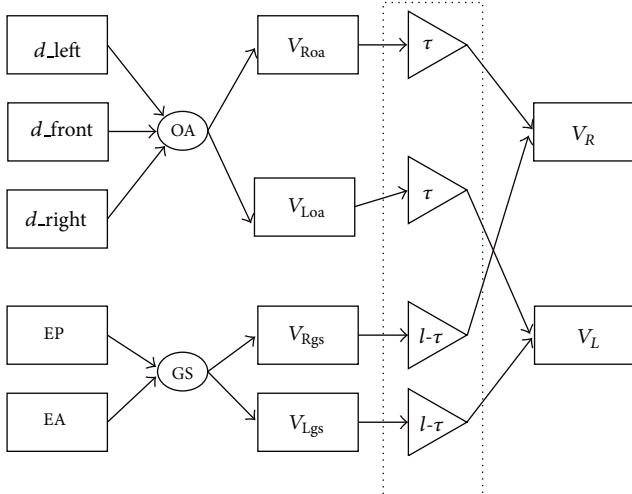


FIGURE 10: Diagram of behavior fusion.

As shown in the figure, the motion control variables (angular velocities of right and left wheels) are inferred by the GS and OA behaviors. And they are weighted by $1 - \tau$ and τ , respectively. The results can be given by

$$\begin{aligned} V_R &= (1 - \tau) V_{Rgs} + \tau V_{Roa}, \\ V_L &= (1 - \tau) V_{Lgs} + \tau V_{Loa}. \end{aligned} \quad (14)$$

For instance, the rule of (3, 3) element of the top layer in Figure 9 can be written as

$$\text{IF } d_{\text{front}} \text{ is very near AND } d_{\text{left}} \text{ is far AND } d_{\text{right}} \text{ is far, THEN } \tau \text{ is large.} \quad (15)$$

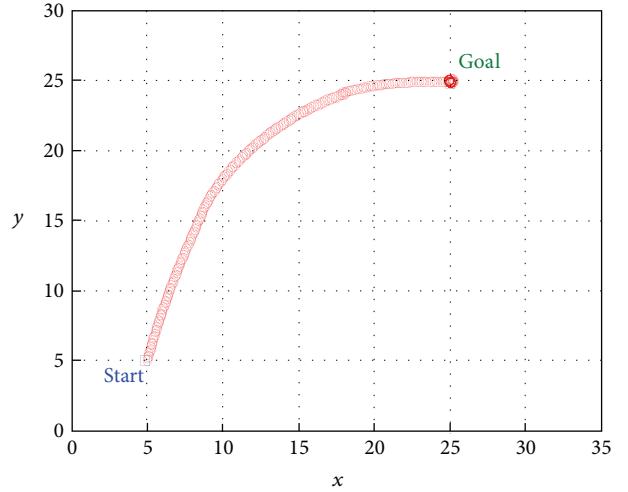
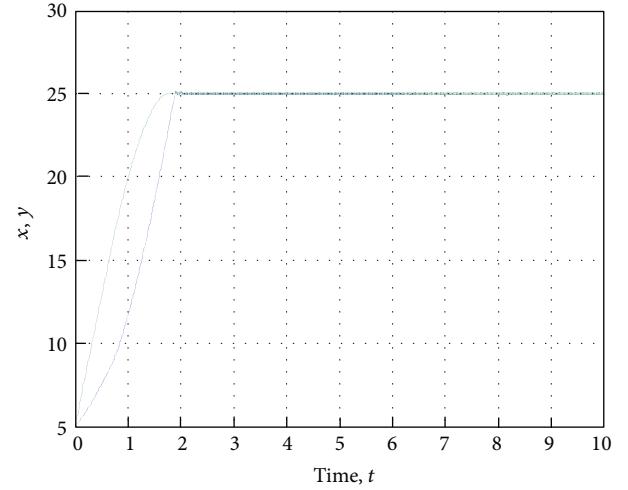


FIGURE 11: Robot's path in the environment without obstacles.

FIGURE 12: Robot's separated trajectories in the environment without obstacles (blue solid— x direction, green dashed— y direction).

5. Simulation

In order to verify the proposed method, simulation has been carried out within MATLAB. We designed the different fuzzy logic controllers based on different behaviors via fuzzy inference system (FIS).

5.1. Simulation in the Environment without Obstacles. In this case, the robot is supposed to move from the start point $(x_m, y_m) = (5, 5)$ to the goal $(x_g, y_g) = (25, 25)$. Note that the initial orientation of the robot is $\theta_m = \pi/2$ and there are no obstacles in the environment.

As illustrated in Figure 11, the robot reaches the goal. The trajectory is indicated by the chain of red circles. The robot's x and y directions separated trajectories are shown in Figure 12.

The weight of each behavior represents its extent of influence on the final motion commands. In this simulation, the proposed control system uses the GS behavior, because

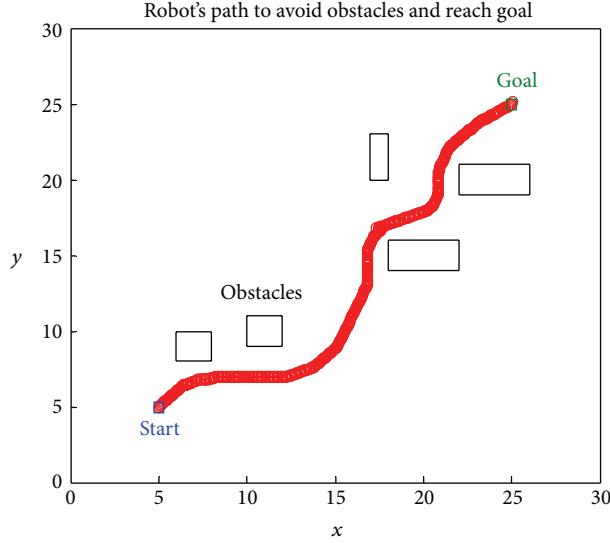


FIGURE 13: Robot's path in the environment with obstacles.

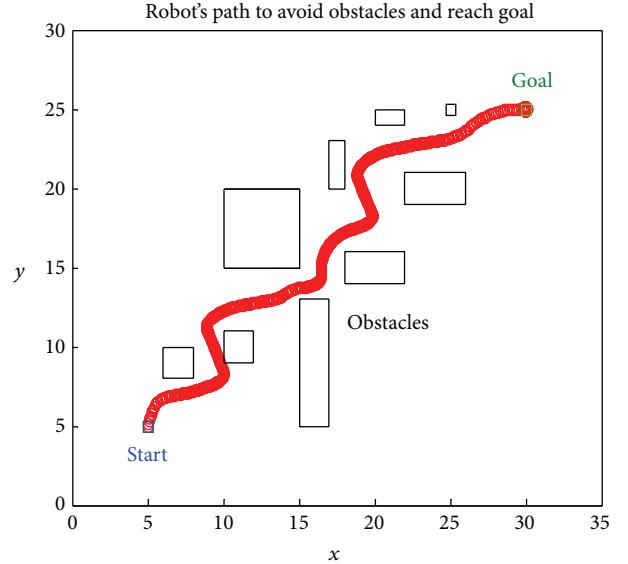
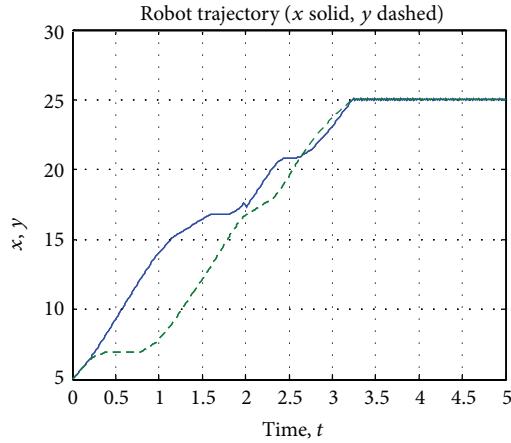


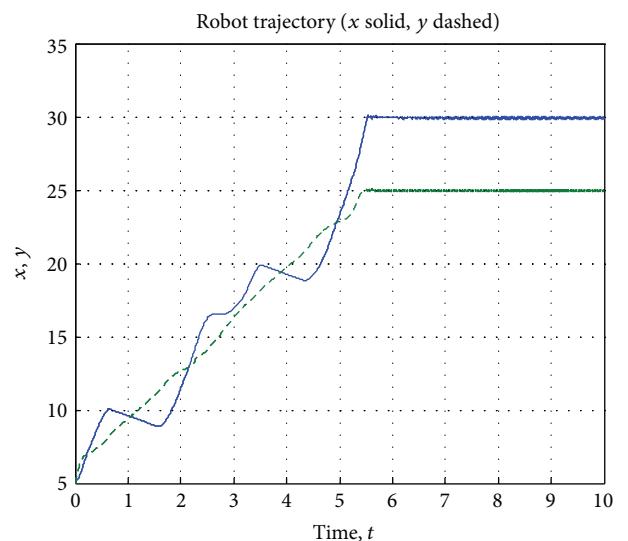
FIGURE 15: Robot's path in a clustered environment.

FIGURE 14: Robot's separated trajectories in the environment with obstacles (blue solid— x direction, green dashed— y direction).

there is no any obstacle so the GS behavior weight is 1 and OA weight is 0.

5.2. Simulation in the Environment with Obstacles. The robot now is moving in the environment with obstacles. The initial position is also $(x_m, y_m) = (5, 5)$ and the final position is $(25, 25)$.

When the robot is close to obstacles, it must decrease its speed and turn for safety. So, the weight of OA behavior is high when the robot is close to the obstacles. Otherwise, the weight is low when the obstacles are far away. The robot is set at a particular start point and the goal is defined in Cartesian coordinates with respect to the start position. The simulation results are shown in Figure 13. It demonstrates that the fuzzy behavior controller has a good performance. From Figure 14, we also can see that the robot reaches the goal after about 3.2 seconds.

FIGURE 16: Robot's separated trajectories in a clustered environment (blue solid— x direction, green dashed— y direction).

5.3. Simulation in a Clustered Environment. Figure 15 shows the results in a more complex situation (a clustered environment). It illustrates the method's ability to navigate the robot in very small gaps and even turning away from the goal when it is necessary to avoid obstacles. Here, the initial position is set at $(x_m, y_m) = (5, 5)$ and the final position is set at $(x_g, y_g) = (30, 25)$.

The robot reaches the goal after about 5.5 seconds as shown in Figure 16. Since there are many obstacles, the robot takes much time to reach this goal compared to the simulations in Sections 5.1 and 5.2.



FIGURE 17: Voyager 2 robot in a laboratory environment: a representative illustration.

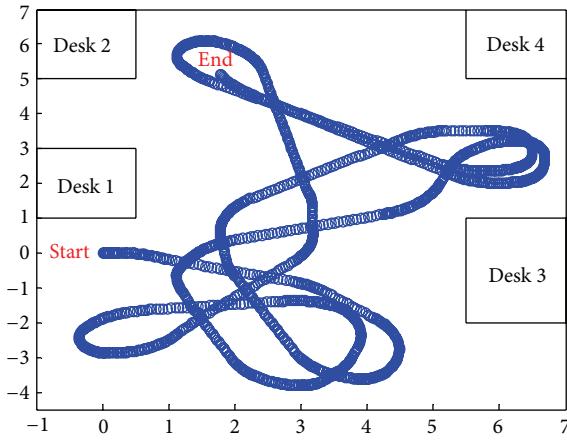


FIGURE 18: Robot roaming path in the laboratory environment.

6. Experiment

Our used real robot is a mobile robot endowed with a ring of 24 sonars, see the real robot in Figure 17. This robot platform is configured with two drive wheels and two swivel casters for balance. Each wheel is driven independently by a motor with 10 : 1 gear ratio which enables the robot to drive at a maximum speed of 3 m/s and climb a 15% grade [15]. The wheel diameter is 217 mm and the mobile robot base width is $D = 446$ mm.

6.1. Experiment for Robot Roaming. This experiment allows the robot to wander within a small area provided with static obstacles, for example, desks and walls, and dynamic obstacles, for example, moving humans. The weight of OA behavior is very high and the GS behavior is very low. The experiment is stopped after a given time or distance of wandering. Figure 18 shows the robot path in the laboratory environment. The trajectory is indicated by the chain of blue circles. The program draws the circle once every 0.02 second. A concentration of circles indicates that the robot is moving more slowly at that moment. Figures 19 and 20 show the variations of robot speed and turn angle during the wandering.

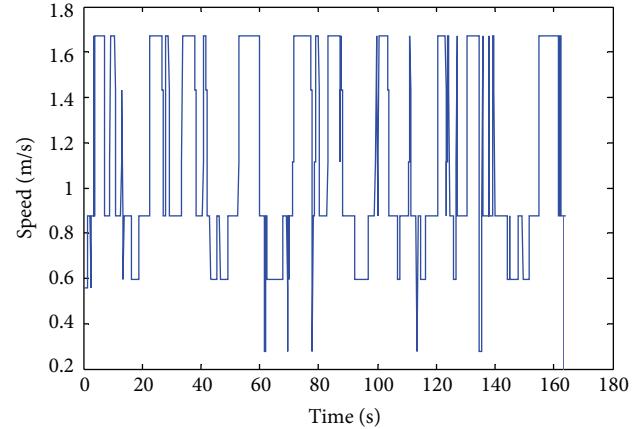


FIGURE 19: The speed of the robot during wandering.

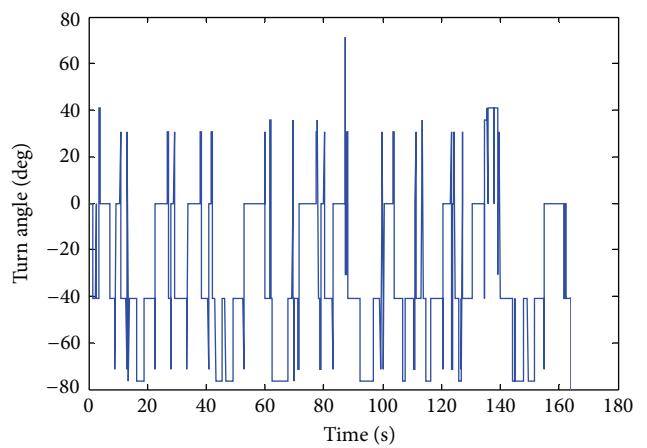


FIGURE 20: The turn angle during wandering.

6.2. Experiment of Robot Avoiding Obstacles and Reaching a Goal. Now the robot's task is using the proposed navigation method to avoid obstacles and reach a goal. For this experiment, the robot is set at the start point $(x_m, y_m) = (0, 0)$ and the goal is defined in Cartesian coordinates with the position of $(x_g, y_g) = (3, 5)$. The obstacles configurations were mapped and the localization data of the robot are recorded and plotted in Figure 21. We can see clearly that the robot reaches the goal and avoids the obstacles.

Figures 22 and 23 show the variations of robot's speed and turn angle, respectively. The positive degree implies turning left while negative indicates right. The robot reaches the goal in almost 20 seconds.

The results presented in the previous sections show the performance of the proposed behavior-based control (navigation) method. Mobile robot usually has uncertain and incomplete information about the environment, so fuzzy rules provide an attractive means for mapping sensor data to appropriate control actions.

The main difference between our approach and the existing ones is that they have different reasoning and different motion commands, while in our approach the final motor

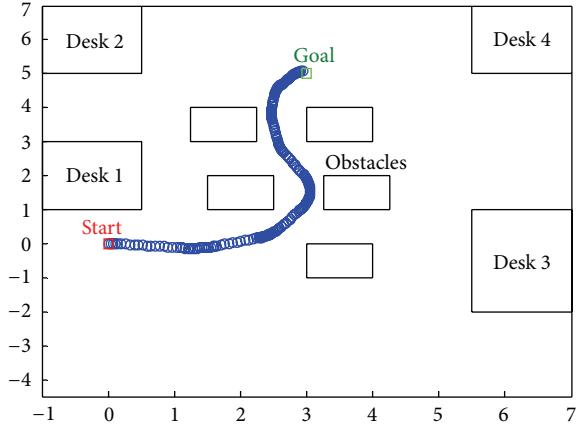


FIGURE 21: Robot avoids obstacles and reaches the goal.

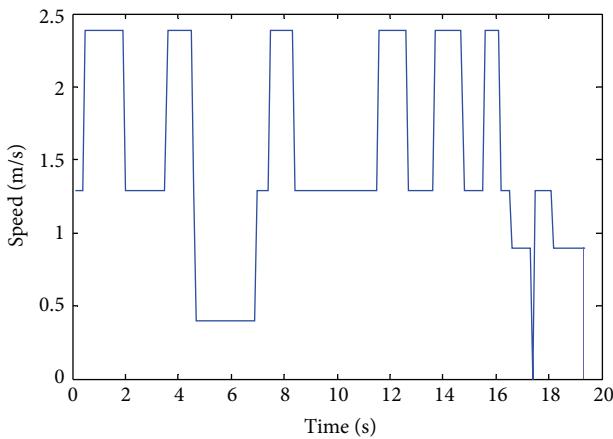


FIGURE 22: The speed of the robot during the task period.

command is generated by fusing different behavior-based fuzzy logic controllers into a uniform representation.

7. Conclusions

This paper presented a new behavior-based fuzzy control method for mobile robot navigation. This method takes angular velocities of driving wheels as outputs of different behaviors. Fuzzy logic is used to implement the specific behaviors. In order to reduce the number of input variables, we introduced a limited number of intermediate variables to guarantee the consistency and completeness of the fuzzy rule bases.

To verify the correctness and effectiveness of the proposed approach, simulation and experiments were performed. A Voyager II robot equipped with a ring of ultrasonic sensors was involved in the real experiments. The promising results demonstrated that our method is feasible and reasonable for navigating mobile robot. This method could be extended for any behavioral system by introducing more flexible behavior arbitration strategies.

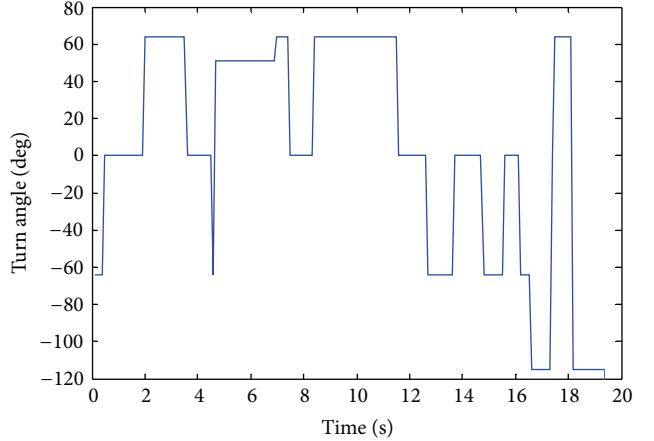


FIGURE 23: The turn angle during the task period.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grant no. 61075113, the Excellent Youth Foundation of Heilongjiang Province of China under Grant no. JC2012012, and the Fundamental Research Funds for the Central Universities, under Grant no. HEUCFZ1209.

References

- [1] A. Saffiotti, "Fuzzy logic in autonomous robotics: Behavior coordination," in *Proceedings of the 6th IEEE International Conference on Fuzzy Systems*, pp. 573–578, July 1997.
- [2] E. Tunstel, Y. Lippincott, and M. Jamshidi, "Behaviour hierarchy for autonomous mobile robots: fuzzy-behaviour modulation and evolution," *International Journal of Intelligent Automation and Soft Computing*, vol. 3, no. 1, pp. 37–49, 1997.
- [3] K.-T. Song and J.-Y. Lin, "Behavior fusion of robot navigation using a fuzzy neural network," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 4910–4915, October 2006.
- [4] B. Lei and W. Li, "A fuzzy behaviours fusion algorithm for mobile robot real-time path planning in unknown environment," in *Proceedings of the IEEE International Conference on Integration Technology (ICIT '07)*, pp. 173–178, March 2007.
- [5] Q.-Y. Bao, S.-M. Li, W.-Y. Shang, and M.-J. An, "A fuzzy behavior-based architecture for mobile robot navigation in unknown environments," in *Proceedings of the International Conference on Artificial Intelligence and Computational Intelligence (AICI '09)*, pp. 257–261, November 2009.
- [6] L. A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems*, vol. 1, no. 1, pp. 3–28, 1978.
- [7] F. Abdessemed, K. Benmahammed, and E. Monacelli, "A fuzzy-based reactive controller for a non-holonomic mobile robot," *Robotics and Autonomous Systems*, vol. 47, no. 1, pp. 31–46, 2004.
- [8] K. Samsudin, F. A. Ahmad, and S. Mashohor, "A highly interpretable fuzzy rule base using ordinal structure for obstacle avoidance of mobile robot," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 1631–1637, 2011.
- [9] M. F. Selekwa, D. D. Dunlap, D. Shi, and E. G. Collins Jr., "Robot navigation in very cluttered environments by preference-based

- fuzzy behaviors," *Robotics and Autonomous Systems*, vol. 56, no. 3, pp. 231–246, 2008.
- [10] M. Wang and J. N. K. Liu, "Fuzzy logic-based real-time robot navigation in unknown environment with dead ends," *Robotics and Autonomous Systems*, vol. 56, no. 7, pp. 625–643, 2008.
- [11] O. R. E. Motagh, T. S. Hong, and N. Ismail, "Development of a new minimum avoidance system for a behavior-based mobile robot," *Fuzzy Sets and Systems*, vol. 160, no. 13, pp. 1929–1946, 2009.
- [12] M. Norouzi, A. Karambakhsh, M. Namazifar, and B. Savkovic, "Object based navigation of mobile robot with obstacle avoidance using fuzzy controller," in *Proceedings of the IEEE International Conference on Control and Automation (ICCA '09)*, pp. 169–174, December 2009.
- [13] D. Wang, Y. Zhang, and W. Si, "Behavior-based hierarchical fuzzy control for mobile robot navigation in dynamic environment," in *Proceedings of the Chinese Control and Decision Conference (CCDC '11)*, pp. 2419–2424, May 2011.
- [14] R. El Harabi, S. B. A. Naoui, and M. N. Abdelkrim, "Fuzzy control of a mobile robot with two trailers," in *Proceedings of the International Conference on Renewable Energies and Vehicular Technology*, pp. 256–262, 2012.
- [15] Beijing UPTECH Robotics, "UP-Voyager IIA, Autonomous Mobile Robot. Experimental Guide Book," <http://www.up-tech.com/eng/>.

Research Article

The Algorithm of Link Prediction on Social Network

Liyan Dong,^{1,2} Yongli Li,³ Han Yin,^{1,2} Huang Le,^{1,2} and Mao Rui^{1,2}

¹ College of Computer Science and Technology, Jilin University, Changchun 130012, China

² Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China

³ School of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, China

Correspondence should be addressed to Yongli Li; liyl603@nenu.edu.cn

Received 20 May 2013; Revised 17 August 2013; Accepted 17 August 2013

Academic Editor: William Guo

Copyright © 2013 Liyan Dong 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.

At present, most link prediction algorithms are based on the similarity between two entities. Social network topology information is one of the main sources to design the similarity function between entities. But the existing link prediction algorithms do not apply the network topology information sufficiently. For lack of traditional link prediction algorithms, we propose two improved algorithms: CNGF algorithm based on local information and KatzGF algorithm based on global information network. For the defect of the stationary of social network, we also provide the link prediction algorithm based on nodes multiple attributes information. Finally, we verified these algorithms on DBLP data set, and the experimental results show that the performance of the improved algorithm is superior to that of the traditional link prediction algorithm.

1. Introduction

Currently with the rapid development, online social network has been a part of people's life. A lot of sociology, biology, and information systems can use the network to describe, in which nodes represent individual and edges represent the relationships between individuals or the interaction between individuals. Therefore, the study of complex networks has been the important branch of many scientific fields. Link prediction is an important task in link mining. Link prediction is to predict whether there will be links between two nodes based on the attribute information and the observed existing link information. Link prediction not only can be used in the field of social network but can also be applied in other fields. As in bioinformatics, link prediction can be used to discover interactions between proteins [1]; in the field of electronic commerce, link prediction can be used to create the recommendation system [2]; and in the security field, link prediction can help to find the hidden terrorist criminal gangs [3]. Link prediction is closely related to many areas. Therefore, in recent years there are a lot of correlation algorithms proposed to solve the problem of link prediction.

2. The Summary of Social Network

In real life, the individuals are not independent of each other. They are mutually contacted and affected. If we only pay attention to individual attributes, while ignoring the relationships between individuals, this is bound to affect the accuracy and comprehensiveness of analysis. Social network represents the relationship between social entities (such as each person, social group). Social network analysis focuses on explaining the hidden patterns and the effects of these relationships. It is based on such an assumption, namely, the individuals in social groups are interdependent, not independent autonomous units. Social network includes a set of objects and relationships between them [4]. These relationships can be any type of social relationships, such as friendship, the purchase of relationship. Social networks can be represented by graph. Graph G contains nodes set V and edges set E . We can use $G = \langle V, E \rangle$ to represent the graph G . In this paper, the nodes set represents objects, and the edges set represents the relationships between objects.

The properties of social network are small world effect, scale-free effect, and cluster effect. Small world effect is

produced by American psychologist Stanley Milgram in 1969 [5]. He found that assign the name of the recipient randomly, and send a message to his own people, and so on, this message can reach the hands of the recipient in a relatively short path (about 6 people). The result was that the famous “six degrees of separation” theory was generated. Social network, internet network, and simulation network have small world properties. In the network, small world effect refers to that the average distance in the network is very small compared to the size of the network. That is to say, each pair of nodes can be connected through a short path in a network [6]. The scale-free effect refers to that most nodes’ links are very small in the network; only a few nodes have lots of links. In this network, nodes with high degree are called hubs (hinge node). The hub node dominates the network operation. Scale-free effect displays that node degree distribution is seriously uneven in large-scale network. Clustering effect of social network refers to that there is a circle of friends, acquaintances, rings, and other small groups in social network. Each member of the small group knows each other. This phenomenon can be described by graph; namely, there is many fully connected subgraphs in social network.

3. The Traditional Link Prediction Algorithms

The link prediction is an important research field in data mining. It has a wide range of scenarios. Many data mining tasks involve the relationship between the objects. Link prediction can be used for recommendation systems, social networks, information retrieval, and many other fields.

Given a snapshot graph of the social network at a moment $G = \langle V, E \rangle$ and the node v_i and the node v_j , link prediction is to predict the probability of the link between the node v_i and the node v_j . It can be seen through the definition of link prediction that the link prediction task is divided into two categories. The first category is to predict that the new link will appear in future time. The second category is to forecast hidden unknown link in the space.

The easiest framework of link prediction algorithm is based on the similarity of the algorithm. Any pair of node x and node y , we have assigned to this node is a function $\text{Similarity}(x, y)$, this function is defined as the similarity function between nodes x and y . Then sorting the nodes pair in accordance with the function values from the largest to smallest, the greater the value of the similarity function, the greater the probability of the link in the nodes.

Here we introduce some simple link prediction similarity indexes.

3.1. Local Similarity Index

3.1.1. Common Neighbors. Assume that the node $v \in V$; then the neighbors of the node set $\Gamma(v) := \{t \mid (t, v) \in E \vee (v, t) \in E \wedge t \neq v\}$; that is, $\Gamma(v)$ is the set of all the neighbors of node v . The common neighbors of node u and node v refer to the jointly owned neighbors by node u and node v .

For the undirected graph, the common neighbors can use the following definition:

$$\text{Similarity}(u, v) = |\Gamma(u) \cap \Gamma(v)|. \quad (1)$$

Kossinets and Watts analysis of the large-scale social network, found that two students who have more mutual friends will have greater possibility to become friends [7].

3.1.2. Preferential Attachment. Preferential attachment mechanism can be used to generate scale-free network evolution model. The probability of generating a new link of node u is directly proportional to the degree of the node [8]. This is the same as the truth “the rich are getting richer” in economics. Therefore, the probability of the link between node u and node v is directly proportional to $d_u \times d_v$. Inspired by this mechanism, the PA similarity index can be defined as follows:

$$\text{Similarity}(u, v) = d_u \times d_v. \quad (2)$$

It may be noted that the similarity index does not require any node neighbor information; therefore, this similarity index has the lowest computational complexity.

3.1.3. Adamic-Adar [9]. This similarity index assigns a higher similarity function value to a small degree node. Adamic-Adar algorithm believes that an affair owned by less objects, compared to owned by more objects, has greater effect on link prediction. Its definition is as follows:

$$\text{Similarity}(u, v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{\log d_z}. \quad (3)$$

3.1.4. Resource Allocation. This similarity index is inspired by the ideas of complex network resources dynamically allocated [10]. In pair of nodes u, v that have no direct link, node u can allocate some resources to the node v through their common neighbor. Their common neighbors assume the role of passers. In the simplest case, we assume that each passer has a unit of resources; it assigns these resources to its neighbors evenly. Therefore, the similarity of node u and node v can be defined as the number of resources that node u get from node v ; namely,

$$\text{Similarity}(u, v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{d_z}. \quad (4)$$

3.2. Overall Similarity Index

3.2.1. Katz [11]. In 1953, Katz described the similarity using the global path. The idea of the method is that the more paths between two nodes are, the greater the similarity between two nodes is. Katz measure is defined as follows:

$$\begin{aligned} \text{Similarity}(u, v) &= \sum_{l=1}^{l_{\max}=\infty} \beta^l \cdot |\text{path}_{u,v}^l| \\ &= \beta A_{uv} + \beta^2 (A^2)_{uv} + \beta^3 (A^3)_{uv} + \dots, \end{aligned} \quad (5)$$

where $|\text{path}_{u,v}^l|$ is the number of paths between node u and node v and the length of the path is l . β is a parameter between 0 and 1. This parameter is used to control the contribution of path to the similarity; the longer the path is, the less contribution the path made to the similarity. The computational complexity of Katz measure is n^3 , so the measure is not suitable for large-scale network.

3.2.2. Random Walk with Restart (RWR) [12]. This indicator is a direct application of the PageRank algorithm. A random walker starting from node u will reach its random neighbor with probability c repeatedly and return the node u with the probability $1 - c$. q_{uv} represents the probability of the random walker reaching node v in the steady state condition. Therefore, we have $\vec{q}_u = cP^T\vec{q}_u + (1 - c)\vec{e}_u$, where P is the transfer matrix. If the node u is connected with node v , then $P_{uv} = 1/d(u)$; else $P_{uv} = 0$. So the solution is simple; namely, $\vec{q}_u = (1 - c)(I - cP^T)^{-1}\vec{e}_u$. RWR coefficient can be defined as

$$\text{Similarity}(u, v) = q_{uv} + q_{vu}. \quad (6)$$

Compared to the local similarity index, the global similarity index needs more overall network topology information. Although the performance of the overall similarity index is better than the local similarity index, it has two fatal flaws: first, the global similarity index calculation is very time consuming, and when the network is huge, this calculation program of the global similarity index does not work. Second, sometimes the global topology information is not available, especially when we use a decentralized approach to implement the algorithm. Therefore, how to design a similarity index is particularly important, which is easy to calculate and its accuracy is high.

Although the traditional link prediction algorithms have made some prediction effect, they do not make full use of the topology information. Common neighbor algorithm treats all the common neighbors equally; it does not distinguish the different neighbors' different effects on the link prediction. Katz algorithm distinguishes the different path's influences which have different lengths, but it does not distinguish the influence of the paths with the same length on link prediction. These algorithms only consider the topology characteristics of the network, treat the social networking static, and ignore the time attributes and node attribute of social network. How to integrate the topology characteristics, time characteristics, and node attributes of social network reasonably is an enormous challenge for link prediction facing.

4. Link Prediction Algorithms Based on Node Guidance Capability

If there is the case in Figure 1, the traditional link prediction algorithm think the degree of similarity calculated between node A and node B is the same as the similarity between node X and node Y . However, when we extracted the subgraph that contained the node A , node B , and their common neighbors, the node X , node Y , and their common neighbors, as shown in Figure 2, we can see that the paths between node A and node B are more than the paths between the node X and the

node Y . According to Katz algorithm the similarity of node A and node B is higher than the similarity of node X and node Y .

Observing the density of the extracted common neighbor subgraph, if the common neighbors subgraph is denser, the nodes in the subgraph made more contribution to link prediction. Assigning the dense of the subgraph to each node, we can see that if the common neighbor occupied the greater the proportion in the neighbor of the node, it has greater ability to generate new link between the node A and node B . Therefore, we designed the following formula to measure the guiding force of the node:

$$\text{Guidance-force}(z) = \frac{|\phi(z)|}{\log d_z}, \quad (7)$$

where d_z is the degree of node z and $|\phi(z)|$ is the degree of node z in the extracted subgraph.

4.1. CNGF Algorithm. By introducing the definition of node guidance capability, we know that the guidance capabilities of the different nodes are different. The similarity between the two nodes can be represented by the sum of each node's guidance capability. The guidance capability of the common neighbors is greater and the likelihood of the new link between the two nodes is greater. On this basis, we define the formula of the similarity degree of two nodes:

$$\text{Similarity}^{\text{CNGF}}(u, v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{|\phi(z)|}{\log d_z}, \quad (8)$$

where $|\phi(z)|$ is the number of links that the node connected with the common neighbors.

With the previous formula, we can give the pseudocode of CNGF algorithm based on node guidance capability (see Algorithm 1).

The computational complexity of CNGF algorithm is $O(N^2)$; N is the maximum of all the nodes' degree. This computational complexity is also acceptable in large-scale network.

4.2. KatzGF Algorithm. The idea of Katz algorithm is that if there are more paths between two nodes, the possibility of a new link existing in the two nodes is greater. Katz algorithm did not distinguish the contribution of the paths with the same path length. The node guidance capability proposed in this paper can distinguish the contribution of different nodes effectively. We can make use of the node guidance capability in Katz algorithm. Because of different nodes, the contribution of the paths with the same path length is different. Based on the previous ideas, we designed the KatzGF algorithm. The KatzGF algorithms integrated the local node information and the global social network information reasonably.

The most important part of the KatzGF algorithm is how to design the formula for the degree of similarity between two nodes. Taking into account the contribution of the different

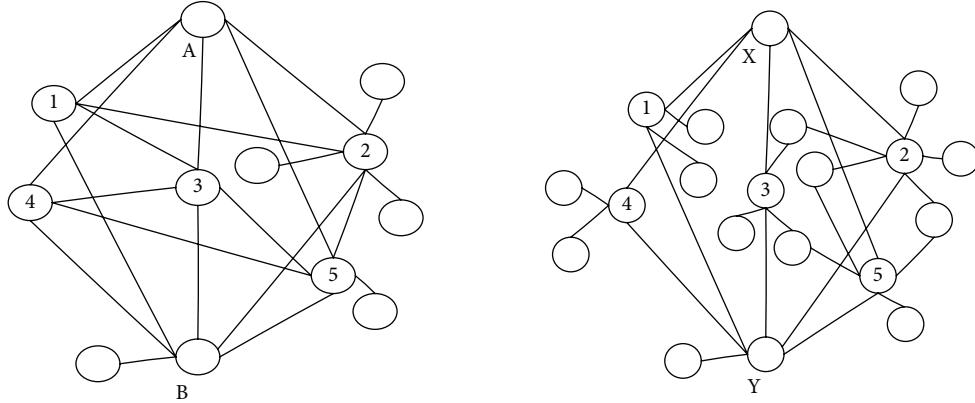


FIGURE 1: Two social network graphs with the same node degree.

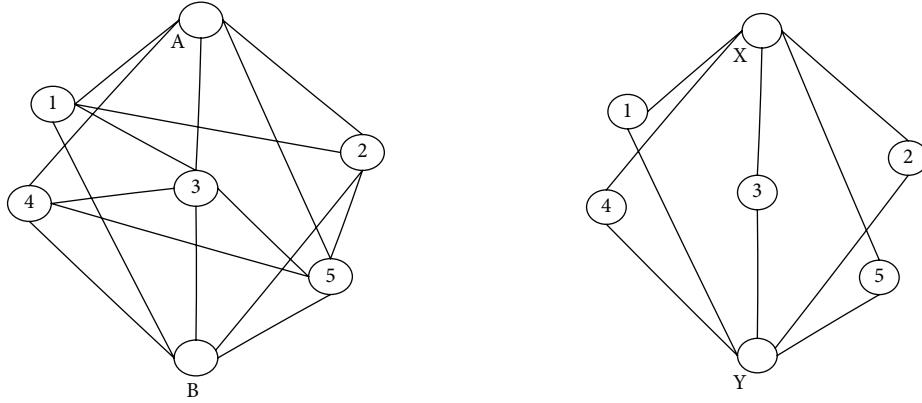


FIGURE 2: The extracted subgraph contains the prediction nodes and their common neighbor.

nodes on the same path length is different, we can design the similarity formula of KatzGF:

$$\text{Similarity}^{\text{KatzGF}}(u, v) = \sum_{l=1}^{l=\infty} \beta^l \left(\sum_{z \in \text{Path}_{uv}^l} \frac{|\phi(z)|}{\log d_z} \right), \quad (9)$$

where β is a parameter between 0 and 1. This parameter is used to control the contribution of path to the similarity; the longer the path is, the less contribution the path made to the similarity. $|\phi(z)|$ is the degree of node z in the subgraph containing the test nodes and all the path nodes between the test nodes. d_z is the degree of node z . The formula first calculated the guidance capability of each node on the path between node pairs and then put the sum of all path nodes' guidance capability as the contribution of the path to link prediction.

With the previous formula, we can give the pseudocode of KatzGF algorithm (see Algorithm 2).

The computational complexity of KatzGF algorithm is $O(K^n)$; K is the number of nodes in social network. It can be seen that when the network size is large, the time complexity degree of the KatzGF algorithm is very high. And KatzGF algorithm needs to know the global information of social networks. This is also difficult to achieve in real life, so KatzGF algorithm is not suitable for large-scale network.

5. Link Prediction Algorithms Based on Node Multiple Attributes

Traditional link prediction algorithm only intercepts a time snapshot of the social network, ignoring the time characteristics of the network. In fact, social network changes constantly over time; it is not static.

Considering the time characteristics of the network, social network graph can be divided into different graph sequences in accordance with a certain time step $G = \langle G_{\Delta t_1}, G_{\Delta t_2}, \dots, G_{\Delta t_n} \rangle$. These snapshots graphs are disjoint. And $\{\Delta t_1, \Delta t_2, \dots, \Delta t_n\}$ is a set of disjoint time. For all i, j , $1 \leq i \leq j \leq n$, Δt_i is earlier than Δt_j . The selection of time step depends on the application. The time step can be set to a month or a year.

In order to illustrate the role of the node time statistics on link prediction better, we can make an analogy of the moving average line of finance. We use the moving average line (moving averages) to extract long-term development trends from the short-term noise data in finance. The moving average is the average of an index value in a certain period time. For example, we consider the average degree value of node v_i within 50 time steps; the average node degree is $\sum_{t=1}^{t=50} \text{degree}_t(v_i)/50$. In social networks, we can remove some

Input: social network graph $G = \langle V, E \rangle$, node x , node y
 Output: the similarity of node x and node y
 The description of the algorithm:
 (1) Find the common neighbor set $xy.common_neighbor$ of the node pair.
 (2) Extract the sub-graph which contains the tested node pair and their common neighbors.
 (3) While (the common neighbor set is not null){
 (4) Calculate the degree of node v , and get $v.degree$. Node v is one node of the
 common neighbor set.
 (5) Calculate the degree of node v in the sub-graph extracted in the Step 2, get
 $v.common_degree$.
 (6) Calculate the guidance capability of node v , Guidance (v) = $v.common_degree / \log(v.degree)$
 (7) The similarity of node x and node y is Similarity. $xy+ = \text{Guidance}(v)$

ALGORITHM 1: CNGF algorithm.

Input: social network graph $G = \langle V, E \rangle$, node x , node y
 Output: the similarity of node x and node y
 The description of the algorithm:
 (1) Find all paths between node x and node y which length are less than 6. Put all the
 nodes in the paths into the set $xy.path.nodes$. Record the length of each path.
 (2) Extract the sub-graph which contains the tested node pair and all the nodes in their paths.
 (3) for (from the first path to the last path){
 (4) while (the nodes set of the path is not null){
 (5) Calculate the degree of node v , and get $v.degree$. Calculate the degree of node v
 in the extracted sub-graph in the Step 2, get $v.path.degree$.
 Calculate the guidance capability of this node
 Guidance-capability. $v = v.path.degree / \log(v.degree)$.
 (6) Calculate the weight of this path Guidance.capability.path+ = Guidance.capability.v}.
 (7) Calculate the similarity of node x and node y Similarity. $xy+ = \beta^l$ Guidance.capability.path}

ALGORITHM 2: KatzGF algorithm.

noise data using the node attribute average value and get a stable long-term trend.

Combining the node guidance capability and the average of node's degree, we propose a link prediction based on node topology attributes and network time attributes. In this section, we make some modifications of the node guidance capability's definition and add the time attributes into the definition of node guidance capability. We get a graph sequences $G = \langle G_{\Delta t_1}, G_{\Delta t_2}, \dots, G_{\Delta t_n} \rangle$ from the social network graph $G = \langle V, E \rangle$, where each snapshot graph is an undirected graph. In each snapshot graph, we calculate $|\phi(z)_{\Delta t_i}|$ and $d_{z-\Delta t_i}$, respectively. Then use the idea of nodes moving average and compute the average degree of node in the entire network and the extracted subgraph. The node's average degree in the extracted subgraph is $|\phi(z)|_{avg} = \sum_{i=1}^{i=n} |\phi(z)_{\Delta t_i}| / n$. The node's average degree in the entire graph is $d_{z-avg} = \sum_{i=1}^{i=n} d_{z-\Delta t_i} / n$. Finally, putting $|\phi(z)_{avg}|$ and d_{z-avg} into the definition formula of node guidance capability, we can get a new formula that combines the time attributes:

$$\text{Guidance-force}(z)_{time} = \frac{|\phi(z)|_{avg}}{\log(d_{z-avg})}. \quad (10)$$

With the previous new definition, we can modify the CNGF algorithm and KatzGF algorithm. We combine CNGF algorithm with social network topology attributes and time attributes and then get CNGF-T algorithm. The new similarity formula is

$$\text{Similarity}^{CNGF-T}(u, v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{|\phi(z)|_{avg}}{\log(d_{z-avg})}. \quad (11)$$

We also can get the new similarity formula in KatzGF-T:

$$\text{Similarity}^{KatzGF-T}(u, v) = \sum_{l=1}^{l=\infty} \beta^l \left(\sum_{z \in \text{Path}_{uv}^l} \frac{|\phi(z)|_{avg}}{\log(d_{z-avg})} \right). \quad (12)$$

6. The Algorithm Implementation and Verification

In this experiment, we used DBLP (Digital Bibliography and Library Project) experimental data sets to verify the performance of the link prediction algorithm. This data set is real. It covers 28 international conferences records related to data mining, machine learning, and so on.

We use AUC indicators to evaluate the experimental results. The AUC index refers to the area under ROC curve. ROC curve puts false positive rate as x -axis and puts true positive rate as cases y -axis. The point in the curve corresponds to the performance of the algorithm under each threshold. The greater the AUC value is the better performance the predictors have.

6.1. Data Preprocessing. The real conetwork is particularly sparse. In order to achieve the best performance of the algorithm, we need to preprocess the data set. The preprocessing of the data set can be divided into the following steps to complete.

Step 1. Intercept the data from 2000 to 2006.

Step 2. Delete the independent writings node.

Step 3. Remove the high degree of complete graphs.

Step 4. Construct the known network. Using the data from 2000 to 2004, we can construct the known conetwork. Put the topology structure attributes and the time attributes as the training set. And put the data from 2005 to 2006 as the unknown network.

Step 5. Screen less the known network. We did not select all the nodes of the data, so we should screen less the appropriate link. If a node is deleted, then all the links on the node should be removed.

6.2. The Experimental Results. By setting different thresholds, we calculate the true positive rate and the false positive rate, respectively. Then we draw the corresponding ROC curve. Firstly, we give the ROC curve of the local algorithm common neighbor and the ROC curve of the improved algorithms CNGF and CNGF_T, as shown in Figure 3. The ROC curve shows the overall impact of different thresholds for link prediction. It can be seen from the figure that the performance of three algorithms is very similar in the beginning; however, with the increase of false positive rate, the true positive rate of CNGF_T is larger than CNGF algorithm and common neighbor algorithm. This shows that the prediction performance of the CNGF_T algorithm is the best of the three algorithms.

The ROC curves of Katz algorithm, KatzGF, and KatzGF_T are shown in Figure 4. From the figure, we can see that KatzGF algorithm is better than Katz algorithm, but it is not very obvious in the overall performance. But the ROC curve of KatzGF.T algorithm is significantly closer to the upper left corner than those of the first two algorithms. KatzGF_T algorithm takes into account the time attribute information and the global topology information of social network, so it has the best prediction performance.

7. Conclusion

Firstly, we introduce the concept of social networks and describe the basic nature of social networks: small world,

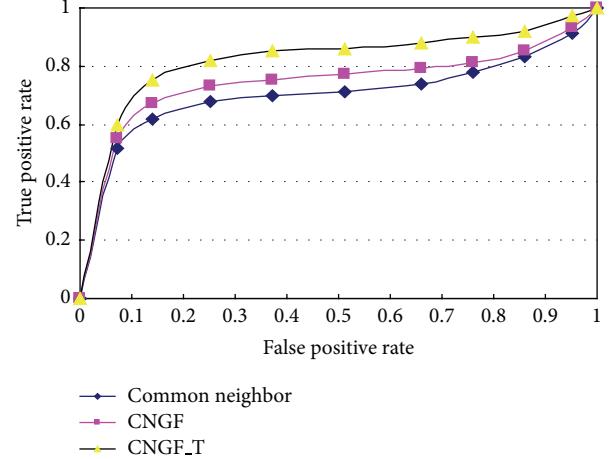


FIGURE 3: The ROC curves of common neighbor algorithm, CNGF algorithm, and CNGF_T algorithm.

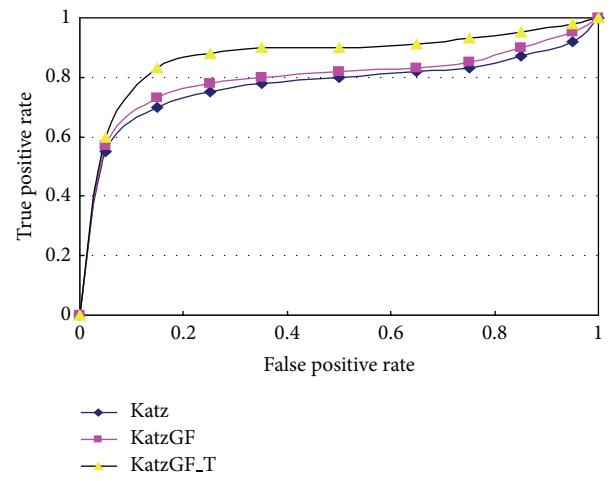


FIGURE 4: The ROC curves of Katz algorithm, KatzGF algorithm, and KatzGF.T algorithm.

scaling, and clustering features. Then we introduce the link prediction definition and link prediction algorithms. Comparing these link prediction algorithms, we analyse the existing problems of the traditional link prediction. For the problems in traditional link prediction on social networks, we proposed the improved CNGF algorithm KatzGF algorithm. And for the lack of static social network, we gave the link prediction algorithm based on multiple attributes. The experimental results show that the prediction performance of the improved algorithm is superior to that of the traditional link prediction algorithm.

References

- [1] E. M. Airoldi, “Mixed membership block models for relational data with application to protein-protein interactions,” in *Proceedings of International Biometric Society-ENAR Annual Meetings*, 2006.

- [2] Z. Huang, X. Li, and H. Chen, "Link prediction approach to collaborative filtering," in *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries, NK*, pp. 141–142, June 2005.
- [3] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," in *SDM Workshop of Link Analysis*, San Francisco, Calif, USA, 2006.
- [4] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*, Cambridge University Press, Cambridge, UK, 1994.
- [5] J. Travers, "An experimental study of the small world problem," *Sociometry*, vol. 32, no. 4, pp. 425–443, 1969.
- [6] R. Alberich, "Marvel Universe looks almost like a real social network," *Physical Review Letters*, vol. 12, no. 4, pp. 12–18, 2006.
- [7] G. Kossinets, "Effects of missing data in social networks," *Social Networks*, vol. 28, no. 3, pp. 247–268, 2006.
- [8] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *American Association for the Advancement of Science*, vol. 286, no. 5439, pp. 509–512, 1999.
- [9] Y. D. Jin, T. Zhou, B. H. Wang, and B. Q. Yin, "Power-law strength-degree correlation from resource-allocation dynamics on weighted networks," *Physical Review Letters*, no. 15, pp. 021–029, 2007.
- [10] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.
- [11] L. A. Adamic and E. Adar, "Friends and neighbors on the Web," *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003.
- [12] S. Brin, "The anatomy of a large-scale hypertextual Web search engine 1," *Computer Networks*, vol. 30, no. 1–7, pp. 107–117, 1998.

Research Article

A Novel Framework Based on ACO and PSO for RNA Secondary Structure Prediction

Gang Wang,^{1,2,3} Wen-yi Zhang,⁴ Qiao Ning,⁴ and Hui-ling Chen⁵

¹ College of Computer Science and Technology, Jilin University, Changchun 130012, China

² Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China

³ College of GeoExploration Science and Technology, Jilin University, Changchun 130026, China

⁴ School of Computer Science and Information Technology, Northeast Normal University, Changchun 130024, China

⁵ College of Physics and Electronic Information, Wenzhou University, Chashan University Town, Wenzhou, Zhejiang, 325035, China

Correspondence should be addressed to Hui-ling Chen; chenhuijing.jlu@gmail.com

Received 24 June 2013; Accepted 12 August 2013

Academic Editor: William Guo

Copyright © 2013 Gang 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.

Prediction of RNA structure is a useful process for creating new drugs and understanding genetic diseases. In this paper, we proposed a particle swarm optimization (PSO) and ant colony optimization (ACO) based framework (PAF) for RNA secondary structure prediction. PAF consists of crucial stem searching (CSS) and global sequence building (GSB). In CSS, a modified ACO (MACO) is used to search the crucial stems, and then a set of stems are generated. In GSB, we used a modified PSO (MPSO) to construct all the stems in one sequence. We evaluated the performance of PAF on ten sequences, which have length from 122 to 1494. We also compared the performance of PAF with the results obtained from six existing well-known methods, SARNA-Predict, RnaPredict, ACRNA, PSOfold, IPSO, and mfold. The comparison results show that PAF could not only predict structures with higher accuracy rate but also find crucial stems.

1. Introduction

RNA functions as an information carrier, catalyst, and regulatory element, perhaps reflecting its importance in the earliest stages of evolution. The structures of RNAs provide insight into the mechanisms behind these functions. Determining sequence is the first step in determining structure, and many billions of nucleotide sequences are now known. The second step is determining secondary structure, and relatively few classes of RNAs currently have known secondary structures [1]. The RNA secondary structure prediction problem is a critical one in molecular biology. Secondary structure as well as tertiary structure can be determined by X-ray crystallography and Nuclear Magnetic Resonance (NMR) spectroscopy. Data analysis tools used for prediction of RNA structure are mainly based on dynamic programming [2]. Recently,

the metaheuristic methods are widely used to predict RNA secondary structure. These methods generally include genetic algorithm (GA) [3], particle swarm optimization (PSO) [4], ant colony optimization (ACO) [5], and simulated annealing (SA) [6]. The state-of-the-art methods are introduced as follows.

For SA, Shapiro and Wu [7] later modified the algorithm by introducing an annealing mutation operator. Tsang and Wiese [8] presented SARNA-Predict, the permutation-based algorithm for RNA secondary structure prediction based on SA with a simple thermodynamic model and studied mainly its convergence behavior. A performance evaluation of SARNA-Predict in terms of prediction accuracy was made via comparison with eight state-of-the-art RNA prediction algorithms. The results presented in this paper demonstrate

that SARNA-Predict can outperform other state-of-the-art algorithms in terms of prediction accuracy. Furthermore, there is substantial improvement of prediction accuracy by incorporating a more sophisticated thermodynamic model.

For GA, Benedetti and Morosetti [9] compared the accuracy of an EA against known RNA structures with the objective of finding optimal and suboptimal structures that were similar. Shapiro et al. [10] modified their EA to study folding pathways using a massively parallel genetic algorithm. Wiese et al. designed a serial EA, RnaPredict [11], which encodes RNA secondary structures as permutations. RnaPredict was parallelized via a coarse-grained distributed EA for RNA secondary structure prediction [12].

For ACO, Yu et al. [13] put forward ACRNA based on ACO for RNA secondary structure prediction. For a given RNA sequence, the set of all possible stems is obtained, and the energy of each stem is calculated and stored at the initial stage. Furthermore, a more realistic formula is used to compute the energy of multibranch loop in the following iteration. Then a folding pathway is simulated, including such processes as construction of the heuristic information, the rule of initializing the pheromone, the mechanism of choosing the initial and next stem, and the strategy of updating the pheromone between two different stems.

For PSO, Geis and Middendorf [14] introduced HelixPSO for finding minimum energy RNA secondary structures. Neethling and Engelbrecht [15] proposed a set-based Particle Swarm Optimization algorithm to optimize the structure of an RNA molecule, using an advanced thermodynamic model. Liu et al. [16] proposed an improved PSO (IPSO). The authors designed an efficient objective function according to the minimum free energy, the number of selected stems, and the average length of selected stems. A promising experimental result was obtained, and the effectiveness and practicability of IPSO for RNA secondary structure prediction was shown. Xing et al. [17, 18] proposed PSOfold based on IPSO. An adaptive parameter controller of PSO based on fuzzy logic is used to improve the balance between exploration and exploitation. Solution conversion strategy (SCS) is designed to enhance the PSO performance in discrete problem such as stem combination.

The metaheuristic methods [19, 20] mentioned previously only paid attention to structure of the sequence but ignored the searching capability of algorithm. PSOfold is our previous work with an attempt to enhance the searching range in PSO, but the crucial stems are not considered. In this paper, we proposed PAF to improve the exploration ability and focused on the influence of the crucial stems.

The main contributions of this paper are described as follows.

- (1) A framework, namely, PAF, was proposed for RNA secondary structure prediction, which includes CSS and GSB.
- (2) In CSS, MACO was proposed to search the crucial stems.

- (3) In GSB, MPSO was designed to construct all the stems in one sequence.

The rest of the paper is organized as follows. Section 2 briefly introduces the theory of ACO and PSO. In Section 3, we present the proposed PAF framework and describe the implementation of CSS and GSB. In Section 4, the experiment results on various public sequences are discussed. Finally, we draw the conclusions of this paper in Section 5.

2. Basic Theory

2.1. ACO. ACO algorithm is biologically inspired from the behavior of colonies of real ants, and in particular how they forage for food. ACO has been formalized into a metaheuristic for combinatorial optimization problems by Dorigo and coworkers [21].

In ACO, an ant k being in node i chooses the next node j with a probability given by the random proportional rule defined as follows [22].

(a) *State Transition Rule.* Consider

$$P_{ij}(t) = \begin{cases} \frac{[\tau_{ij}(t)] \cdot [\eta_{ij}]^\beta}{\sum_{j \in I_i^k} [\tau_{ij}(t)] \cdot [\eta_{ij}]^\beta} & \text{if } j \in I_i^k, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where I_i^k is its feasible neighborhood. The feasible neighborhood excludes nodes already visited in the partial tour of ant i , and it may be further restricted to a candidate set of the nearest neighbors of a city i . Once an ant has visited all nodes, it returns to its starting node.

(b) *State Updating Rule.* After all ants have completed their solutions, pheromone evaporation on all nodes is triggered according to (2). The pheromone on each edge is updated according to the following equation:

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \sum_{k=1}^N \Delta \tau_{ij}^k(t), \quad (2)$$

where m is the number of ants at each iteration and $\rho \in (0, 1)$ is the pheromone evaporation rate. Consider

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{if } \tau_{ij} \in \text{tour done by ant } k, \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where L_k denotes the tour length and Q is a predefined constant.

2.2. PSO. PSO originated from the simulation of social behavior of birds in a flock [23, 24]. In PSO, each particle flies in the search space with a velocity adjusted by its own flying memory and its companion's flying experience. All particles

have objective function values which are decided by a fitness function. Consider the following:

$$\begin{aligned} v_{id}^t &= w \times v_{id}^{t-1} + c_1 \times r_1 (p_{id}^t - x_{id}^t) + c_2 \times r_2 (p_{gd}^t - x_{id}^t), \\ d &= 1, 2, 3, 4, \dots, D, \end{aligned} \quad (4)$$

where c_1 indicates the cognition learning factor, c_2 indicates the social learning factor, and r_1 and r_2 are random numbers uniformly distributed in $U(0, 1)$. Each particle then moves to a new potential solution based on the following equation:

$$x_{id}^{t+1} = x_{id}^t + v_{id}^t, \quad d = 1, 2, \dots, D. \quad (5)$$

Kennedy and Eberhart [25] proposed a binary PSO in which a particle moves in a state space restricted to 0 and 1 on each dimension, in terms of the changes in probabilities that a bit will be in one state or the other. Consider the following:

$$\begin{aligned} x_{id} &= \begin{cases} 1, & \text{rand}() < S(v_{i,d}), \\ 0, & \text{otherwise,} \end{cases} \\ S(v) &= \frac{1}{1 + e^{-v}}. \end{aligned} \quad (6)$$

The function $S(v)$ is a sigmoid limiting transformation and $\text{rand}()$ is a random number selected from a uniform distribution in $[0.0, 1.0]$.

3. PAF

3.1. CSS

3.1.1. MACO

(1) *State Transition Rule.* The task of each ant is to build a set of stems. The ants find an RNA secondary structure via a probabilistic decision rule to move through adjacent states. An ant selects a stem as follows:

$$P_i = \begin{cases} \frac{[\tau_i]^\alpha [\eta_i]^\beta}{\sum_{j \in I_i^k} [\tau_j]^\alpha [\eta_j]^\beta} & \text{if } j \in I_i^k, \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

where α and β are the regulatory factors, τ_i is the amount of pheromone trail on stem i , η_i is the priori available heuristic information, and I_i^k is the remaining stems.

(2) *State Update Rule.* The pheromone trails are updated according to (8) and (9). Consider

$$\tau = (1 - \rho) \cdot \tau_i + \sum_{k=1}^n \Delta \tau_i^k, \quad (8)$$

where ρ is the pheromone trail evaporation rate and τ_i is the quantity per unit of length of the trail substance that is laid on stem i by the k th ant. Also,

$$\Delta \tau_i^k = \begin{cases} \text{Energy} & \text{if } k \text{ uses stem } i \text{ in its tour} \\ 0 & \text{otherwise,} \end{cases} \quad (9)$$

where Energy represents the quality of an ant's solution. If stem i is not included, the zero is returned.

3.1.2. *Algorithm of MACO.* See Algorithm 1.

3.2. GSB

3.2.1. *MPSO.* MPSO was modified based on our previous studies IPSO [16] and PSOfold [18] which could predict the RNA secondary structure with excellent performance. The objective function is improved according to the size of stem and the number of pseudoknots. Consider the following:

$$f_i = aE_i - b \left[\frac{\sum_{j=1}^N \delta(j)}{P} \right] - c \left[\frac{\sum_{k=1}^M \varphi(k)/L + S}{W} \right], \quad (10)$$

where a, b, c is the weight; E_i is the free energy for the secondary structure in the i th particle; δ is the number of pairs of the j th stem; P is the length of possible pairs; φ is the size of stem k which is higher than 4; M is the number of selected stems; L is the total number of stems which are higher than 4; S is the size of pseudoknots; W is the size of possible pairs.

3.2.2. *Algorithm of MPSO.* See Algorithm 2.

4. Results

The parameter details of ACRNA are number of ants = 100, number of iterations = 600, $\rho = 0.2$, $\alpha = 1$, and $\beta = 1$. For IPSO, number of particles = 100, number of iterations = 600, $\omega = 0.9$, $c_1 = 2$, and $c_2 = 2$. For PSOfold, number of particles = 100, number of iterations = 600, $\omega = 0.9$, $c_1 = 2$, and $c_2 = 2$. For CSS, number of ants = 100, number of iterations = 600, $\rho = 0.2$, $\alpha = 1$, and $\beta = 1$. For GSB, number of particles = 100, number of iterations = 600, $\omega = 0.9$, $c_1 = 2$, and $c_2 = 2$. To generate the mfold results presented here, the mfold Web server version 3.1 was used with default settings. One noteworthy setting is the percentage of suboptimality. This percentage allows the user to control the number of suboptimal structures predicted by mfold. In this experiment, the value was set to return the 5 percent lowest energy structures.

The measures used for prediction accuracy on the majority of documents currently are sensitivity, specificity, and F-measure. In RNA secondary structure prediction, TP (true positive) indicates the number of base pairs predicted correctly; FN (false negative) denotes the number of base pairs which existed in real structure but were not predicted correctly; FP (false positive) represents the number of base pairs which existed in no real structure but was mistakenly

TABLE 1: RNA sequence details.

Organism	Accession number	RNA class	Length	Base pairs in known structure
<i>Drosophila virilis</i>	X05914	16S rRNA	784	233
<i>Xenopus laevis</i>	M27605	16S rRNA	945	251
<i>Haloarcula marismortui</i>	AF034620	5S rRNA	122	38
<i>Aureoumbra lagunensis</i>	U40258	Group I intron, 16S rRNA	468	113
<i>Metahizium anisopliae</i> var.	AF197120	Group I intron, 23S rRNA	394	120
<i>Hildenbrandia rubra</i>	L19345	Group I intron, 16S rRNA	543	138
<i>Acanthamoeba griffini</i>	U02540	Group I intron, 16S rRNA	556	131
<i>Homo sapiens</i>	J01415	16S rRNA	954	266
<i>Ailurus fulgens</i>	Y08511	16S rRNA	964	265
<i>Sulfolobus acidocaldarius</i>	D14876	16S rRNA	1494	468

TABLE 2: A comparison of the highest matching base pair structures from PAF and mfold in terms of sensitivity, specificity, and F-measure.

Sequence	Known bps	Predicted bps		TP		FP		FN		Sensitivity		Specificity		F-measure	
		PAF	mfold	PAF	mfold	PAF	mfold	PAF	mfold	PAF	mfold	PAF	mfold	PAF	mfold
<i>D. virilis</i>	233	247	252	103	82	144	170	130	151	44.2	35.2	41.7	32.5	42.9	33.8
<i>X. laevis</i>	251	235	245	116	113	119	132	135	138	46.2	45.0	49.4	46.1	47.7	45.6
<i>H. marismortui</i>	38	31	34	28	29	3	5	10	9	73.7	76.3	90.3	85.3	81.2	80.6
<i>A. lagunensis</i>	113	124	133	85	74	39	59	28	39	75.2	65.5	68.6	55.6	71.7	60.2
<i>M. anisopliae</i> var.	120	110	116	91	92	19	24	29	28	75.8	76.7	82.7	79.3	79.1	78.0
<i>H. rubra</i>	138	141	167	82	83	59	84	56	55	59.4	60.1	58.1	49.7	58.8	54.4
<i>A. griffini</i>	131	160	174	102	95	58	79	29	36	77.9	72.5	63.8	54.6	70.1	62.3
<i>H. sapiens</i>	266	260	258	124	95	136	163	142	171	46.6	35.7	47.7	36.8	47.2	36.3
<i>A. fulgens</i>	265	249	241	99	74	150	167	166	191	37.4	27.9	39.8	30.7	38.5	29.2
<i>S. acidocaldarius</i>	468	483	496	281	271	202	225	187	197	60	57.9	58.1	54.6	59.1	56.2
Averages	202.3	204	211.6	111.1	100.8	92.9	110.8	91.2	101.5	59.64	55.28	60.02	52.52	59.63	53.66

TABLE 3: A comparison of the highest matching base pair structures from SARNA-Predict, RnaPredict, ACRNA, PAF, PSOfold, and mfold in terms of sensitivity and specificity.

Sequence	Length	SARNA-Predict		RnaPredict		ACRNA		PAF		PSOfold		mfold	
		Se	Sp	Se	Sp	Se	Sp	Se	Sp	Se	Sp	Se	Sp
<i>D. virilis</i>	784	42.1	42.2	27.9	26.9	27.9	26.9	44.2	41.7	42.9	41.5	35.2	32.5
<i>X. laevis</i>	945	46.2	48.9	37.1	38.8	45.8	46.9	46.2	49.4	45.6	48.5	45.0	46.1
<i>H. marismortui</i>	122	71.1	90.0	71.1	90.0	76.3	78.4	73.7	90.3	71.1	87.1	76.3	85.3
<i>A. lagunensis</i>	468	74.3	64.6	60.2	51.9	63.7	54.1	75.2	68.6	76.1	64.7	65.5	55.6
<i>M. anisopliae</i> var.	394	63.3	65.0	62.5	62.0	63.3	65.0	75.8	82.7	75.0	74.4	76.7	79.3
<i>H. rubra</i>	543	64.5	56.7	57.2	49.1	58.0	47.9	59.4	58.1	58.0	53.0	60.1	49.7
<i>A. griffini</i>	556	74.0	58.8	61.8	50.3	73.3	55.2	77.9	63.8	76.3	59.2	72.5	54.6
<i>H. sapiens</i>	954	44.7	48.8	33.5	35.6	44.7	48.8	46.6	47.7	44.4	47.2	35.7	36.8
<i>A. fulgens</i>	964	35.1	38.9	30.9	33.9	34.3	38.7	37.4	39.8	35.1	39.6	27.9	30.7
<i>S. acidocaldarius</i>	1494	52.4	51.9	52.4	51.9	58.3	58.0	60	58.1	58.3	58.0	57.9	54.6
Averages	722.4	56.77	56.58	49.46	49.04	54.56	51.99	59.64	60.02	58.28	57.32	55.28	52.52

```

Initialize the parameters of MACO
Randomly initialize the solutions for all the ants
While current number of iterations < Max iteration
    For each ant in the population
        For each stem
            Decide whether to select current stem according to (7)
        End for
        Evaluate the solution according to the energy.
    End for
    For each stem in the set
        Update the pheromones according to (8) and (9)
    End for
End while

```

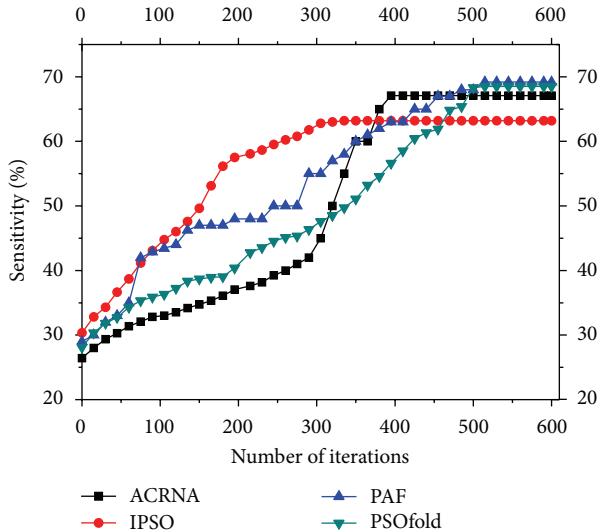
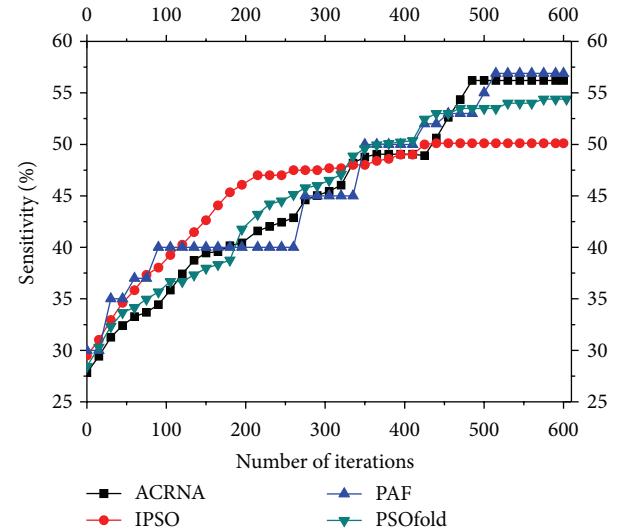
ALGORITHM 1

```

Initialize all the parameters of MPSO
While current number of iterations < Max iteration
    For each particle
        Update its velocity
        Update its position
        Restrict position and velocity
        Calculate fitness and Update local best
    End for
    Update the global best
    Turn the parameters of MPSO via fuzzy logic controllers
End while

```

ALGORITHM 2

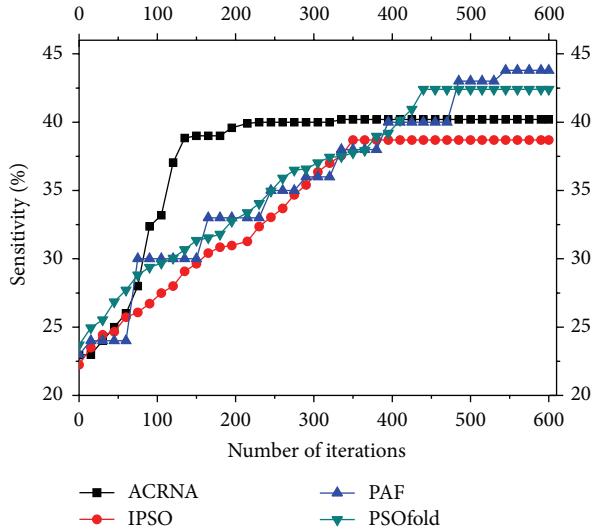
FIGURE 1: Average best sensitivity by running the algorithms 10 times on *A. griffini*.FIGURE 2: Average best sensitivity by running the algorithms 10 times on *H. rubra*.

predicted; TN (true negative) stands for the number of base pairs which were not matched and predicted correctly. The TN is rarely used in actual measurement because it is generally much larger than TP, FN, and FP. Sensitivity (Se)

means the percentage of all base pairs which was correctly predicted in the real structure; specific (Sp) refers to the percentage of all predicted base pairs which was correctly predicted. The general prediction is very difficult for both and

TABLE 4: Average sensitivity and specificity via ACRNA, IPSO, PAF, and PSOfold.

Sequence	Length	ACRNA		IPSO		PAF		PSOfold	
		se	sp	se	sp	se	sp	se	sp
<i>D. virilis</i>	784	25.1	24.5	27.8	27.1	34.1	30.4	32.6	29.9
<i>X. laevis</i>	945	40.2	38.8	38.7	36.7	43.8	41.5	42.4	40.1
<i>H. marismortui</i>	122	70.5	73.6	67.8	68.1	70.6	76.0	69.2	75.4
<i>A. lagunensis</i>	468	61.3	52.6	60.5	51.8	62.4	53.8	63.9	52.6
<i>M. anisopliae</i> var.	394	60.9	63.5	57.1	56.2	66.5	64.1	65.1	63.3
<i>H. rubra</i>	543	56.2	48.8	50.1	41.3	56.9	47.8	54.4	47.8
<i>A. griffini</i>	556	67.1	54.1	63.2	47.3	69.2	52.9	68.5	54.5
<i>H. sapiens</i>	954	34.1	32.1	31.9	33.5	35.8	34.9	32.4	33.3
<i>A. fulgens</i>	964	31.1	30.2	30.2	32.5	34.2	31.8	32.6	31.8
<i>S. acidocaldarius</i>	1494	56.0	55.4	52.5	51.4	54.5	56.3	56.5	55.9
Averages	722.4	50.25	47.36	47.98	44.59	52.8	48.95	51.76	48.46

FIGURE 3: Average best sensitivity by running the algorithms 10 times on *X. laevis*.

is always biased in favor of one side. A metric that combines both the specificity and sensitivity measures into one is *F*-measure; it can be used as a single performance measure for a predictor. The main result of the paper will be concluded by the sensitivity and specificity [18]. The specific formula is as follows:

$$\begin{aligned}
 \text{Se} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\
 \text{Sp} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\
 F &= \frac{2 \times \text{specificity} \times \text{sensitivity}}{\text{specificity} + \text{sensitivity}}.
 \end{aligned} \tag{11}$$

Ten sequences from the comparative RNA website are selected for evaluation of the proposed method, and the details of the sequence are described in Table 1. For these sequences the natural secondary structures are also available from the comparative RNA website. These sequences were chosen as they represent different sequence lengths and come from various genomes of organisms that are exposed to a range of physiological conditions. They represent four RNA classes: 5S rRNA, Group I intron 16S rRNA, 16S rRNA, and Group I intron 23S rRNA. Due to space constraints in some tables, we refer to these specific RNA sequences by an abbreviation of the name of the organism from which they originated [18].

Table 2 shows the comparative results of the highest matching base pair structures between PAF and mfold in regard to sensitivity, specificity, and *F*-measure. PAF predicts fewer base pairs on 8 sequences. PAF obtained higher TP in 7 cases and lower FP in all cases. The values FN predicted by PAF are also lower than mfold in 7 out of 10 cases. For sensitivity, specificity, and *F*-measure, PAF won in 7, 10, and 10 cases, respectively. Generally, PAF performs significantly better than mfold with respect to sensitivity, specificity, and *F*-measure.

Table 3 shows a more detailed analysis of comparing the highest matching base pair structures from SARNA-Predict, RnaPredict, ACRNA, PAF, PSOfold, and mfold according to sensitivity and specificity. The results generated by SARNA-Predict and RnaPredict were taken from the literature [8, 11]. From Table 3, it can be seen that PAF gets better sensitivity on 6 sequences and predicts higher specificity in 8 cases. SARNA-Predict gets better results in 2 cases in terms of sensitivity and in 2 cases with regard to specificity. ACRNA wins on 1 sequence for sensitivity and on 1 sequence for specificity. PSOfold and mfold obtained better sensitivity on one sequence and two sequences, respectively. Among the six methods, PAF gets the best results in most cases. From another point of view, the average performance of PAF on sensitivity and specificity exceeds that of the other methods. It

is demonstrated that PAF is significantly superior to the other 5 methods.

In order to validate the stability of the proposed method, we ran ACRNA, IPSO, PAF, and PSOfold ten times and calculated the average highest matching base pair structures of each algorithm in terms of sensitivity and specificity. The detail results are shown in Table 4. From Table 4, it is easy to see that PAF obtains best sensitivity and specificity on 8 sequences and on 7 sequences, respectively. Thus, this proves that the proposed method performs stably on multiple sequences, considerably surpassing the other methods. The convergence progress is shown in Figures 1, 2, and 3. From the figures, we can clearly see that the proposed method could avoid trapping in the local optimum during the iterations. This is because the crucial stems lead the algorithm to the right direction.

5. Conclusion

In this paper, a framework, PAF, was proposed for RNA secondary structure prediction, which consists of CSS and GSB. In order to preserve crucial structures, MACO in CSS is proposed to find the important stems. MPSO in GSB is developed to generate predicted structures in order to save searching spaces. The experimental results show that the performance of the proposed method is significantly better than those of the other metaheuristic methods in terms of sensitivity, specificity, and *F*-measure. We will try to enhance the performance of convergence and reduce time complexity in the future.

Acknowledgment

This research was supported by The Chinese Government's Executive Program "Instrumentation development and field experimentation" (SinoProbe-09).

References

- [1] D. H. Mathews and D. H. Turner, "Prediction of RNA secondary structure by free energy minimization," *Current Opinion in Structural Biology*, vol. 16, no. 3, pp. 270–278, 2006.
- [2] S. Ray and S. Pal, "RNA secondary structure prediction using soft computing," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 99, pp. 1–16, 2012.
- [3] E. Khosravani, G. Moradi, and S. Sajjadifar, "An accurate thermodynamic model to predict phase behavior of clathrate hydrates in the absence and presence of methanol based on the genetic algorithm," *Journal of Chemical Thermodynamics*, vol. 57, pp. 286–294, 2013.
- [4] Y. Liu, G. Wang, H. Chen, H. Dong, X. Zhu, and S. Wang, "An improved particle swarm optimization for feature selection," *Journal of Bionic Engineering*, vol. 8, no. 2, pp. 191–200, 2011.
- [5] Y. Li, G. Wang, and H. Chen, "An ant colony optimization based dimension reduction method for high-dimensional datasets," *Journal of Bionic Engineering*, vol. 10, no. 2, pp. 242–250, 2013.
- [6] D. F. Ouyang, "Investigating the molecular structures of solid dispersions by the simulated annealing method," *Chemical Physics Letters*, vol. 554, pp. 177–184, 2012.
- [7] B. A. Shapiro and J. C. Wu, "An annealing mutation operator in the genetic algorithms for RNA folding," *Computer Applications in the Biosciences*, vol. 12, no. 3, pp. 171–180, 1996.
- [8] H. H. Tsang and K. C. Wiese, "SARNA-predict: accuracy improvement of RNA secondary structure prediction using permutation-based simulated annealing," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 7, no. 4, pp. 727–740, 2010.
- [9] G. Benedetti and S. Moroletti, "A genetic algorithm to search for optimal and suboptimal RNA secondary structures," *Biophysical Chemistry*, vol. 55, no. 3, pp. 253–259, 1995.
- [10] B. A. Shapiro, D. Bengali, W. Kasprzak, and J. C. Wu, "RNA folding pathway functional intermediates: their prediction and analysis," *Journal of Molecular Biology*, vol. 312, no. 1, pp. 27–44, 2001.
- [11] K. C. Wiese, A. A. Deschenes, and A. G. Hendriks, "RnaPredict—an evolutionary algorithm for RNA secondary structure prediction," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 5, no. 1, pp. 25–41, 2008.
- [12] K. C. Wiese and A. Hendriks, "Comparison of P-RnaPredict and mfold-algorithms for RNA secondary structure prediction," *Bioinformatics*, vol. 22, no. 8, pp. 934–942, 2006.
- [13] J. Yu, C. Zhang, Y. Liu, and X. Li, "Simulating the folding pathway of RNA secondary structure using the modified ant colony algorithm," *Journal of Bionic Engineering*, vol. 7, no. 4, pp. 382–389, 2010.
- [14] M. Geis and M. Middendorf, "A particle swarm optimizer for finding minimum free energy RNA secondary structures," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '07)*, pp. 1–8, Honolulu, Hawaii, USA, April 2007.
- [15] M. Neethling and A. P. Engelbrecht, "Determining RNA secondary structure using set-based particle swarm optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 1670–1677, Vancouver, Canada, July 2006.
- [16] Y. Liu, H. Dong, H. Zhang, G. Wang, Z. Li, and H. Chen, "Prediction of RNA secondary structure based on particle swarm optimization," *Chemical Research in Chinese Universities*, vol. 27, no. 1, pp. 108–112, 2011.
- [17] C. Xing, G. Wang, Y. Wang, W. Shen, Y. Liang, and Z. Ji, "A novel method for RNA secondary structure prediction," in *Proceedings of the 7th International Conference on Natural Computation (ICNC '11)*, pp. 1136–1140, Shanghai, China, July 2011.
- [18] C. Xing, G. Wang, Y. Wang, Y. Zhou, K. Wang, and L. Fan, "PSOfold: a metaheuristic for RNA folding," *Journal of Computational Information Systems*, vol. 8, no. 2, pp. 915–923, 2012.
- [19] X. Li and M. Yin, "Application of differential evolution algorithm on self-potential data," *PLoS ONE*, vol. 7, no. 12, Article ID e51199, 2012.
- [20] X. Li and M. Yin, "An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure," *Advances in Engineering Software*, vol. 55, pp. 10–31, 2013.
- [21] M. Dorigo, G. Di Caro, and L. M. Gambardella, "Ant algorithms for discrete optimization," *Artificial Life*, vol. 5, no. 2, pp. 137–172, 1999.
- [22] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 26, no. 1, pp. 29–41, 1996.

- [23] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.
- [24] R. Eberhart and J. Kennedy, "New optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, October 1995.
- [25] J. Kennedy and R. C. Eberhart, "Discrete binary version of the particle swarm algorithm," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 4104–4108, October 1997.

Research Article

Identification of Protein Pupylation Sites Using Bi-Profile Bayes Feature Extraction and Ensemble Learning

Xiaowei Zhao,^{1,2} Jian Zhang,¹ Qiao Ning,¹ Pingping Sun,¹ Zhiqiang Ma,¹ and Minghao Yin²

¹ College of Computer Science and Information Technology, Northeast Normal University, 2555 Jingyue Street, Changchun 130117, China

² Key Laboratory of Intelligent Information Processing of Jilin Universities, Northeast Normal University, Changchun 130117, China

Correspondence should be addressed to Zhiqiang Ma; zhiqiang.ma967@gmail.com and Minghao Yin; minghao.yin197@gmail.com

Received 13 July 2013; Accepted 1 August 2013

Academic Editor: William Guo

Copyright © 2013 Xiaowei Zhao 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.

Pupylation, one of the most important posttranslational modifications of proteins, typically takes place when prokaryotic ubiquitin-like protein (Pup) is attached to specific lysine residues on a target protein. Identification of pupylation substrates and their corresponding sites will facilitate the understanding of the molecular mechanism of pupylation. Comparing with the labor-intensive and time-consuming experiment approaches, computational prediction of pupylation sites is much desirable for their convenience and fast speed. In this study, a new bioinformatics tool named EnsemblePup was developed that used an ensemble of support vector machine classifiers to predict pupylation sites. The highlight of EnsemblePup was to utilize the Bi-profile Bayes feature extraction as the encoding scheme. The performance of EnsemblePup was measured with a sensitivity of 79.49%, a specificity of 82.35%, an accuracy of 85.43%, and a Matthews correlation coefficient of 0.617 using the 5-fold cross validation on the training dataset. When compared with other existing methods on a benchmark dataset, the EnsemblePup provided better predictive performance, with a sensitivity of 80.00%, a specificity of 83.33%, an accuracy of 82.00%, and a Matthews correlation coefficient of 0.629. The experimental results suggested that EnsemblePup presented here might be useful to identify and annotate potential pupylation sites in proteins of interest. A web server for predicting pupylation sites was developed.

1. Introduction

As the firstly identified posttranslational small protein modifier in prokaryotes, prokaryotic ubiquitin-like protein (Pup) in *Mycobacterium tuberculosis* (Mtb) is an important signal for the selective degradation of proteins [1]. Pup attaches to substrate lysine via isopeptide bonds in a manner reminiscent of ubiquitin (Ub) and ubiquitin-like modifier (Ubl) conjugation to proteins in eukaryotes [2]. Although pupylation and ubiquitylation are functional similarity, the enzymology of pupylation and ubiquitylation is different [3]. Generally, there are three-step reaction and three kinds of enzymes participating in the eukaryotic ubiquitylation process, including ubiquitin-activating enzymes, ubiquitin-conjugating enzymes, and ubiquitin ligases [4, 5], but only two-step reaction and two kinds of enzymes participating in the prokaryotic pupylation process. Firstly, the Pup-GGQ C-terminal is deamidated to -GGE by deamidase of Pup [6], and then the proteasome accessory factor A (PafA) attaches

the deamidated Pup to specific lysine residues of substrates [7].

Since identification of protein pupylation sites is of fundamental importance to understand the molecular mechanism of pupylation in biological systems, much interest has focused on this field and large-scale proteomics technology has been applied to identify pupylation proteins and pupylation sites [8–10]. However, the experimental determination of exact modified sites of pupylated substrates is labor intensive and time consuming, especially for large-scale data sets. In this regard, the computation approaches which could effectively and accurately identify the pupylation sites are urgently needed. Liu et al. had constructed the first online predictor, GPS-PUP, for the prediction of pupylation sites [11]. In their method, 127 experimentally identified pupylation sites in 109 prokaryotic proteins had been utilized as the training dataset, with an accuracy of 0.789 and an MCC of 0.286. However, there is significant room for improvement of the prediction performance.

TABLE 1: Number of pupylation and non-pupylation sites in each dataset.

	Pupylation proteins sequences	Positive sites	Negative sites
Dataset 1	153	183	2288
Dataset 2	109	127	1405

In this study, the prediction performance of pupylation sites has been improved by using a new encoding scheme, Bi-profile Bayes feature extraction (BPB), which has been widely used to deal with diverse prediction topics in the field of bioinformatics [12–15]. Since the new constructed pupylation sites dataset was highly imbalanced: the number of pupylation sites was much smaller than the number of nonpupylation sites, the ensemble learning method was adopted here to deal with the imbalanced data classification problem. The performance of EnsemblePup was measured with a sensitivity of 79.49%, a specificity of 82.35%, an accuracy of 85.43%, and a Matthews correlation coefficient of 0.617 using the 5-fold cross validation on the training dataset. When compared with other existing methods on a benchmark dataset, the EnsemblePup provided better predictive performance, with a sensitivity of 80.00%, a specificity of 83.33%, an accuracy of 82.00%, and a Matthews correlation coefficient of 0.629. The experimental results suggested that EnsemblePup presented here might be useful to identify and annotate potential pupylation sites in proteins of interest. A web server for predicting pupylation sites was developed and was available at <http://210.47.24.217:8080/EnsemblePup/>.

The organization of this paper is as follows. Section 2 introduces the dataset for establishing the predictor, the vector encoding schemes, and the proposed prediction model. Section 3 shows the experimental results, discusses the performance of the proposed predictor, and compares the proposed predictor with other methods. Finally Section 4 gives the conclusions.

2. Materials and Methods

2.1. Dataset. The pupylated proteins used in this study were extracted from PupDB [3]. Protein sequences with less than 50 amino acids were excluded because they may be just fragments [16, 17]. Protein sequences including nonstandard amino acids, such as “B,” “J,” “O,” “U,” “X,” and “Z” were excluded as well. As a result, there were 182 pupylated proteins with 215 known pupylation sites. After a homology-reducing screening procedure by using CD-HIT [18, 19] to remove those proteins that had 40% sequence identity to any other, we finally got 153 pupylated proteins with 183 positive sites, which constructed the nonredundant training dataset named as Dataset 1 in this study (see Supporting Information Text S1 available online at <http://dx.doi.org/10.1155/2013/283129>). In order to fairly compare our proposed method EnsemblePup with a previously developed method GPS-PUP, the dataset collected by Liu et al. [11] was also adopted here. We named it as Dataset 2 in this work, and the details of Dataset 2 were listed in Table 1.

Subsequently, similar to the development of other PTM site predictors [20, 21], the sliding window strategy was utilized to extract positive and negative samples. In order to ensure the peptides (sequence fragments) with a unified length, a nonexistent residue coded by “_” was used to fill the corresponding position. Peptides with pupylation lysine as the middle residue were regarded as positive samples, and the remaining peptides with nonpupylation lysine as the middle residue were regarded as negative samples.

2.2. Vector Encoding Schemes. In this study, the Bi-profile Bayes feature extraction (BPB) based encoding scheme was used. For details on this encoding scheme, readers are advised to refer to Shao et al. [14]. Briefly, let $S = s_1, s_2, \dots, s_n$ represent a sequence fragment, where s_j denotes one amino acid and n stands for the length of the sequence fragment. S belongs to two categories C_1 and C_{-1} , where C_1 and C_{-1} represent pupylation sites and nonpupylation sites, respectively. Then, a feature vector can be described as

$$\vec{P} = (p_1, p_2, \dots, p_n, p_{n+1}, \dots, p_{2n}), \quad (1)$$

where p_1, p_2, \dots, p_n represent the posterior probability of each amino acid at each position for the sequence fragment of pupylation sites (category C_1) and p_{n+1}, \dots, p_{2n} represent the posterior probability of each amino acid at each position for the sequence fragment of nonpupylation sites (category C_{-1}), which is the so-called Bi-profile. In this paper, the posterior probability is estimated by the occurrence of each amino acid at each position in the training datasets [14].

The binary encoding scheme was also carried out here to be compared with the BPB encoding scheme. As it is known to all, there are 20 types of amino acids in protein sequences, which are given as ACDEFGHIJKLMNOPQRSTUVWXYZ. Therefore, each amino acid is represented by a 20-dimensional binary vector; that is, A corresponds to (100000000000000000000000), C corresponds to (010000000000000000000000), and Y corresponds to (000000000000000000000001). For each sequence fragment with length n , the total dimension of the binary feature vector is $20 \times (n - 1)$, since the central amino acid is always K, which is not necessary to be considered.

2.3. Support Vector Machine Learning and Imbalanced Data. Support vector machine (SVM) is a popular machine learning algorithm mainly used in dealing with binary classification problems. SVM looks for a rule that best maps each member of training set to the correct classification [22, 23], and it has been widely used in bioinformatics community. In this paper, LIBSVM package [24] with radial basis kernels (RBF) is used, where the kernel width parameter γ represents how the samples are transformed to a high dimensional space. Grid search strategy based on 5-fold cross-validation is utilized to find the optimal parameters C and $\gamma \in \{2^{-7}, 2^{-6}, \dots, 2^8\}$, so that a total number of 256 grids are evaluated.

Since the training dataset was imbalanced, in which the number of pupylation sites was much smaller than the

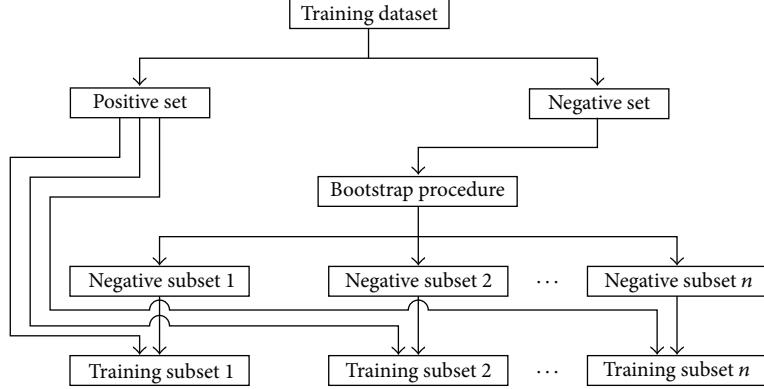


FIGURE 1: The bootstrap procedure for the imbalanced dataset.

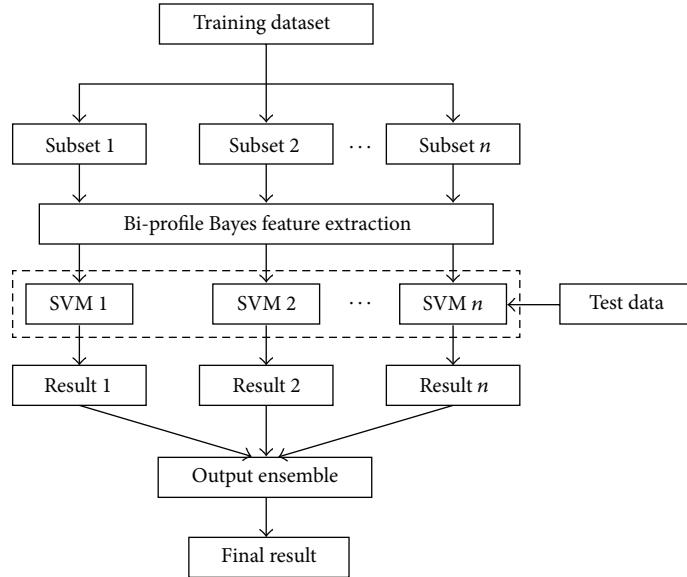


FIGURE 2: The entire schematic diagram for the prediction of pupylation sites.

number of nonpupylation sites, the bootstrap procedure was used to deal with this situation. As shown in Figure 1, we obtained n training subsets using the bootstrap procedure, where n represented the times of data sampling. In this study, the bootstrap procedure was implemented by WEKA package [25] and the parameter n was set as the ratio of the number of positive samples divided by the number of negative samples.

2.4. The Ensemble Model for Pupylation Sites Identification. Since ensemble learning methods have unique advantages in dealing with high-dimensional and complicated data, there is an increasing use of it in the field of bioinformatics [26–30]. In this study, the ensemble model was established by a collection of SVM classifiers, each was trained on a subset of the original training dataset (obtained by the bootstrap procedure in Figure 1). Figure 2 showed the entire schematic diagram for the prediction of pupylation sites. As shown in Figure 2, the final result was computed from the prediction result of the individual SVM classifier. For example, when

given a new unlabeled test data x , the j th SVM classifier returned a probability P_j of x belonged to the positive class, where $j = 1, 2, \dots, n$. The collection estimated probability was obtained by $P_{\text{Ensemble}} = (1/n) \sum_{j=1}^n P_j$.

2.5. Performance Assessment. In this study, 5-fold cross validation and jackknife cross validation tests were chosen for evaluating the proposed predictor. More details about these two methods can be found in two recent papers [31, 32]. In order to evaluate the proposed predictor, four measurements are used: sensitivity (Sn), specificity (Sp), accuracy (Ac), and Matthews correlation coefficient (MCC). They are defined by the following formulas:

$$\text{Sn} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

$$\text{Sp} = \frac{\text{TN}}{\text{TN} + \text{FP}},$$

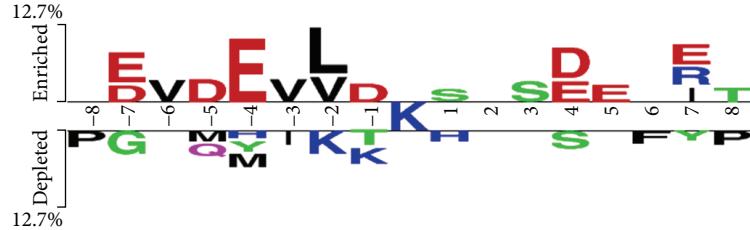


FIGURE 3: The Two-Sample-Logo of the position-specific residue composition surrounded the pupylation sites and nonpupylation sites. This logo was generated using the web server <http://www.twosamplelogo.org/> and only residues significantly enriched and depleted surrounding pupylation sites (t -test, $P < 0.1$) were shown.

TABLE 2: Results of the SVM prediction on Dataset 1.

SVM classifier	Sensitivity (%)	Specificity (%)	Accuracy (%)	MCC	A_{ROC}
BPB-SVM15	78.79	67.50	72.60	0.462	0.791
BPB-SVM17	75.76	72.50	73.97	0.480	0.807
BPB-SVM19	69.23	67.65	68.49	0.368	0.738
BPB-SVM21	65.79	80.00	72.60	0.463	0.741
BPB-SVM23	63.16	80.00	71.23	0.436	0.788
Binary-SVM17	55.50	51.52	53.42	0.065	0.527

$$Ac = \frac{TP + TN}{TP + TN + FP + FN},$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN) \times (TN + FP) \times (TP + FP) \times (TN + FN)}}, \quad (2)$$

where TP, TN, FP, and FN stand for the number of true positive, true negative, false positive, and false negative, respectively. In addition, the receiver operating characteristic (ROC) curves and the area under the curve (AUC) values are also carried out.

3. Results and Discussion

3.1. Determination of the Best Window Size. We firstly analyzed the position-specific propensities of the residues surrounding pupylation sites and nonpupylation sites using Two-Sample-Logo, which generated the graphical sequences logo for the relative frequency of the corresponding amino acid at each position around pupylation sites and nonpupylation sites. As shown in Figure 3, we found that the characteristics of the residues had significant differences between pupylation sites and nonpupylation sites. To encapsulate the position-specific propensities of residues for computational prediction, we established SVM prediction models of different lengths (represented as BPB-SVM15, BPB-SVM17, BPB-SVM19, BPB-SVM21, and BPB-SVM23) trained on a balanced training dataset (constructed by sampling a number of nonpupylation sites equal to the number of pupylation sites) using the Bi-profile Bayes feature extraction (BPB) method. As shown in Table 2, after a preliminary evaluation, the optimal window size was 17 in this paper (BPB-SVM17), with 8 residues located upstream and 8 residues located downstream of the pupylation sites in the protein sequence.

TABLE 3: The comparison of predictive performance between single SVM and ensemble of SVMs using the 5-fold cross validation on Dataset 1.

SVM classifier	Sensitivity (%)	Specificity (%)	Accuracy (%)	MCC	A_{ROC}
SinglePup	75.76	72.50	73.97	0.480	0.807
EnsemblePup	79.49	82.35	80.82	0.617	0.862

However, when the Bi-profile Bayes feature extraction encoding scheme was replaced by the binary encoding scheme of window size of 17 (represented as Binary-SVM17), the binary encoding scheme showed mediocre prediction performance, the prediction accuracy was 20.55% lower than that of the Bi-profile Bayes feature extraction encoding scheme, which indicated that the Bi-profile Bayes feature extraction encoding scheme has an advantage over the binary encoding scheme in predicting pupylation sites. Therefore, we adopted Bi-profile Bayes feature extraction encoding scheme in this study.

3.2. Comparison of EnsemblePup with a Single SVM Classifier. In order to enhance the prediction performance of the pupylation sites predictor, ensemble learning was used, and the final results were obtained by combining the outputs of different single SVM classifier. Here, we compared the performance of the ensemble of SVM classifiers with that of a single SVM classifier. All experiments were performed and reported the Sn, Sp, Ac, and MCC. The comparison results of the two prediction models by 5-fold cross validation test on the Dataset 1 were shown in Table 3 we can see the ensemble predictor got the accuracy of 80.82%, higher than the result obtained by using a single SVM classifier with 73.97%, and the AUC value was 0.55 higher than that of the a single

TABLE 4: The comparison of predictive performance between our method and GPS-PUP using the leave-one-out cross validation on Dataset 2.

Prediction method	Sensitivity (%)	Specificity (%)	Accuracy (%)	MCC	A_{ROC}
GPS-PUP	63.78	80.21	78.85	0.286	0.708
SinglePup	73.91	78.95	76.19	0.526	0.825
EnsemblePup	80.00	83.33	82.00	0.629	0.873

FIGURE 4: The prediction page of the EnsemblePup web server at <http://210.47.24.217:8080/EnsemblePup/>.

SVM classifier. In summary, the ensemble learning had an advantage in predicting pupylation sites.

3.3. Comparison of EnsemblePup with Other Methods. We have demonstrated that EnsemblePup could achieve a promising prediction performance in the 5-fold cross validation on Dataset 1. To objectively evaluate our proposed predictor, we further compared the EnsemblePup predictor with GPS-PUP [11]. Liu et al. searched PubMed with the keywords of “pupylation” and “prokaryotic ubiquitin” and collected 127 experimentally identified pupylation sites in 109 prokaryotic proteins; we named the data from Liu et al. as Dataset 2 in this work (the details were listed in Table 1). The compared results were shown in Table 4. As can be seen from the table, the EnsemblePup predictor proposed in this study obtained an accuracy of 82.00%, higher than the GPS-PUP predictor with the accuracy of 80.21%, and MCC of EnsemblePup was 0.343 greater than that of GPS-PUP.

3.4. The EnsemblePup Web Server. The EnsemblePup was implemented in Java and hosted on Windows platform. For the convenience of experimental scientists, we gave a step-by-step guide on how to use it to get the desired results as follows. (i) Open the web server at <http://210.47.24.217:8080/EnsemblePup/> and you can see the prediction page on your computer screen, as shown in Figure 4. You must input your email address since the prediction process may take a long time. (ii) Input your query protein sequence to the text box in Figure 4. Note that

the input protein sequence must be in the FASTA format. The FASTA format sequence consists of a single initial line beginning with a greater-than symbol (“>”), followed by lines of amino acid sequence. You can click on the “example and note” button to see the example protein sequence. (iii) Choose a threshold value in the drop-down list. For prediction with high confidence (less probability of false positive prediction), high threshold should be chosen. (iv) Click on the submit button to see the predicted result. For example, if you use the first sequence in the example page, the prediction results will be “>A0QNF6 K147 0.7450251 yes,” which means that the lysine on the position of 147 is a pupylation site with the probability of 0.7450251. Generally, it takes about 50 seconds to predict the pupylation site for a protein sequence shorter than 1000 amino acids before the predicted result appears.

4. Conclusion

Prediction of pupylation sites is important to understand the molecular mechanism of pupylation in biological systems. Though some researchers have focused on this problem, the accuracy of prediction is still not satisfied. In this study, we have presented a new predictor EnsemblePup for the prediction of pupylation sites based on Bi-profile Bayes feature extraction encoding scheme. Since the new constructed pupylation sites dataset was highly imbalanced: the number of pupylation sites was much smaller than the number of nonpupylation sites, the ensemble learning

method was adopted here to deal with the imbalanced data classification problem. The performance of EnsemblePup was measured with a sensitivity of 79.49%, a specificity of 82.35%, an accuracy of 85.43%, and a Matthews correlation coefficient of 0.617 using the 5-fold cross-validation on the training dataset. When compared with other existing methods on a benchmark dataset, the EnsemblePup provided better predictive performance, with a sensitivity of 80.00%, a specificity of 83.33%, an accuracy of 82.00%, and a Matthews correlation coefficient of 0.629. Experimental results have shown that our method is very promising and may be a useful supplement tool to existing methods. Due to the considerable performance, we have made EnsemblePup freely available as a web server. Although the results obtained here were very promising, further investigation was needed to further clarify the mechanism of pupylation process.

Acknowledgments

Jian Zhang and Qiao Ning collected data, wrote codes, and developed the web server. Zhiqiang Ma and Minghao Yin participated in the research design, method assessment, and preparation of the paper. Xiaowei Zhao directed the research and wrote the paper. All authors read and approved the final paper. This research is partially supported by the Science Foundation for Young Teachers of Northeast Normal University (no. 12QNJJ005) and the Natural Science Foundation of JiLin Province (nos. 20101506 and 20110104).

References

- [1] R. A. Festa, F. McAllister, M. J. Pearce et al., "Prokaryotic ubiquitin-like protein (Pup) proteome of *Mycobacterium tuberculosis*," *PLoS ONE*, vol. 5, no. 9, Article ID e8589, 2010.
- [2] K. H. Darwin, "Prokaryotic ubiquitin-like protein (Pup), proteasomes and pathogenesis," *Nature Reviews Microbiology*, vol. 7, no. 7, pp. 485–491, 2009.
- [3] C.-W. Tung, "PupDB: a database of pupylated proteins," *BMC Bioinformatics*, vol. 13, article 40, 2012.
- [4] R. L. Welchman, C. Gordon, and R. J. Mayer, "Ubiquitin and ubiquitin-like proteins as multifunctional signals," *Nature Reviews Molecular Cell Biology*, vol. 6, no. 8, pp. 599–609, 2005.
- [5] A. Hershko and A. Ciechanover, "The ubiquitin system," *Annual Review of Biochemistry*, vol. 67, pp. 425–479, 1998.
- [6] F. Striebel, F. Imkamp, M. Sutter, M. Steiner, A. Mamedov, and E. Weber-Ban, "Bacterial ubiquitin-like modifier Pup is deamidated and conjugated to substrates by distinct but homologous enzymes," *Nature Structural and Molecular Biology*, vol. 16, no. 6, pp. 647–651, 2009.
- [7] E. Guth, M. Thommen, and E. Weber-Ban, "Mycobacterial ubiquitin-like protein ligase PafA follows a two-step reaction pathway with a phosphorylated Pup intermediate," *Journal of Biological Chemistry*, vol. 286, no. 6, pp. 4412–4419, 2011.
- [8] F. A. Cerdá-Maira, F. McAllister, N. J. Bode, K. E. Burns, S. P. Gygi, and K. H. Darwin, "Reconstitution of the *Mycobacterium tuberculosis* pupylation pathway in *Escherichia coli*," *EMBO Reports*, vol. 12, no. 8, pp. 863–870, 2011.
- [9] C. Poulsen, Y. Akhter, A. H. Jeon et al., "Proteome-wide identification of mycobacterial pupylation targets," *Molecular Systems Biology*, vol. 6, article 386, 2010.
- [10] J. Watrous, K. Burns, W.-T. Liu et al., "Expansion of the mycobacterial pUPylome," *Molecular BioSystems*, vol. 6, no. 2, pp. 376–385, 2010.
- [11] Z. Liu, Q. Ma, J. Cao, X. Gao, J. Ren, and Y. Xue, "GPS-PUP: computational prediction of pupylation sites in prokaryotic proteins," *Molecular BioSystems*, vol. 7, no. 10, pp. 2737–2740, 2011.
- [12] L. J. Wee, D. Simarmata, Y.-W. Kam, L. F. Ng, and J. C. Tong, "SVM-based prediction of linear B-cell epitopes using Bayes Feature Extraction," *BMC Genomics*, vol. 11, article S21, supplement 4, 2010.
- [13] C. Jia, T. Liu, A. K. Chang, and Y. Zhai, "Prediction of mitochondrial proteins of malaria parasite using bi-profile Bayes feature extraction," *Biochimie*, vol. 93, no. 4, pp. 778–782, 2011.
- [14] J. Shao, D. Xu, S.-N. Tsai, Y. Wang, and S.-M. Ngai, "Computational identification of protein methylation sites through Bi-profile Bayes feature extraction," *PLoS ONE*, vol. 4, no. 3, Article ID e4920, 2009.
- [15] J. L. Shao, D. Xu, L. Hu et al., "Systematic analysis of human lysine acetylation proteins and accurate prediction of human lysine acetylation through bi-relative adapted binomial score Bayes feature representation," *Molecular BioSystems*, vol. 11, no. 8, pp. 2964–2973, 2012.
- [16] K.-C. Chou and H.-B. Shen, "Hum-PLoc: a novel ensemble classifier for predicting human protein subcellular localization," *Biochemical and Biophysical Research Communications*, vol. 347, no. 12, pp. 150–157, 2006.
- [17] K.-C. Chou and H.-B. Shen, "Large-scale plant protein subcellular location prediction," *Journal of Cellular Biochemistry*, vol. 100, no. 3, pp. 665–678, 2007.
- [18] Y. Huang, B. Niu, Y. Gao, L. Fu, and W. Li, "CD-HIT Suite: a web server for clustering and comparing biological sequences," *Bioinformatics*, vol. 26, no. 5, Article ID btq003, pp. 680–682, 2010.
- [19] W. Li and A. Godzik, "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, 2006.
- [20] L.-L. Hu, Z. Li, K. Wang et al., "Prediction and analysis of protein methylarginine and methyllysine based on Multisequence features," *Biopolymers*, vol. 96, no. 5, pp. 763–771, 2011.
- [21] X. Zhao, X. Li, Z. Ma, and M. Yin, "Prediction of lysine ubiquitylation with ensemble classifier and feature selection," *International Journal of Molecular Sciences*, vol. 12, no. 6, pp. 8347–8361, 2011.
- [22] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.
- [23] C.-W. Tung and S.-Y. Ho, "Computational identification of ubiquitylation sites from protein sequences," *BMC Bioinformatics*, vol. 9, article 310, 2008.
- [24] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machine," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 1, pp. 1–27, 2011.
- [25] H. Mark, F. Eibe, H. Geofry, P. Bernhard, and R. Peter, "The WEKA data mining software: an update," *SIGKDD Explorations*, vol. 11, no. 1, pp. 361–369, 2009.
- [26] J. Hu, Y. D. Yang, and D. Kihara, "EMD: an ensemble algorithm for discovering regulatory motifs in DNA sequences," *BMC Bioinformatics*, vol. 7, article 342, 2006.
- [27] M. Netzer, G. Millonig, M. Osl et al., "A new ensemble-based algorithm for identifying breath gas marker candidates in liver disease using ion molecule reaction mass spectrometry," *Bioinformatics*, vol. 25, no. 7, pp. 941–947, 2009.

- [28] T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, and Y. Saeys, "Robust biomarker identification for cancer diagnosis with ensemble feature selection methods," *Bioinformatics*, vol. 26, no. 3, Article ID btp630, pp. 392–398, 2009.
- [29] L. Deng, J. Guan, Q. Dong, and S. Zhou, "Prediction of protein-protein interaction sites using an ensemble method," *BMC Bioinformatics*, vol. 10, article 426, 2009.
- [30] W. Zhang, Y. Niu, Y. Xiong, M. Zhang, R. Yu, and J. Liu, "Computational prediction of conformational B-cell epitopes from antigen primary structures by ensemble learning," *PLoS ONE*, vol. 7, no. 8, Article ID e43575, 2012.
- [31] K. C. Chou and C. T. Zhang, "Review: prediction of protein structural classes," *Critical Reviews in Biochemistry and Molecular Biology*, vol. 30, no. 5, pp. 275–349, 1995.
- [32] K.-C. Chou and H.-B. Shen, "Cell-PLoc: a package of Web servers for predicting subcellular localization of proteins in various organisms," *Nature Protocols*, vol. 3, no. 2, pp. 153–162, 2008.

Research Article

Sliding Mode Control for NSVs with Input Constraint Using Neural Network and Disturbance Observer

Yan-long Zhou and Mou Chen

College of Automation Engineering, Nanjing Aeronautic and Astronautic University, Nanjing 210016, China

Correspondence should be addressed to Mou Chen; chenmou@nuaa.edu.cn

Received 10 May 2013; Accepted 15 July 2013

Academic Editor: William Guo

Copyright © 2013 Y.-l. Zhou and M. Chen. 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 sliding mode control (SMC) scheme is proposed for near space vehicles (NSVs) with strong nonlinearity, high coupling, parameter uncertainty, and unknown time-varying disturbance based on radial basis function neural networks (RBFNNs) and the nonlinear disturbance observer (NDO). Considering saturation characteristic of rudders, RBFNNs are constructed as a compensator to overcome the saturation nonlinearity. The stability of the closed-loop system is proved, and the tracking error as well as the disturbance observer error can converge to the origin through the Lyapunov analysis. Simulation results are presented to demonstrate the effectiveness of the proposed flight control scheme.

1. Introduction

The NSV [1–3] is a kind of vehicles and flights above 20 km from the ground, which can be divided into low-speed NSV and high-speed NSV. Among them, the high-speed NSV has some significant advantages such as rapid response ability, difficulty of detecting, and strong penetration ability. Thus, it has drawn much attention around the world in recent years. Since the NSV has the large flight envelop as well as the changeable flight environment and it is susceptible to the severely external time-varying disturbance, it is necessary to develop the robust control scheme for the NSV. However, strong nonlinearity, high coupling, parameter uncertainty, and unknown time-varying disturbance will further increase the design difficulty of the robust flight control [4]. To efficiently handle the system uncertainty and the suffering unknown external disturbance, NDO can be employed to design the robust control scheme. According to the known information of the studied system, NDO can estimate the compounded disturbance on line and feedback the estimate value to the control system. As a result, the disturbance cancellation is guaranteed to improve the performance and robustness of the closed-loop system. Therefore, the technology of NDO has been extensively studied. In [4–7], NDO was used to estimate the system uncertainty and the unknown

external disturbance, and it was successfully applied to the robust control design of inverted pendulum and air vehicles, respectively.

Saturation as a common input nonlinearity exists in a wide range of practical systems, such as the NSV. The existence of input saturation will degrade the control system performance and even leads to the system instability if it is ignored in the control design. On the other hand, the control design considering input saturation is a challenging problem for uncertain nonlinear systems. Several control schemes for nonlinear systems with input saturation have been proposed in recent years. In [8–10], neural network was proposed to approximate the input-output difference of the actuator, and a compensator was designed to overcome the saturation nonlinearity. In [11, 12], considering the smoothness of a hyperbolic tangent function, the robust adaptive control was proposed based on the backstepping approach via using the special property of a Nussbaum function to handle input saturation. However, the rudder saturation of the NSV needs to be further studied in the robust attitude control design.

For the nonlinear systems, many robust control schemes were studied [13–18] in which SMC is an effective method. By applying a discontinuous control signal to change the system state, the system is forced to slide along a predesigned sliding mode. It has higher robustness because of the insensitivity to

the parameter uncertainty. However, the control input needs to be changed according to the system states, which causes the system trajectory passing across the sliding mode repeatedly. Thus, there exists chattering problem in the system [19]. By designing a new switching function which is related to the first-order or higher-order derivative of the control input, the continuous dynamic sliding mode control law can be obtained through integration. Therefore, the system chattering can be effectively reduced. In [20], a robust dynamic sliding mode control law was proposed to control a multi-degree-of-freedom humanoid robot arm, and satisfactory control results had been obtained through simulation. In [21], higher-order sliding mode control and sliding mode differentiator were studied. In [22], a second-order dynamic sliding control scheme was proposed to overcome chattering problem for the NSV. However, the sliding mode control considering input saturation should be further developed for the NSV based on NDO.

Motivated by above discussion and analysis, the robust attitude control is developed for the NSV in the presence of the system uncertainty, the unknown time-varying disturbance, and the input saturation. To handle the compounded disturbance, the NDO is proposed. Since the rudders of the NSV have the saturation characteristic, RBFNNs are constructed as a compensator to overcome the saturation nonlinearity. The stability of the closed-loop system is proved, and the tracking error as well as the disturbance observer error can converge to the origin through the Lyapunov analysis. The organization of the paper is as follows. In Section 2, the dynamic sliding mode control is reviewed. The robust attitude control is investigated for the NSV by considering the system uncertainty, the unknown external disturbance, and the input saturation in Section 3, and simulation results are given to illustrate the effectiveness of the proposed robust attitude control scheme. Section 4 concludes the remark.

For the convenience of description, the following notations are required.

Notations. Throughout this paper, $|\bullet|$ stands for absolute value of each element of the vector; $|\bullet|^c$ denotes power operation for the absolute value of each element of the vector, where c is a power exponent; $\text{sgn}(\bullet)$ stands for sign function operation of each element of the vector; $\|\bullet\|$ represents the Euclidean norm (or Frobenius norm for a matrix); $\int_0^t (\bullet) dt$ stands for integral operation of each element of the vector; $\text{diag}(\bullet)$ represents a diagonal matrix constructed by the elements of the vector; $\text{diag}\{\text{sgn}(\bullet)\}$ denotes constructing a diagonal matrix after sign function operation of the vector. Specifically, given a matrix W and a vector χ , then we have

$$|\chi| = [| \chi_1 |, | \chi_2 |, \dots, | \chi_n |]^T,$$

$$|\chi|^c = [| \chi_1 |^c, | \chi_2 |^c, \dots, | \chi_n |^c]^T,$$

$$\text{sgn}(\chi) = [\text{sgn}(\chi_1), \text{sgn}(\chi_2), \dots, \text{sgn}(\chi_n)]^T,$$

$$\|\chi\| = \sqrt{\chi_1^2 + \chi_2^2 + \dots + \chi_n^2},$$

$$\begin{aligned} \|W\| &= \sqrt{\sum_{i=1}^n \sum_{j=1}^m w_{ij}^2}, W \in R^{n \times m}, \\ \int_0^t \chi dt &= \left[\int_0^t \chi_1 dt, \int_0^t \chi_2 dt, \dots, \int_0^t \chi_n dt \right]^T, \\ \text{diag}(\chi) &= \begin{bmatrix} \chi_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \chi_n \end{bmatrix} \in R^{n \times n}, \\ \text{diag}(\text{sgn}(\chi)) &= \begin{bmatrix} \text{sgn}(\chi_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \text{sgn}(\chi_n) \end{bmatrix} \in R^{n \times n}. \end{aligned} \quad (1)$$

2. Design of Dynamic Sliding Mode Control for MIMO System

In this paper, the dynamic sliding mode control scheme is used to the robust attitude control design of the NSV. For the convenience of control development, the dynamic sliding mode control scheme is described as follows according to [22, 23].

Consider that a class of multiinput and multioutput (MIMO) nonlinear affine systems is given by

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + g(x(t))u(t), \\ y(t) &= h(x(t)), \end{aligned} \quad (2)$$

where $x \in R^n$ is the state vector of the nonlinear system, $y \in R^m$ is the output vector, $u \in R^m$ is the control input vector, and $f(x) \in R^n$, $g(x) \in R^{n \times m}$, and $h(x) \in R^m$ are continuous functions which are related to the state vector x . Suppose that the sliding mode surface for the MIMO nonlinear system (2) can be written as $\sigma = [\sigma_1(x), \sigma_2(x), \dots, \sigma_m(x)]^T \in R^m$.

To analyze the nonlinear system (2), the following definitions and assumptions are required.

Definition 1 (see [24]). The gradient of the smooth scalar function $\sigma_i(x)$ with respect to the state vector x is defined as

$$\nabla \sigma_i(x) = \frac{\partial \sigma_i(x)}{\partial x} = \left[\frac{\partial \sigma_i(x)}{\partial x_1}, \frac{\partial \sigma_i(x)}{\partial x_2}, \dots, \frac{\partial \sigma_i(x)}{\partial x_n} \right]. \quad (3)$$

Definition 2 (see [24]). The Lie derivative of the smooth scalar function $\sigma_i(x)$ with respect to the vector field $f(x)$ is given by

$$L_f \sigma_i(x) = \nabla \sigma_i(x) f(x). \quad (4)$$

The multiple Lie derivative can be defined as the following recurrence relations:

$$\begin{aligned} L_f^0 \sigma_i(x) &= \sigma_i(x), \\ L_f^{k_i} \sigma_i(x) &= L_f L_f^{k_i-1} \sigma_i(x) = \nabla \left(L_f^{k_i-1} \sigma_i(x) \right) f(x), \end{aligned} \quad (5)$$

$$k_i = 1, 2, 3, \dots$$

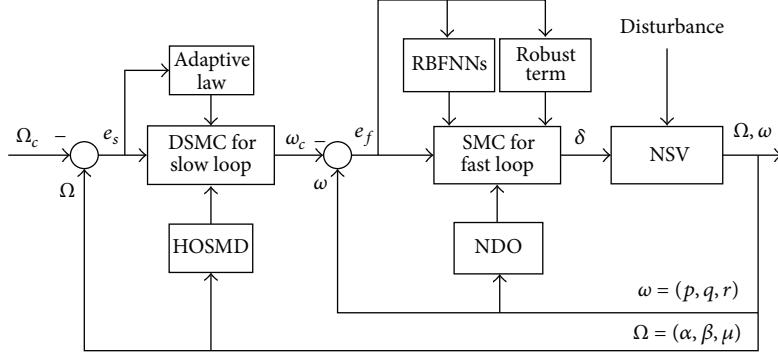


FIGURE 1: structure of the robust control scheme for the NSV.

Definition 3 (see [24]). If the Lie derivative of the smooth scalar function $\sigma_i(x)$ with respect to the vector field $f(x)$ satisfies the following two expressions:

$$\begin{aligned} L_f^{k_i} \sigma_i(x) &= 0, \quad k_i = 0, 1, \dots, r_i - 1, \\ L_f^{r_i} \sigma_i(x) &\neq 0, \end{aligned} \quad (6)$$

then the relative degree of $\sigma_i(x)$ with respect to $f(x)$ is r_i . Also, the relative degree of $\sigma = [\sigma_1(x), \sigma_2(x), \dots, \sigma_m(x)]^T$ with respect to $f(x)$ is r , where $r = \max\{r_1, r_2, \dots, r_m\}$.

Assumption 4 (see [22]). For the MIMO nonlinear system (2), the state x can converge to the origin when the system enters the sliding mode of $\sigma = 0$.

Assumption 5 (see [22]). For the MIMO nonlinear system (2), the relative degree vector of the sliding mode σ with respect to the input u is $[r_1, r_2, \dots, r_m]$. That is, the following equations are always held for all x in the domain:

$$\begin{aligned} \sum_{j=1}^m L_{g_j} L_f^{k_i} \sigma_i(x) &= 0, \quad k_i = 0, 1, \dots, r_i - 1, \\ \sum_{j=1}^m L_{g_j} L_f^{r_i-1} \sigma_i(x) &\neq 0, \end{aligned} \quad (7)$$

where g_j is the j th column vector of the matrix g , $i = 1, 2, \dots, m$.

Assumption 6 (see [22]). For the MIMO nonlinear system (2), the following matrix is invertible for all x in the domain:

$$\begin{bmatrix} L_{g_1} (L_f^{r_1-1} \sigma_1(x)) & L_{g_2} (L_f^{r_1-1} \sigma_1(x)) & \cdots & L_{g_m} (L_f^{r_1-1} \sigma_1(x)) \\ L_{g_1} (L_f^{r_2-1} \sigma_2(x)) & L_{g_2} (L_f^{r_2-1} \sigma_2(x)) & \cdots & L_{g_m} (L_f^{r_2-1} \sigma_2(x)) \\ \vdots & \vdots & \ddots & \vdots \\ L_{g_1} (L_f^{r_m-1} \sigma_m(x)) & L_{g_2} (L_f^{r_m-1} \sigma_m(x)) & \cdots & L_{g_m} (L_f^{r_m-1} \sigma_m(x)) \end{bmatrix} \in R^{m \times m}. \quad (8)$$

The derivative of $\sigma_i(x)$ is given by

$$\sigma_i^{(k_i)} = L_f^{k_i} \sigma_i(x), \quad k_i = 0, 1, \dots, r_i - 1, \quad (9)$$

$$\sigma_i^{(r_i)} = L_f^{r_i} \sigma_i(x) + \sum_{j=1}^m L_{g_j} L_f^{r_i-1} \sigma_i(x) u_j, \quad (10)$$

$$\begin{aligned} \sigma_i^{(r_i+1)} &= L_f^{r_i+1} \sigma_i(x) + \sum_{j=1}^m L_{g_j} L_f^{r_i} \sigma_i(x) u_j \\ &+ \sum_{l=1}^m L_{g_l} \left(\sum_{j=1}^m L_{g_j} L_f^{r_i-1} \sigma_i(x) u_j \right) u_l \\ &+ \sum_{j=1}^m L_f L_{g_j} L_f^{r_i-1} \sigma_i(x) u_j \\ &+ \sum_{j=1}^m L_{g_j} L_f^{r_i-1} \sigma_i(x) \dot{u}_j, \end{aligned} \quad (11)$$

where $i = 1, 2, \dots, m$.

Design a new sliding mode as

$$\vartheta_i = \sigma_i^{(r_i)} + c_{i,1} \sigma_i^{(r_i-1)} + \cdots + c_{i,r_i-1} \dot{\sigma}_i + c_{i,r_i} \sigma_i + c_{i,r_i+1}, \quad (12)$$

where $c_{i,j}$ ($i = 1, 2, \dots, m$, $j = 1, 2, \dots, r_i + 1$) must make the polynomial (12) Hurwitz stable.

The derivative of $\dot{\vartheta}_i$ is given by

$$\begin{aligned} \dot{\vartheta}_i &= \sigma_i^{(r_i+1)} + c_{i,1} \sigma_i^{(r_i)} + \cdots + c_{i,r_i-1} \ddot{\sigma}_i + c_{i,r_i} \dot{\sigma}_i \\ &= \sigma_i^{(r_i+1)} + \sum_{j=1}^{r_i} c_{i,j} \sigma_i^{(r_i+1-j)}. \end{aligned} \quad (13)$$

Considering (11) and (13), we obtain

$$\begin{aligned} \dot{\vartheta} &= A(x, u) + B(x) \dot{u} + C(x, u), \\ A(x, u) &= A_1(x) + A_2(x) u + A_3(x) u + A_4(x) u, \end{aligned} \quad (14)$$

where $\dot{\vartheta} = [\dot{\vartheta}_1, \dot{\vartheta}_2, \dots, \dot{\vartheta}_m]^T$, and we have

$$\begin{aligned}
A_1(x) &= \left[L_f^{r_1+1} \sigma_1(x), L_f^{r_2+1} \sigma_2(x), \dots, L_f^{r_m+1} \sigma_m(x) \right]^T \in R^m, \\
A_2(x) &= \begin{bmatrix} L_{g_1} L_f^{r_1} \sigma_1(x) & \cdots & L_{g_m} L_f^{r_1} \sigma_1(x) \\ \vdots & \ddots & \vdots \\ L_{g_1} L_f^{r_m} \sigma_m(x) & \cdots & L_{g_m} L_f^{r_m} \sigma_m(x) \end{bmatrix} \in R^{m \times m}, \\
A_3(x) &= \begin{bmatrix} L_f L_{g_1} L_f^{r_1-1} \sigma_1(x) & \cdots & L_f L_{g_m} L_f^{r_1-1} \sigma_1(x) \\ \vdots & \ddots & \vdots \\ L_f L_{g_1} L_f^{r_m-1} \sigma_m(x) & \cdots & L_f L_{g_m} L_f^{r_m-1} \sigma_m(x) \end{bmatrix} \in R^{m \times m}, \\
A_4(x) &= \begin{bmatrix} L_{g_1} \left(\sum_{j=1}^m L_{g_j} L_f^{r_1-1} \sigma_1(x) u_j \right) & \cdots & L_{g_m} \left(\sum_{j=1}^m L_{g_j} L_f^{r_1-1} \sigma_1(x) u_j \right) \\ \vdots & \ddots & \vdots \\ L_{g_1} \left(\sum_{j=1}^m L_{g_j} L_f^{r_m-1} \sigma_m(x) u_j \right) & \cdots & L_{g_m} \left(\sum_{j=1}^m L_{g_j} L_f^{r_m-1} \sigma_m(x) u_j \right) \end{bmatrix} \in R^{m \times m}, \\
B(x) &= \begin{bmatrix} L_{g_1} L_f^{r_1-1} \sigma_1(x) & \cdots & L_{g_m} L_f^{r_1-1} \sigma_1(x) \\ \vdots & \ddots & \vdots \\ L_{g_1} L_f^{r_m-1} \sigma_m(x) & \cdots & L_{g_m} L_f^{r_m-1} \sigma_m(x) \end{bmatrix} \in R^{m \times m}, \\
C(x, u) &= \left[\sum_{j=1}^{r_1} c_{1,j} \sigma_1^{(r_1+1-j)}, \sum_{j=1}^{r_2} c_{2,j} \sigma_2^{(r_2+1-j)}, \dots, \sum_{j=1}^{r_m} c_{m,j} \sigma_m^{(r_m+1-j)} \right]^T \in R^m.
\end{aligned} \tag{15}$$

To ensure the reaching condition of the sliding mode surface hold, the reaching law Φ_ϑ can take an exponential form which is described by

$$\Phi_\vartheta = -\bar{C}_1 \vartheta - \bar{C}_2 \operatorname{sgn}(\vartheta), \tag{16}$$

where $\bar{C}_i = \operatorname{diag}\{\bar{c}_{i,1}, \bar{c}_{i,2}, \dots, \bar{c}_{i,m}\} > 0$, $i = 1, 2$, are designed matrices.

Considering (14) and (16), we have

$$\dot{u} = B^{-1}(x) (\Phi_\vartheta - A(x, u) - C(x, u)). \tag{17}$$

Integration of (17) yields

$$u(t) = u(0) + \int_0^t (B^{-1}(x) (\Phi_\vartheta - A(x, u) - C(x, u))) dt. \tag{18}$$

Since the discontinuous term of the reaching law is taken into integration, we can obtain a continuous control input in

the whole time domain. Thus, the chattering of the system can be effectively reduced.

3. Design of Robust Attitude Control for the NSV

3.1. Attitude Dynamic Model for the NSV. Due to the fact that the change of attitude angular rate is faster than that of attitude angle, the attitude motion can be decomposed into two affine MIMO nonlinear systems [25] in accordance with singularly perturbed theory and time-scale separation principle:

$$\begin{aligned}
\dot{\Omega} &= f_s(\Omega) + g_s(\Omega) \omega + g_{s\delta}(\Omega) \delta(v(t)), \\
y_s &= \Omega, \\
\dot{\omega} &= f_f(\omega) + g_{f1} g_{f\delta}(\omega) \delta(v(t)), \\
y_f &= \omega,
\end{aligned} \tag{19}$$

where (19) are called slow-loop equation and fast-loop equation, respectively, and $\Omega = [\alpha, \beta, \mu]^T$ is the vector of attitude angles which are angle of attack, sideslip angle, and roll angle, $\omega = [p, q, r]^T$ is the vector of attitude angular rates which are roll angular rate, pitch angular rate, and yaw angular rate. In slow-loop equation, $f_s(\Omega) = [f_\alpha, f_\beta, f_\mu]^T$

is the state function vector, $g_s(\Omega)$ is the system matrix, and $g_{s\delta}(\Omega)$ is allocation matrix of rudders. In fast-loop equation, $f_f(\omega) = [f_p, f_q, f_r]^T$ is the state function vector, g_{f1} is the system matrix, $g_{f\delta}(\omega)$ is allocation matrix of rudders, $v(t) = [v_1, v_2, v_3, v_4, v_5]^T$ is the control input vector, and $\delta(v(t)) = [\delta_a, \delta_e, \delta_r, \delta_y, \delta_z]^T$ denotes the plant input vector subject to saturation nonlinearity which are ailerons, elevator, rudder, lateral deflection, and longitudinal deflection of the thrust vector control surface. The detailed expressions of corresponding terms in (19) can be found in [25].

The saturated function of $\delta_i(v_i)$ can be expressed as

$$\delta_i(v_i) = \text{sat}(v_i) = \begin{cases} \delta_{iM}, & v_i > \delta_{iM}, \\ v_i, & -\delta_{iM} \leq v_i \leq \delta_{iM}, \\ -\delta_{iM}, & v_i < -\delta_{iM}, \end{cases} \quad (20)$$

where δ_{iM} is the bound of $\delta_i(v_i)$, $i = 1, 2, 3, 4, 5$, $\delta(v) = [\delta_1(v_1), \delta_2(v_2), \delta_3(v_3), \delta_4(v_4), \delta_5(v_5)]^T$.

Considering that the control effect of Ω is mainly determined by $g_s\omega$ and the effect of $g_{s\delta}\delta$ on slow-loop system is very small, $g_{s\delta}\delta$ is taken as a part of the compounded disturbance D_s . On the other hand, there exists modeling error for the NSV, and the NSV is affected by the unknown time-varying external disturbance in flight. Thus, (19) can be rewritten as

$$\begin{aligned} \dot{\Omega} &= f_s + g_s\omega + D_s, \\ y_s &= \Omega, \end{aligned} \quad (21)$$

$$\begin{aligned} \dot{\omega} &= f_f + g_f\delta(v) + D_f, \\ y_f &= \omega, \end{aligned} \quad (22)$$

where $D_s = \Delta f_s + \Delta g_s\omega + g_{s\delta}\delta$ and $D_f = \Delta f_f + \Delta g_f\delta + d_f$ are the compounded disturbance of slowloop and fastloop, respectively; Δf_s , Δg_s , Δf_f , and Δg_f are modeling error of slowloop and fastloop, d_f is the time-varying external disturbance of fastloop, $g_f = g_{f1}g_{f\delta}$.

In this paper, the control objective is that the robust attitude control is designed to render y_s to follow the given desired trajectory Ω_c in the presence of the system uncertainty, unknown external disturbance, and input saturation. The external disturbance only acts on fast-loop system in the form of torque. On the other hand, the slow-loop system is only affected by the system uncertainty. The block diagram of the developed robust control scheme is shown in Figure 1.

To proceed the design of flight control scheme for the attitude motion (21) and (22) of the NSV, the following assumptions are required.

Assumption 7. For the attitude motion (21) and (22) of the NSV, the compounded disturbance of slow-loop system and its first-order derivative are bounded, that is, $|D_{s,i}| \leq \beta_{s,i}$, $\beta_{s,i} > 0$, $|\dot{D}_{s,i}| \leq \beta_{ds,i}$, $\beta_{ds,i} > 0$, $D_s = [D_{s,1}, D_{s,2}, D_{s,3}]^T$, $\dot{D}_s = [\dot{D}_{s,1}, \dot{D}_{s,2}, \dot{D}_{s,3}]^T$, $\beta_s = [\beta_{s,1}, \beta_{s,2}, \beta_{s,3}]^T$, $\beta_{ds} = [\beta_{ds,1}, \beta_{ds,2}, \beta_{ds,3}]^T$. At the same time, the compounded disturbance of fastloop is bounded, that is, $|D_{f,i}| \leq \beta_{f,i}$, $\beta_{f,i} > 0$, $D_f = [D_{f,1}, D_{f,2}, D_{f,3}]^T$.

Assumption 8 (see [22]). For the attitude motion (21) and (22) of the NSV, the desired attitude angle Ω_c and its second-order derivative are known, bounded, and continuous.

Assumption 9 (see [26]). For the attitude motion (21) and (22) of the NSV, generalized inverse matrices of the control gain matrices g_s and g_f exist.

Remark 10. For a practical system, the external disturbance is bounded since it actually exists. In addition, the parameter uncertainty $\Delta(\bullet)$ is the function of system state Ω or ω . Thus, we assume that the boundaries of D_s , \dot{D}_s , and D_f satisfy $|D_{s,i}| \leq \beta_{s,i}$, $|\dot{D}_{s,i}| \leq \beta_{ds,i}$, and $|D_{f,i}| \leq \beta_{f,i}$, respectively. Accordingly, the Assumption 7 is reasonable.

3.2. Design of Slow-Loop System for the NSV. In slow-loop system, only the system uncertainty is considered. The adaptive control approach is used to estimate the upper bound of the system uncertainty. The robust control scheme of slow-loop system is proposed based on the dynamic sliding mode control. To obtain the corresponding function's derivative, the higher-order sliding mode differentiator (HOSMD) [27–29] is employed because of the ability of its arbitrary approximation.

Theorem 11. Consider slow-loop system (21) of the NSV satisfied Assumptions 7–9. Under sliding modes (23) and (24), exponential reaching law (25), parameter updated law (26), and the dynamic sliding mode control law (27), the tracking error of slow-loop system asymptotically converges to the origin:

$$\sigma_{s1} = e_s + \int_0^t A_{s1}e_s dt, \quad (23)$$

$$\sigma_{s2} = \dot{\sigma}_{s1} + A_{s2}\sigma_{s1}, \quad (24)$$

$$\Phi_{\sigma_{s2}} = -B_{s1}\sigma_{s2} - B_{s2}\text{sgn}(\sigma_{s2}), \quad (25)$$

$$\dot{\hat{\beta}}_{ds} = B_{s3}|\sigma_{s2}|, \quad (26)$$

$$\omega_c = g_s^{-1} \int_0^t (\psi_1 + \psi_2) dt, \quad (27)$$

$$\psi_1 = -\dot{f}_s + \ddot{\Omega}_c - (A_{s1} + A_{s2})\dot{e}_s - A_{s2}A_{s1}e_s,$$

$$\psi_2 = -B_{s1}\sigma_{s2} - B_{s2}\text{sgn}(\sigma_{s2}) - \text{diag}\{\text{sgn}(\sigma_{s2})\}\hat{\beta}_{ds},$$

where $e_s = \Omega - \Omega_c$ is the tracking error of slowloop, $A_{s1} = \text{diag}\{a_{s1,1}, a_{s1,2}, a_{s1,3}\} > 0$, $A_{s2} = \text{diag}\{a_{s2,1}, a_{s2,2}, a_{s2,3}\} > 0$, $B_{s1} = \text{diag}\{b_{s1,1}, b_{s1,2}, b_{s1,3}\} > 0$, $B_{s2} = \text{diag}\{b_{s2,1}, b_{s2,2}, b_{s2,3}\} > 0$, and $B_{s3} = \text{diag}\{b_{s3,1}, b_{s3,2}, b_{s3,3}\} > 0$ are designed matrices, and $\hat{\beta}_{ds}$ is the estimate value of β_{ds} .

Proof. Consider the Lyapunov function as

$$V_s = \frac{1}{2}\sigma_{s2}^T\sigma_{s2} + \frac{1}{2}\hat{\beta}_{ds}^T B_{s3}^{-1}\hat{\beta}_{ds}, \quad (28)$$

where $\tilde{\beta}_{ds} = \beta_{ds} - \hat{\beta}_{ds}$ and $\dot{\hat{\beta}}_{ds} = \dot{\beta}_{ds} - \hat{\dot{\beta}}_{ds} = -\dot{\hat{\beta}}_{ds}$.

Differentiating (24) and considering (21), (23), and (27), we have

$$\begin{aligned}\dot{\sigma}_{s2} &= \dot{f}_s + \psi_1 + \psi_2 + \dot{D}_s - \ddot{\Omega}_c + A_{s1}\dot{e}_s + A_{s2}\dot{e}_s + A_{s2}A_{s1}e_s \\ &= \dot{D}_s - B_{s1}\sigma_{s2} - B_{s2}\operatorname{sgn}(\sigma_{s2}) - \operatorname{diag}\{\operatorname{sgn}(\sigma_{s2})\}\tilde{\beta}_{ds}.\end{aligned}\quad (29)$$

Invoking (26) and (29), the time derivative of V_s is given by

$$\begin{aligned}\dot{V}_s &= \sigma_{s2}^T\dot{\sigma}_{s2} - \tilde{\beta}_{ds}^T B_{s3}^{-1}\dot{\beta}_{ds} \\ &= \sigma_{s2}^T(\dot{D}_s - B_{s1}\sigma_{s2} - B_{s2}\operatorname{sgn}(\sigma_{s2})) \\ &\quad - \operatorname{diag}\{\operatorname{sgn}(\sigma_{s2})\}\tilde{\beta}_{ds} - \tilde{\beta}_{ds}^T|\sigma_{s2}| \\ &\leq -\sigma_{s2}^T B_{s1}\sigma_{s2} - \sigma_{s2}^T B_{s2}\operatorname{sgn}(\sigma_{s2}) + |\sigma_{s2}|^T\beta_{ds} \\ &\quad - |\sigma_{s2}|^T\tilde{\beta}_{ds} - \tilde{\beta}_{ds}^T|\sigma_{s2}| \\ &= -\sum_{i=1}^3 b_{s1,i}\sigma_{s2,i}^2 - \sum_{i=1}^3 b_{s2,i}|\sigma_{s2,i}|,\end{aligned}\quad (30)$$

where $\sigma_{s2} = [\sigma_{s2,1}, \sigma_{s2,2}, \sigma_{s2,3}]$.

It is obvious that if $\sigma_{s2} \neq 0$, then $\dot{V}_s < 0$. Thus, the sliding mode σ_{s2} satisfies the reaching condition, and σ_{s2} can asymptotically converge to the origin. From (23) and (24), we can know that the sliding mode σ_{s1} and the tracking error e_s converge to the origin when σ_{s2} converges to the origin. This concludes the proof. \square

Remark 12. ω_c in (27) is the controller of slow-loop system as well as the desired input of fast-loop system.

Remark 13. In the control law of the slow-loop system, the differential terms, such as \dot{f}_s , \dot{e}_s , and $\dot{\sigma}_{s1}$, are difficult to obtain through derivative method. Thus, HOSMD is employed to get the corresponding derivatives. The form of n th-order HOSMD is given by [27–29]

$$\begin{aligned}\dot{z}_0 &= \zeta_0 = -\lambda_0|z_0 - f(t)|^{n/(n+1)}\operatorname{sgn}(z_0 - f(t)) + z_1 \\ &\vdots \\ \dot{z}_i &= \zeta_i = -\lambda_i|z_i - \zeta_{i-1}|^{(n-i)/(n-i+1)}\operatorname{sgn}(z_i - \zeta_{i-1}) + z_{i+1} \\ &\vdots \\ \dot{z}_{n-1} &= \zeta_{n-1} = -\lambda_{n-1}|z_{n-1} - \zeta_{n-2}|^{(1/2)}\operatorname{sgn}(z_{n-1} - \zeta_{n-2}) + z_n \\ \dot{z}_n &= -\lambda_n\operatorname{sgn}(z_n - \zeta_{n-1}),\end{aligned}\quad (31)$$

where z_i and ζ_j are states of the system (31), $\lambda_0, \lambda_1, \dots, \lambda_n$ are designed parameters, and $f(t)$ is the known function. The aim of HOSMD is to make ζ_j approximate the differential term $f(t)^{(j+1)}$ to arbitrary any accuracy, $i = 0, 1, \dots, n$, $j = 0, 1, \dots, n-1$.

3.3. Design of Fast-Loop System for the NSV. In fast-loop system, the compounded disturbance is subjected to the system uncertainty as well as the unknown time-varying external disturbance. In general, the order of magnitude of the external disturbance is much larger than that of the system uncertainty. If we still use the adaptive scheme to handle the compounded disturbance of fast-loop system, the system trajectory will pass through the equilibrium point repeatedly. Thus, the designed disturbance cancellation can not accurately compensate the suffering disturbance and leads to the performance of the control system degeneracy. Here, the nonlinear disturbance observer is proposed to estimate the compounded disturbance. However, the robust control design of fast-loop system is different from that of slow-loop system based on DSMC and NDO. The reason is that the output derivative of NDO will be contained in the control law and it is more complex and difficult to prove the closed-loop system stability as well as the boundedness of disturbance observer error. Therefore, the traditional sliding mode control is used in fast-loop system, and the double power reaching law is designed to reduce the chattering of the system.

Considering that rudders are subject to saturation nonlinearity in fast-loop system and RBFNNs can smoothly approximate any continuous function over the compact set to arbitrary accuracy [30, 31]. Here, RBFNNs are constructed as a compensator to overcome the saturation nonlinearity. Specifically, we employ RBFNNs to estimate the limited part exceeding the bound of saturation and use it to design the robust control law. As a result, the actuator can drop out the saturation nonlinearity.

Considering (20), we obtain

$$\delta(v) = \operatorname{sat}(v) = v + \xi(v), \quad (32)$$

where $\xi(v)$ is the limited part exceeding the actuator.

Using RBFNNs to approximate $\xi(v)$, we have

$$\xi(v) = W_\xi^{*T} s_\xi(z) + \varepsilon_\xi, \quad (33)$$

where $W_\xi^* \in R^{l \times 5}$ are the optimal weights and satisfy $\|W_\xi^*\| \leq W_M$, l is the number of nodes; $z = [\omega_c, e_f]^T$ is the input vector of RBFNNs; $\varepsilon_\xi = [\varepsilon_{\xi,1}, \varepsilon_{\xi,2}, \varepsilon_{\xi,3}, \varepsilon_{\xi,4}, \varepsilon_{\xi,5}]^T$ is the smallest approximated error, and ε_ξ can be arbitrarily small through turning weights and nodes; suppose $|\varepsilon_{\xi,i}| \leq \varepsilon_{M\xi,i}$ and $\varepsilon_{M\xi,i} > 0$ in this paper; $s_\xi(z) = [s_{\xi,1}, s_{\xi,2}, \dots, s_{\xi,l}]^T$ is the radial basis function vector, and $s_{\xi,k}$ has the form as follows:

$$s_{\xi,k} = \exp\left(-\frac{\|z - c_k\|^2}{2b_k^2}\right), \quad (34)$$

where c_k and b_k are the center and width of the neural cell of the k th hidden layer, $k = 1, 2, \dots, l$.

Design the control law as

$$v = v_0 - v_\xi + v_r, \quad (35)$$

where v_0 is the controller by neglecting saturation nonlinearity, $v_\xi = \widehat{W}_\xi^T s_\xi(z)$ is the output of RBFNNs, and v_r is the robust term.

Substituting (33) and (35) into (32), we obtain

$$\begin{aligned}\delta(v) &= v_0 - v_\xi + v_r + W_\xi^{*T} s_\xi(z) + \varepsilon_\xi \\ &= v_0 + \widetilde{W}_\xi^T s_\xi(z) + v_r + \varepsilon_\xi,\end{aligned}\quad (36)$$

where $\widetilde{W}_\xi = W_\xi^* - \widehat{W}_\xi$.

Considering the unknown compounded disturbance D_f of the fast-loop system (22), it cannot be directly used in the design of fast-loop control scheme. To estimate the compounded disturbance, the nonlinear disturbance observer is presented below.

An auxiliary variable is given by

$$\sigma = \eta - \omega. \quad (37)$$

And the derivative of η is

$$\dot{\eta} = -K\sigma - \text{diag}\{\text{sgn}(\sigma)\} \widehat{\beta}_f + f_f + g_f \delta(v) - K\sigma_f, \quad (38)$$

where $K = \text{diag}\{k_1, k_2, k_3\} > 0$ is a designed matrix, $\widehat{\beta}_f = [\widehat{\beta}_{f,1}, \widehat{\beta}_{f,2}, \widehat{\beta}_{f,3}]^T$ is the estimate value of $\beta_f = [\beta_{f,1}, \beta_{f,2}, \beta_{f,3}]^T$, $\sigma = [\sigma_1, \sigma_2, \sigma_3]^T$ is the auxiliary variable, and $\sigma_f = [\sigma_{f,1}, \sigma_{f,2}, \sigma_{f,3}]^T$ is the sliding mode surface of fast-loop system, which will be introduced later.

The estimate value of D_f can be written as

$$\begin{aligned}\widehat{D}_f &= -K\sigma - \text{diag}\{\text{sgn}(\sigma)\} \widehat{\beta}_f + \text{diag}\{\text{sgn}(\sigma_f)\} |\widehat{\beta}_f| \\ &\quad + \text{diag}\{\text{sgn}(\sigma_f)\} \widehat{\beta}_f.\end{aligned}\quad (39)$$

Differentiating (37) and considering (22) and (38), we have

$$\dot{\sigma} = \dot{\eta} - \dot{\omega} = -K\sigma - \text{diag}\{\text{sgn}(\sigma)\} \widehat{\beta}_f - K\sigma_f - D_f. \quad (40)$$

The disturbance approximation error is given by

$$\widetilde{D}_f = D_f - \widehat{D}_f. \quad (41)$$

Substituting (39) and (40) into (41), we obtain

$$\begin{aligned}\widetilde{D}_f &= K\sigma + \text{diag}\{\text{sgn}(\sigma)\} \widehat{\beta}_f \\ &\quad + D_f - \text{diag}\{\text{sgn}(\sigma_f)\} |\widehat{\beta}_f| \\ &\quad - \text{diag}\{\text{sgn}(\sigma_f)\} \widehat{\beta}_f \\ &= -\dot{\sigma} - K\sigma_f - \text{diag}\{\text{sgn}(\sigma_f)\} |\widehat{\beta}_f| \\ &\quad - \text{diag}\{\text{sgn}(\sigma_f)\} \widehat{\beta}_f.\end{aligned}\quad (42)$$

Theorem 14. Consider that fast-loop system (22) of the NSV satisfied Assumptions 7–9. The sliding mode is designed as (43), and the reaching law is chosen as (44). The nonlinear disturbance observer is proposed as (39), and the parameter adaptation law is designed as (45). RBFNNs are constructed as a compensator to overcome the saturation nonlinearity, and the updated law of the neural network weight is chosen as

(46). Under the control law (47), the tracking error of fast-loop system asymptotically converges to the origin, and the error of the disturbance observer also converges to the origin:

$$\sigma_f = e_f + \int_0^t A_f e_f dt, \quad (43)$$

$$\begin{aligned}\Phi_{\sigma_f} &= -B_{f1} \text{diag}\{\text{sgn}(\sigma_f)\} |\sigma_f|^{c_1} \\ &\quad - B_{f2} \text{diag}\{\text{sgn}(\sigma_f)\} |\sigma_f|^{c_2},\end{aligned}\quad (44)$$

$$\dot{\widehat{\beta}}_f = \Gamma_\beta (|\sigma| + |\sigma_f|), \quad (45)$$

$$\dot{\widetilde{W}}_\xi = \Gamma_\xi s_\xi(z) \sigma_f^T g_f - \gamma \|\sigma_f^T g_f\| \Gamma_\xi \widehat{W}_\xi, \quad (46)$$

$$v = v_0 - v_\xi + v_r,$$

$$\begin{aligned}v_0 &= g_f^T (g_f g_f^T)^{-1} (-f_f + \dot{\omega}_c - A_f e_f - \widehat{D}_f \\ &\quad - B_{f1} \text{diag}\{\text{sgn}(\sigma_f)\} |\sigma_f|^{c_1} \\ &\quad - B_{f2} \text{diag}\{\text{sgn}(\sigma_f)\} |\sigma_f|^{c_2}),\end{aligned}\quad (47)$$

$$v_\xi = \widehat{W}_\xi^T s_\xi(z),$$

$$v_r = -K_r \text{sgn}(g_f^T \sigma_f),$$

where $e_f = \omega - \omega_c$ is the tracking error of fast-loop; $A_f = \text{diag}\{a_{f,1}, a_{f,2}, a_{f,3}\} > 0$, $B_{f1} = \text{diag}\{b_{f,1,1}, b_{f,1,2}, b_{f,1,3}\} > 0$, $B_{f2} = \text{diag}\{b_{f,2,1}, b_{f,2,2}, b_{f,2,3}\} > 0$, $\Gamma_\beta = \Gamma_\beta^T > 0$, $\Gamma_\xi = \Gamma_\xi^T > 0$, and $K_r = \text{diag}\{k_{r,1}, k_{r,2}, k_{r,3}, k_{r,4}, k_{r,5}\}$ are designed matrices, and $k_{r,i} - \varepsilon_{M\xi,i} - W_M^2 \gamma / 4 > 0$; $\gamma > 0$, $c_1 > 1$ and $0 < c_2 < 1$ are designed parameters.

Proof. Consider the Lyapunov function as

$$V_f = \frac{1}{2} \sigma_f^T \sigma_f + \frac{1}{2} \sigma^T \sigma + \frac{1}{2} \widetilde{\beta}_f^T \Gamma_\beta^{-1} \widetilde{\beta}_f + \frac{1}{2} \text{tr}(\widetilde{W}_\xi^T \Gamma_\xi^{-1} \widetilde{W}_\xi), \quad (48)$$

where $\widetilde{\beta}_f = \beta_f - \widehat{\beta}_f$, $\widetilde{W}_\xi = W_\xi^* - \widehat{W}_\xi$, and $\dot{\widetilde{\beta}}_f = \dot{\beta}_f - \dot{\widehat{\beta}}_f = -\dot{\widehat{\beta}}_f$ and $\dot{\widetilde{W}}_\xi = \dot{W}_\xi^* - \dot{\widehat{W}}_\xi = -\dot{\widehat{W}}_\xi$.

Differentiating (43) and considering (22), we obtain

$$\dot{\sigma}_f = f_f + g_f \delta(v) + D_f - \dot{\omega}_c + A_f e_f. \quad (49)$$

Substituting (36), (41), and (47) into (49), we have

$$\begin{aligned}\dot{\sigma}_f &= f_f + D_f - \dot{\omega}_c + A_f e_f \\ &\quad + g_f (v_0 + \widetilde{W}_\xi^T s_\xi(z) + v_r + \varepsilon_\xi) \\ &= \widetilde{D}_f - B_{f1} \text{diag}\{\text{sgn}(\sigma_f)\} |\sigma_f|^{c_1} \\ &\quad - B_{f2} \text{diag}\{\text{sgn}(\sigma_f)\} |\sigma_f|^{c_2} \\ &\quad + g_f \widetilde{W}_\xi^T s_\xi(z) - g_f K_r \text{sgn}(g_f^T \sigma_f) + g_f \varepsilon_\xi.\end{aligned}\quad (50)$$

Invoking (40) and (50), the time derivative of V_f is given by

$$\begin{aligned}\dot{V}_f &= \sigma_f^T \dot{\sigma}_f + \sigma^T \dot{\sigma} - \tilde{\beta}_f^T \Gamma_\beta^{-1} \dot{\tilde{\beta}}_f - \text{tr} \left(\widetilde{W}_\xi^T \Gamma_\xi^{-1} \dot{\widetilde{W}}_\xi \right) \\ &= \sigma_f^T \widetilde{D}_f - \sum_{i=1}^3 b_{f1,i} |\sigma_{f,i}|^{1+c_1} - \sum_{i=1}^3 b_{f2,i} |\sigma_{f,i}|^{1+c_2} \\ &\quad + \sigma_f^T g_f \widetilde{W}_\xi^T s_\xi(z) - \sigma_f^T g_f K_r \text{sgn} \left(g_f^T \sigma_f \right) \\ &\quad + \sigma_f^T g_f \varepsilon_\xi - \sum_{i=1}^3 k_i \sigma_i^2 - |\sigma|^T \tilde{\beta}_f - \sigma^T K \sigma_f \\ &\quad - \sigma^T D_f - \tilde{\beta}_f^T \Gamma_\beta^{-1} \dot{\tilde{\beta}}_f - \text{tr} \left(\widetilde{W}_\xi^T \Gamma_\xi^{-1} \dot{\widetilde{W}}_\xi \right).\end{aligned}\quad (51)$$

Considering (42), we obtain

$$\begin{aligned}\sigma_f^T \widetilde{D}_f &= \sigma_f^T K \sigma + \sigma_f^T \text{diag} \{ \text{sgn}(\sigma) \} \tilde{\beta}_f \\ &\quad + \sigma_f^T D_f - |\sigma_f|^T |\tilde{\beta}_f| - |\sigma_f|^T \tilde{\beta}_f \\ &\leq \sigma_f^T K \sigma + |\sigma_f|^T |\tilde{\beta}_f| + |\sigma_f|^T \beta_f \\ &\quad - |\sigma_f|^T |\tilde{\beta}_f| - |\sigma_f|^T \tilde{\beta}_f \\ &= \sigma_f^T K \sigma + |\sigma_f|^T \tilde{\beta}_f.\end{aligned}\quad (52)$$

Substituting (52) into (51) yields

$$\begin{aligned}\dot{V}_f &\leq - \sum_{i=1}^3 b_{f1,i} |\sigma_{f,i}|^{1+c_1} - \sum_{i=1}^3 b_{f2,i} |\sigma_{f,i}|^{1+c_2} \\ &\quad - \sum_{i=1}^3 k_i \sigma_i^2 - \sigma_f^T g_f K_r \text{sgn} \left(g_f^T \sigma_f \right) + \sigma_f^T g_f \varepsilon_\xi \\ &\quad + |\sigma_f|^T \tilde{\beta}_f - |\sigma|^T \tilde{\beta}_f + |\sigma|^T \beta_f + \sigma_f^T g_f \widetilde{W}_\xi^T s_\xi(z) \\ &\quad - \tilde{\beta}_f^T \Gamma_\beta^{-1} \dot{\tilde{\beta}}_f - \text{tr} \left(\widetilde{W}_\xi^T \Gamma_\xi^{-1} \dot{\widetilde{W}}_\xi \right) \\ &\leq - \sum_{i=1}^3 b_{f1,i} |\sigma_{f,i}|^{1+c_1} - \sum_{i=1}^3 b_{f2,i} |\sigma_{f,i}|^{1+c_2} \\ &\quad - \sum_{i=1}^3 k_i \sigma_i^2 - \sum_{i=1}^3 (k_{r,i} - \varepsilon_{M\xi,i}) |\sigma_f^T g_{f,i}| + |\sigma_f|^T \tilde{\beta}_f \\ &\quad + |\sigma|^T \tilde{\beta}_f - \tilde{\beta}_f^T \Gamma_\beta^{-1} \dot{\tilde{\beta}}_f \\ &\quad - \text{tr} \left(\widetilde{W}_\xi^T \Gamma_\xi^{-1} \dot{\widetilde{W}}_\xi - \widetilde{W}_\xi^T s_\xi(z) \sigma_f^T g_f \right),\end{aligned}\quad (53)$$

where $g_{f,i}$ is the i th column vector of the matrix g_f . \square

Considering parameter updated laws (45) and (46), we have

$$\begin{aligned}\dot{V}_f &\leq - \sum_{i=1}^3 b_{f1,i} |\sigma_{f,i}|^{1+c_1} - \sum_{i=1}^3 b_{f2,i} |\sigma_{f,i}|^{1+c_2} - \sum_{i=1}^3 k_i \sigma_i^2 \\ &\quad - \sum_{i=1}^3 (k_{r,i} - \varepsilon_{M\xi,i}) |\sigma_f^T g_{f,i}| + \text{tr} \left(\gamma \|\sigma_f^T g_f\| \widetilde{W}_\xi^T \widetilde{W}_\xi \right).\end{aligned}\quad (54)$$

Considering the following fact that

$$\text{tr} \left(\widetilde{W}_\xi^T (W_\xi^* - \widetilde{W}_\xi) \right) \leq \|\widetilde{W}_\xi\| \|W_\xi^*\| - \|\widetilde{W}_\xi\|^2,\quad (55)$$

we have

$$\begin{aligned}\text{tr} \left(\gamma \|\sigma_f^T g_f\| \widetilde{W}_\xi^T \widetilde{W}_\xi \right) &= \gamma \|\sigma_f^T g_f\| \text{tr} \left(\widetilde{W}_\xi^T (W_\xi^* - \widetilde{W}_\xi) \right) \\ &\leq \gamma \|\sigma_f^T g_f\| (\|\widetilde{W}_\xi\| W_M - \|\widetilde{W}_\xi\|^2) \\ &= \gamma \|\sigma_f^T g_f\| \left(- \left(\|\widetilde{W}_\xi\| - \frac{1}{2} W_M \right)^2 + \frac{1}{4} W_M^2 \right).\end{aligned}\quad (56)$$

Substituting (56) into (54) yields

$$\begin{aligned}\dot{V}_f &\leq - \sum_{i=1}^3 b_{f1,i} |\sigma_{f,i}|^{1+c_1} - \sum_{i=1}^3 b_{f2,i} |\sigma_{f,i}|^{1+c_2} \\ &\quad - \sum_{i=1}^3 k_i \sigma_i^2 - \sum_{i=1}^3 (k_{r,i} - \varepsilon_{M\xi,i}) |\sigma_f^T g_{f,i}| \\ &\quad + \frac{1}{4} W_M^2 \|\sigma_f^T g_f\| - \gamma \|\sigma_f^T g_f\| \left(\|\widetilde{W}_\xi\| - \frac{1}{2} W_M \right)^2 \\ &\leq - \sum_{i=1}^3 b_{f1,i} |\sigma_{f,i}|^{1+c_1} - \sum_{i=1}^3 b_{f2,i} |\sigma_{f,i}|^{1+c_2} \\ &\quad - \sum_{i=1}^3 k_i \sigma_i^2 - \sum_{i=1}^3 \left(k_{r,i} - \varepsilon_{M\xi,i} - \frac{1}{4} W_M^2 \gamma \right) |\sigma_f^T g_{f,i}| \\ &\quad - \gamma \|\sigma_f^T g_f\| \left(\|\widetilde{W}_\xi\| - \frac{1}{2} W_M \right)^2.\end{aligned}\quad (57)$$

If $k_{r,i} - \varepsilon_{M\xi,i} - W_M^2 \gamma / 4 > 0$, then $\dot{V}_f < 0$. Thus, σ_f and σ converge to the origin. According to (43), we obtain that the tracking error e_f converges to the origin. When σ converges to the origin, we have $\dot{\sigma} = 0$. Thus, we can know that the approximation error \widetilde{D}_f converges to the origin from (42). This includes the proof.

Remark 15. For the double power reaching law [32–34], the first term of (44) plays a key role when the system state is far away from the sliding mode ($|\sigma_{f,i}| > 1$), and the speed of the

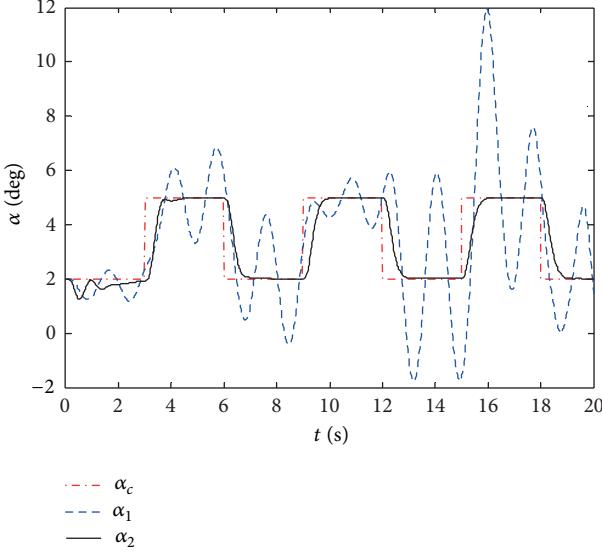


FIGURE 2: Angle of attack.

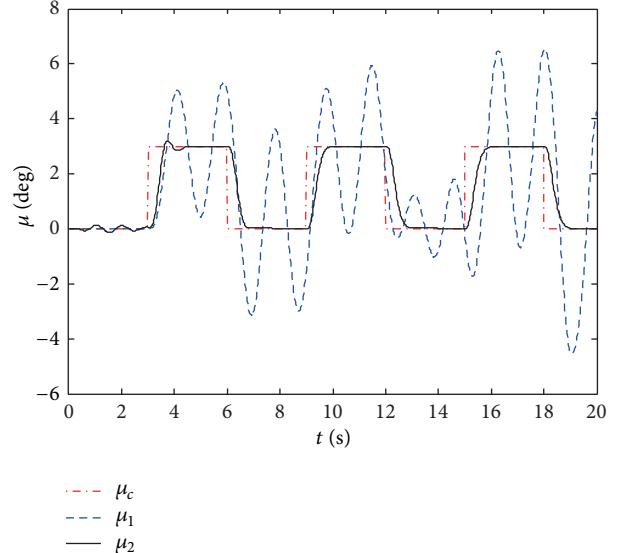


FIGURE 4: Roll angle.

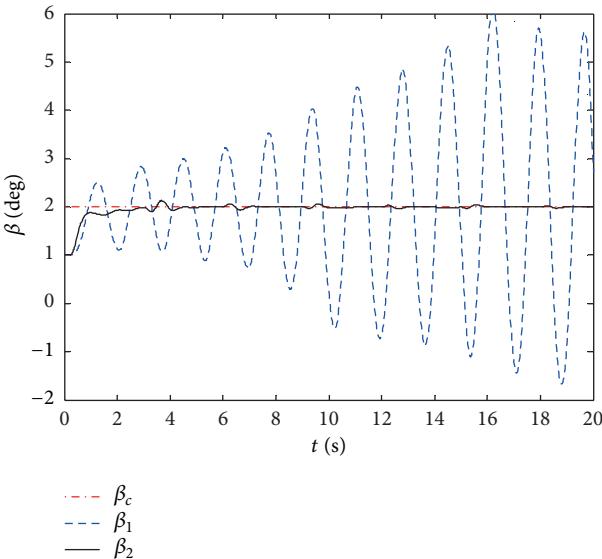


FIGURE 3: Sideslip angle.

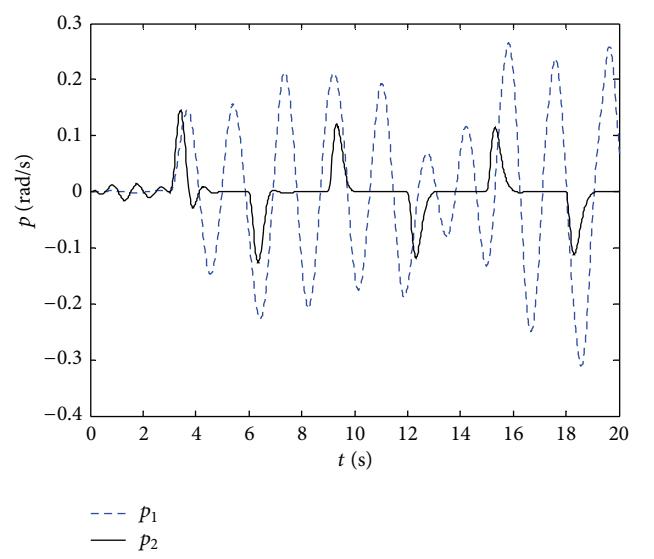


FIGURE 5: Roll angular rate.

double power reaching law is faster than that of the standard reaching law. On the other hand, when the system state is close to the sliding mode ($|\sigma_{f,i}| < 1$), the second term of (44) plays a key role, and the speed of the double power reaching law is lower than that of the standard reaching law. Therefore, the chattering phenomenon can be attenuated through the double power reaching law.

3.4. Simulation Study for the NSV. In this section, the simulation results are given to illustrate the effectiveness of the proposed attitude control schemes for the NSV. The initial values are chosen as $\alpha_0 = 2^\circ$, $\beta_0 = 1^\circ$, $\mu_0 = 0^\circ$, $p_0 = q_0 = r_0 = 0$ rad/s, $H_0 = 22000$ m, $V_0 = 600$ m/s, $\delta_{aM} = \delta_{eM} = \delta_{rM} = 30^\circ$, and $\delta_{yM} = \delta_{zM} = 15^\circ$. The desired flight attitudes are

chosen as

$$\begin{aligned}\alpha_c &= \begin{cases} 2^\circ, & 3k \leq t < 3(k+1), \\ 5^\circ, & 3(k+1) \leq t < 3(k+2), \end{cases} \\ \beta_c &= 2^\circ, \\ \mu_c &= \begin{cases} 0^\circ, & 3k \leq t < 3(k+1), \\ 3^\circ, & 3(k+1) \leq t < 3(k+2), \end{cases} \\ k &= 0, 1, 2, \dots\end{aligned}\tag{58}$$

Suppose that there are +20% and -20% uncertainties on aerodynamic coefficients and aerodynamic moment coefficients, respectively. On the other hand, the unknown external

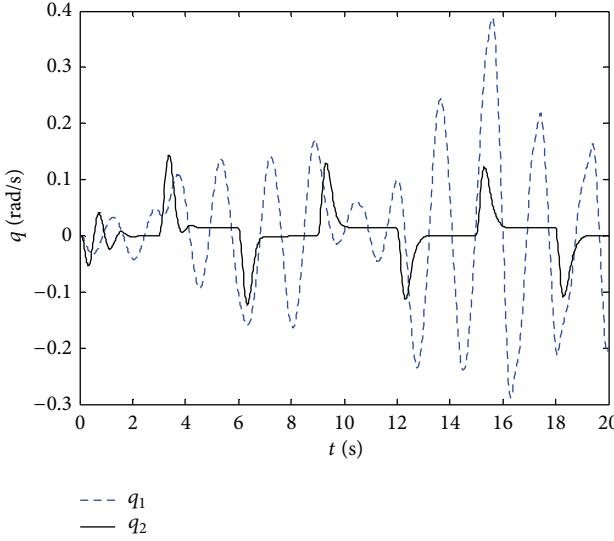


FIGURE 6: Pitch angular rate.

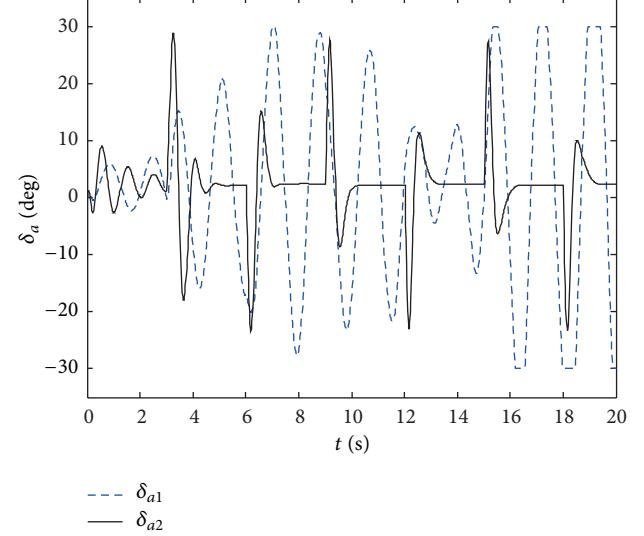


FIGURE 8: Ailerons.

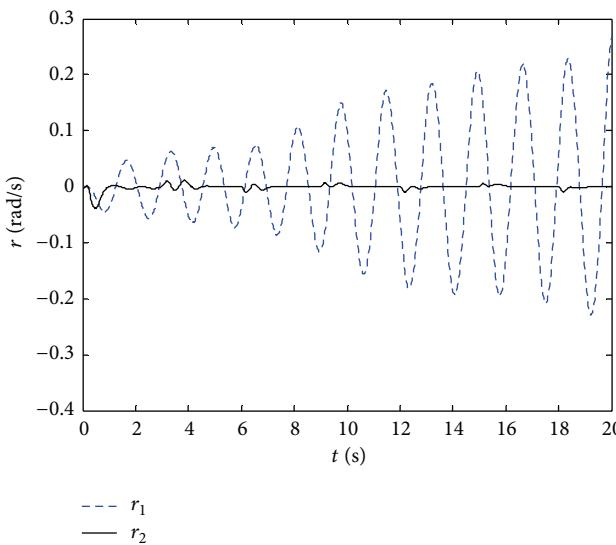


FIGURE 7: Yaw angular rate.

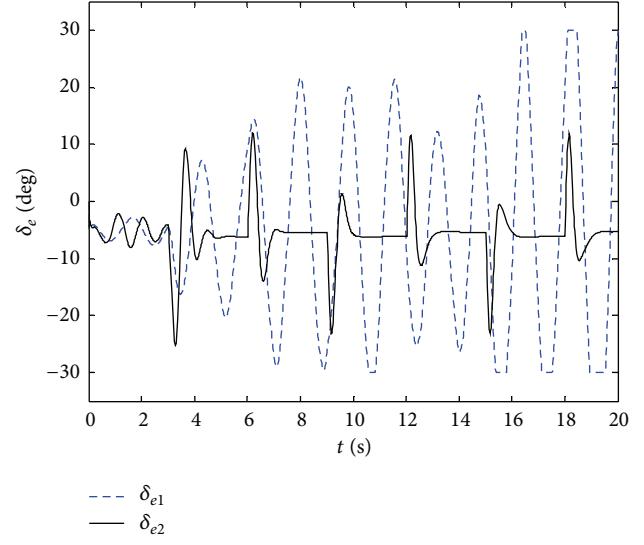


FIGURE 9: Elevator.

disturbance moments in fast-loop system of the NSV are given by

$$\begin{aligned} d_{Mf,1}(t) &= 4 \times 10^5 (\sin(5t) + 0.2) \text{ Nm}, \\ d_{Mf,2}(t) &= 4 \times 10^6 (\cos(5t) + 0.1) \text{ Nm}, \\ d_{Mf,3}(t) &= 4 \times 10^6 \sin(5t) \text{ Nm}. \end{aligned} \quad (59)$$

In order to obtain continuous and bounded derivatives of the desired flight attitudes, two-order reference model $G(s) = \omega_s^2/(s^2 + 2\xi\omega_s s + \omega_s^2)$ is employed, where ω_s and ξ are designed parameters. All parameters are given as $A_{s1} = \text{diag}\{1.5, 1, 1\}$, $A_{s2} = \text{diag}\{1, 1, 1\}$, $B_{s1} = \text{diag}\{0.2, 0.2, 0.2\}$,

$$\begin{aligned} B_{s2} &= \text{diag}\{0.2, 0.2, 0.2\}, \quad B_{s3} = \text{diag}\{0.1, 0.1, 0.1\}, \quad A_f = \text{diag}\{0.5, 0.5, 0.5\}, \quad B_{f1} = \text{diag}\{0.18, 0.18, 0.18\}, \quad B_{f2} = \text{diag}\{0.18, 0.18, 0.18\}, \quad K = \text{diag}\{6, 6, 6\}, \quad \Gamma_\beta = \text{diag}\{4, 4, 4\}, \\ \Gamma_\xi &= \text{diag}\{50\}_{15 \times 15}, \quad K_r = \text{diag}\{20, 20, 20, 20, 20\}, \quad \gamma = 0.001, \quad c_1 = 2.5, \quad c_2 = 0.5, \quad \omega_s = 4, \quad \text{and } \xi = 0.8. \end{aligned}$$

The simulation results are shown in Figures 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and 12, where dash dot lines (with subscript “c”) represent the desired attitudes and dot lines (with subscript “1”) stand for the responses without NDO and saturation compensation, while solid lines (with subscript “2”) represent the responses with NDO and saturation compensation.

From Figures 2 to 4, we can observe that the small tracking errors of attitude angles are obtained and the attitude angular rates tend to be stable from Figures 5 to 7 based on the proposed sliding mode control scheme while the oscillation phenomena emerge without the nonlinear

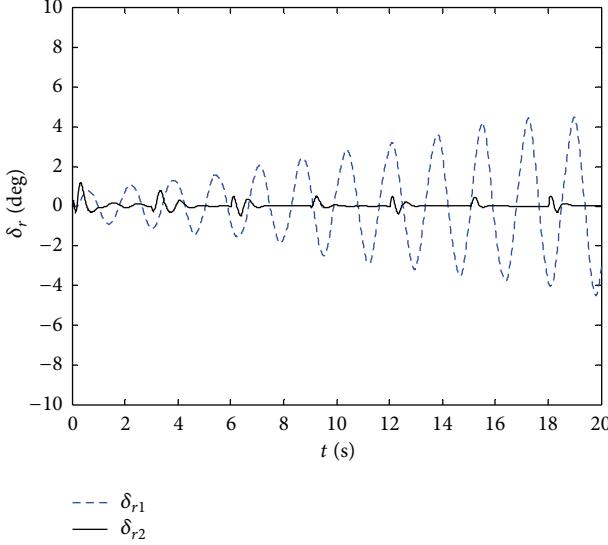


FIGURE 10: Rudder.

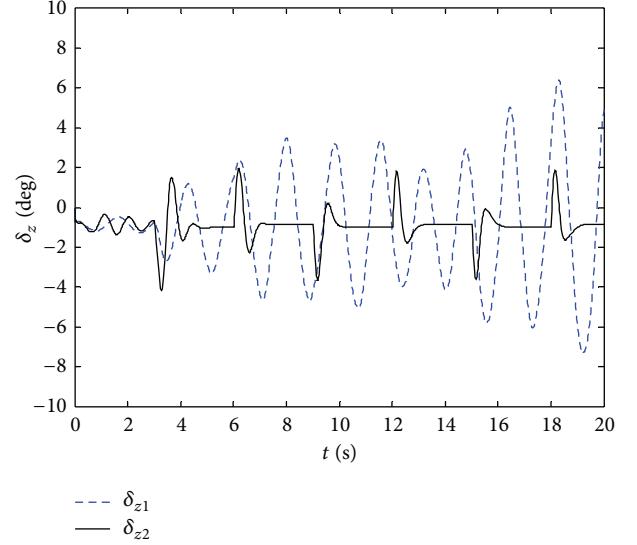


FIGURE 12: Longitudinal deflection of the thrust vectoring control surface.

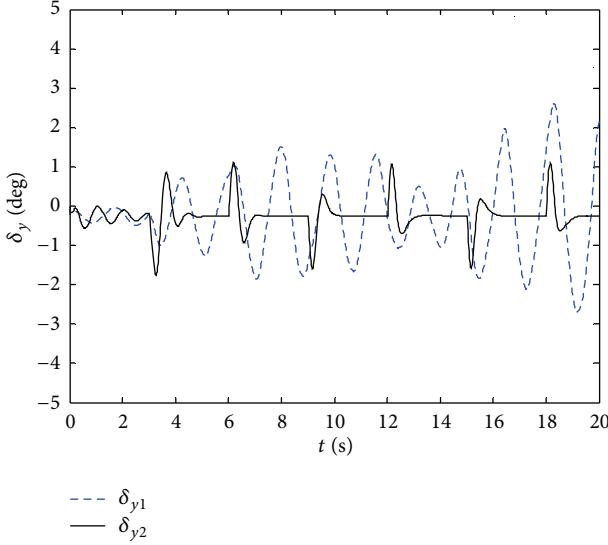


FIGURE 11: Lateral deflection of the thrust vectoring control surface.

disturbance observer and RBFNNs saturation compensator. On the other hand, the control inputs are presented from Figures 8 to 12. Under the proposed robust control scheme, it is clear that all deflections of rudders are in the saturation range. The deflections of ailerons and elevator reach the saturation limits without the nonlinear disturbance observer and RBFNNs saturation compensator. That is, the developed robust control scheme for the NSV can prevent the control signals from reaching saturation limits. From the previously stated analysis, the satisfactory attitude tracking control performance is obtained under the proposed robust control scheme for the NSV in the presence of the unknown external disturbance and the input saturation. Thus, the proposed robust control scheme is valid for the NSV.

4. Conclusion

An effective control scheme is proposed for the NSV with strong nonlinearity, high coupling, parameter uncertainty, and unknown time-varying disturbance based on SMC and NDO. Firstly, the nonlinear disturbance observer is designed to handle the compounded disturbance. Secondly, considering saturation characteristic of rudders, RBFNNs are constructed as a compensator to overcome the saturation nonlinearity. The stability of the closed-loop system is proved, and the tracking error as well as the disturbance observer error can converge to the origin through the Lyapunov analysis. Finally, Simulation results demonstrate the effectiveness of the proposed flight control scheme.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (Grant no. 61174102), Jiangsu Natural Science Foundation of China (Grant no. SBK2011069), Program for New Century Excellent Talents in University of China (Granted no. NCET-11-0830), NUAA Research Funding (Grant no. NS2013028), and “Liu Da Ren Cai Gao Feng” scheme of Jiangsu Province in China (Granted no. 2012-XXRJ-010).

References

- [1] Y. G. Wang, J. Q. Li, Y. Li, and W. Yao, “Characters and application prospects of near space flying vehicles,” *Spacecraft Engineering*, vol. 16, no. 1, pp. 50–57, 2007.
- [2] E. J. Cui, “Research statutes, development trends and key technical problems of near space flying vehicles,” *Advances Mechanics*, vol. 39, no. 6, pp. 658–673, 2009.
- [3] W. S. Nie, S. B. Luo, S. J. Feng, and F. C. Zhuang, “Analysis of key technologies and development trend of near space vehicle,”

- Journal of National University of Defense Technology*, vol. 34, no. 2, pp. 107–113, 2012.
- [4] M. Chen and B. Jiang, “Robust attitude control of near space vehicles with time-varying disturbances,” *International Journal of Control, Automation, and Systems*, vol. 11, no. 1, pp. 182–187, 2013.
- [5] M. Chen and W.-H. Chen, “Sliding mode control for a class of uncertain nonlinear system based on disturbance observer,” *International Journal of Adaptive Control and Signal Processing*, vol. 24, no. 1, pp. 51–64, 2010.
- [6] J. H. Wang, J. B. Hu, and B. F. Zhang, “Backstepping terminal sliding mode for flight control based on nonlinear disturbance observer,” *Journal of Applied Sciences*, vol. 30, no. 4, pp. 408–414, 2012.
- [7] M. Chen, R. Mei, and B. Jiang, “Sliding mode control for a class of uncertain MIMO nonlinear systems with application to near-space vehicles,” *Mathematical Problems in Engineering*, vol. 2013, Article ID 180589, 9 pages, 2013.
- [8] Y. H. Luo, H.-G. Zhang, and D.-S. Yang, “Neural network controller design and simulation based on saturation compensator,” *Journal of System Simulation*, vol. 21, no. 1, pp. 184–188, 2009.
- [9] W. Z. Gao, *Intelligent Control of Nonlinear Systems with Actuator Saturation Using Neural Networks*, Louisiana Tech University, 2005.
- [10] B. Xu, *Design of Adaptive Controller for a Class of Uncertain Nonlinear Systems with Input Saturation*, China University of Petroleum, 2010.
- [11] C. Wen, J. Zhou, Z. Liu, and H. Su, “Robust adaptive control of uncertain nonlinear systems in the presence of input saturation and external disturbance,” *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1672–1678, 2011.
- [12] Y. M. Li, S. C. Tong, and T. S. Li, “Direct adaptive fuzzy backstepping control of uncertain nonlinear systems in the presence of input saturation,” *Neural Computing and Applications*, 2012.
- [13] Z. Li and Y. Zhang, “Robust adaptive motion/force control for wheeled inverted pendulums,” *Automatica*, vol. 46, no. 8, pp. 1346–1353, 2010.
- [14] D. Qian, J. Yi, and D. Zhao, “Robust control using sliding mode for a class of under-actuated systems with mismatched uncertainties,” in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA ’07)*, pp. 1449–1454, Roma, Italy, April 2007.
- [15] Z. Gao, B. Jiang, H. Gong, and Y. Xu, “Fault-tolerant sliding mode control design for near space vehicle based on T-S fuzzy model,” in *Proceedings of the 4th International Conference on Innovative Computing, Information and Control (ICICIC ’09)*, pp. 211–214, December 2009.
- [16] Z. Li, J. Li, and Y. Kang, “Adaptive robust coordinated control of multiple mobile manipulators interacting with rigid environments,” *Automatica*, vol. 46, no. 12, pp. 2028–2034, 2010.
- [17] C. J. Schumacher, G. C. Cottrill, and H. H. Yeh, “Optimal sliding mode flight control,” in *AIAA Guidance, Navigation and Control Conference and Exhibit*, pp. 1–9, AIAA, Portlad, Or, USA, 1999.
- [18] Z. Li and Y. Kang, “Dynamic coupling switching control incorporating Support Vector Machines for wheeled mobile manipulators with hybrid joints,” *Automatica*, vol. 46, no. 5, pp. 832–842, 2010.
- [19] J. K. Liu, *MATLAB Simulation for Sliding Mode Control*, University Press, Beijing, China, 2005.
- [20] J.-Y. Kuan and H.-P. Huang, “Independent joint dynamic sliding mode control of a humanoid robot arm,” in *Proceedings of International Conference on Robotics and Biomimetics (ROBIO ’08)*, pp. 602–607, Bangkok, Thailand, February 2009.
- [21] A. J. Koshkouei, K. J. Burnham, and A. S. I. Zinober, “Dynamic sliding mode control design,” *IEE Proceedings: Control Theory and Applications*, vol. 152, no. 4, pp. 392–396, 2005.
- [22] M. Pu, Q.-X. Wu, C.-S. Jiang, and J. Zhang, “Near space vehicle control based on second-order dynamic Terminal sliding mode,” *Journal of Astronautics*, vol. 31, no. 4, pp. 1056–1062, 2010.
- [23] M. Pu, Q. Wu, C. Jiang, and L. Cheng, “Application of second-order dynamic Terminal sliding mode control based on nonlinear disturbance observer to near space vehicle,” *Journal of Southeast University*, vol. 39, no. 1, pp. 68–75, 2009.
- [24] A. Isidori, *Nonlinear Control Systems. II*, Communications and Control Engineering Series, Springer, London, UK, 1999.
- [25] J. Pang, R. Mei, and M. Chen, “Modeling and control for near space vehicles with oblique wing,” in *Proceedings of the 10th World Congress on Intelligent Control and Automation*, pp. 1773–1778, Beijing, China, 2012.
- [26] C. Yang, Y. Li, S. S. Ge, and T. H. Lee, “Adaptive control of a class of discrete-time MIMO nonlinear systems with uncertain couplings,” *International Journal of Control*, vol. 83, no. 10, pp. 2120–2133, 2010.
- [27] A. Levant, “Higher-order sliding modes, differentiation and output-feedback control,” *International Journal of Control*, vol. 76, no. 9–10, pp. 924–941, 2003.
- [28] J.-S. Fan, H.-X. Zhang, G.-M. Wang, and X.-G. Wang, “Improvement of higher order sliding mode control and its application,” *Control and Decision*, vol. 26, no. 9, pp. 1436–1440, 2011.
- [29] B. L. Tian, W. R. Fan, Q. Zong, J. Wang, and F. Wang, “Adaptive high order sliding mode controller design for hypersonic vehicle with flexible body dynamics,” *Mathematical Problems in Engineering*, vol. 2013, Article ID 357685, 11 pages, 2013.
- [30] C. Yang, S. S. Ge, C. Xiang, T. Chai, and T. H. Lee, “Output feedback NN control for two classes of discrete-time systems with unknown control directions in a unified approach,” *IEEE Transactions on Neural Networks*, vol. 19, no. 11, pp. 1873–1886, 2008.
- [31] Z. Li and C. Yang, “Neural-adaptive output feedback control of a class of transportation vehicles based on wheeled inverted pendulum models,” *IEEE Transactions on Control Systems Technology*, vol. 20, no. 6, pp. 1583–1591, 2012.
- [32] L. P. Xi, Z. L. Chen, and S. H. Zhang, “Design of sliding mode control scheme based on improved idempotent trending law for robotic manipulators,” *Computer Measurement & Control*, vol. 20, no. 2, pp. 380–382, 2012.
- [33] H. Mei and Y. Wang, “Fast convergent sliding mode variable structure control of robot,” *Information and Control*, vol. 38, no. 5, pp. 552–557, 2009.
- [34] H. X. Zhang, J. S. Fan, F. Meng, and J. F. Huang, “A new double power reaching law for sliding mode control,” *Control and Decision*, vol. 28, no. 2, pp. 289–293, 2013.

Research Article

An Abstract Description Method of Map-Reduce-Merge Using Haskell

Lei Liu,¹ Dongqing Liu,¹ Shuai Lü,^{1,2} and Peng Zhang¹

¹College of Computer Science and Technology, Jilin University, Changchun 130012, China

²College of Mathematics, Jilin University, Changchun 130012, China

Correspondence should be addressed to Shuai Lü; lus@jlu.edu.cn

Received 17 May 2013; Accepted 17 July 2013

Academic Editor: William Guo

Copyright © 2013 Lei Liu 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.

Map-Reduce-Merge is an improved parallel programming model based on Map-Reduce in cloud computing environment. Through the new Merge module, Map-Reduce-Merge can support processing multiple related heterogeneous datasets more efficiently. In order to demonstrate the validity and effectiveness of this new model, we present a rigorous description for Map-Reduce-Merge model using Haskell. Firstly, we describe the basic program skeleton of Map-Reduce-Merge programming model. Secondly, an abstract description for the Merge module is presented by analyzing the structure and function of the Merge module with Haskell as the description tool. Thirdly, we evaluate the Map-Reduce-Merge model on the basis of our description. We capture the functional characteristics of the Map-Reduce-Merge model by our abstract description, which can provide theoretical basis for designing more efficient parallel programming model to process join operation.

1. Introduction

Recently, lots of research works on improving Google's Map-Reduce [1] model have been proposed to analyze large volumes of data [2–5]. One important type of data analysis is joining multiple datasets. There are increasing efforts for implementing join algorithms using Map-Reduce or the improved Map-Reduce programming model [6–11]. Map-Reduce-Merge is such an effort that can directly express join operation and implement several join algorithms by the new Merge module. In this new model, Map and Reduce modules are inherited from Map-Reduce model, so that existing Map-Reduce programs can run directly on this new framework without modifications. Not only join operator but also all the other relational operators can be modeled using various combinations of the three primitives: Map, Reduce, and Merge. Map-Reduce-Merge removes the burden of implementing join algorithms. The emergency of Map-Reduce-Merge shows a trend that parallel databases and Map-Reduce learn with each other and new data analysis ecosystems are developed [12, 13].

Many formal methods can be used to describe programming model [14–17]. Lämmel [18] first delivers a rigorous description of Map-Reduce programming model as

well as its advancement called Sawzall [19]. He uses typed functional programming Haskell as a tool to describe the fundamental characteristics underlying the Map-Reduce and Sawzall model. The description is made up of several Haskell functions. Our paper is based on his work. We will present an abstract description for Map-Reduce-Merge model, especially for the new added Merge module. This paper makes the following contributions. Firstly, we define the basic program skeleton of Map-Reduce-Merge to capture the abstraction for Map-Reduce-Merge computations. Secondly, we decompose the Merge module according to its structure and present the rigorous description called *moduleMerge*. Thirdly, we analyze the Map-Reduce-Merge programming model based on our abstract description with an example. Some implementation details (such as fault tolerance and task scheduling [20–25]) will be considered in the future.

Haskell is characterized by strong type inference and type checking [26]. The recent paper [18] suggests that Haskell can be used as a tool to support executable specification. Using Haskell as a description tool can be beneficial for both programming model designers and users. For designers, they can know explicitly what will happen during the execution of a Map-Reduce-Merge job, which is good for them to analyze

$\begin{aligned} \text{map} &:: (a \rightarrow b) \rightarrow [a] \rightarrow [b] \\ \text{map } f [] &= [] \\ \text{map } f (x: xs) &= fx : \text{map } f xs \end{aligned}$	--type of map --the empty list case --the non-empty list case
--	---

ALGORITHM 1

$\begin{aligned} \text{map (+1)} [1,2,3] &= 2: \text{map (+1)} [2,3] \\ &= 2: 3: \text{map (+1)} [3] \\ &= 2: 3: 4: [] \\ &= [2, 3, 4] \end{aligned}$

ALGORITHM 2

and evaluate this new model. For users, they can use the abstract description as an executable specification in software development to ensure the correctness and robustness of software.

This paper is organized as follows. Section 2 gives a brief introduction of Map-Reduce-Merge programming model and Haskell programming language. Section 3 defines the basic programming skeleton of Map-Reduce-Merge programming model. Section 4 designs the helper functions composing the Merge module. Section 5 defines the helper functions designed in Section 4. Section 6 shows an example and gives a brief comment. Section 7 concludes this paper.

2. Background

In this section, we will briefly recall Map-Reduce-Merge programming model and some concepts in Haskell.

2.1. Map-Reduce-Merge Programming Model. The most important feature of Map-Reduce-Merge is that it adds to Map-Reduce a Merge phase, so that it can directly support join algorithms of different datasets. Figure 1 illustrates the data flow of the Map-Reduce-Merge framework. It consists of three phases, two independent Map-Reduce phases, and a Merge phase. At first, two datasets are processed by corresponding Map and Reduce phases. Then in the Merge phase, a merger can select data to be merged from those two reducer outputs according to different join algorithm. Notice that the reducer outputs are stored in local disks instead of distributed file system because the Reduce phase is not the last phase any more.

The main purpose of the Merge module is to implement join algorithms. The Merge module includes four components: *Partition Selector*, *Processors*, *Configurable Iterators*, and *Merger* as shown in Figure 2. *Partition Selector* is a selector that determines which data partitions produced by up-stream reducers should be retrieved and then merged. *Configurable Iterators* include two logical iterators which can implement different join algorithms, including sort-merge join, nested-loop join, and hash join. *Processors* include two processor functions which define the logic to process data

from different datasets. *Merger* includes a merger function where users can implement data processing logic on data from two sources.

2.2. Haskell Programming Language. To avoid confusion, we will introduce the *map* function and the *Map* type in Haskell, as well as the *map* primitive in Map-Reduce programming model.

In Haskell, the *map* function is a higher-order function. It takes two parameters, the first is a function whose type is $a \rightarrow b$, and the second is a list whose type is $[a]$. It returns a list whose type is $[b]$. The formal definition of *map* is as in Algorithm 1.

We illustrate how to use *map* to add one to every element in a list. Here, the expression $(+1)$ represents a function that adds one to a variable, which equals to the function $f(x) = x + 1$. The example is shown as in Algorithm 2.

The type *Map* is a build-in type in Haskell. It is an efficient implementation of maps from keys to values. We can use the function *toList* to convert a *Map* to an association list and the function *fromList* to build a *Map* from an association list.

The signature of *map* primitive is $(K1, V1) \rightarrow [(K2, V2)]$. The *map* primitive in essence corresponds to the first parameter of the function *map* in Haskell [18]. We define the signatures of *map*, *reduce*, and *merge* primitives in Table 1. The first digit after letter *K/V* is used to distinguish different keys/values, while the second digit is used to distinguish different datasets.

3. Definition of *mapReduceMerge*

In this section, we define the *mapReduceMerge* function to model the abstraction for Map-Reduce-Merge computations. A full Map-Reduce-Merge job includes two individual Map-Reduce phases and a Merge phase as shown in Figure 1. Hence, we take for granted that the *mapReduceMerge* function can be decomposed into three helper functions that represent these three phases, respectively. The type and definition for *mapReduceMerge* are shown as in Algorithm 3.

The *mapReduceMerge* function is defined in terms of function application in Haskell. The arguments *lTable* and *rTable* corresponding to the types $[Map K11 V11]$ and $[Map K12 V12]$ are the input data for a Map-Reduce-Merge job. They are first processed by the functions *mapReduce1* and *mapReduce2*, respectively, and then are merged by the function *moduleMerge*. These helper functions correspond to three phases in a Map-Reduce-Merge computation.

By taking advantage of the works done in [18], a Map-Reduce job is divided into three phases: Map, Shuffle, and Reduce. We can define *mapReduce* as in Algorithm 4.

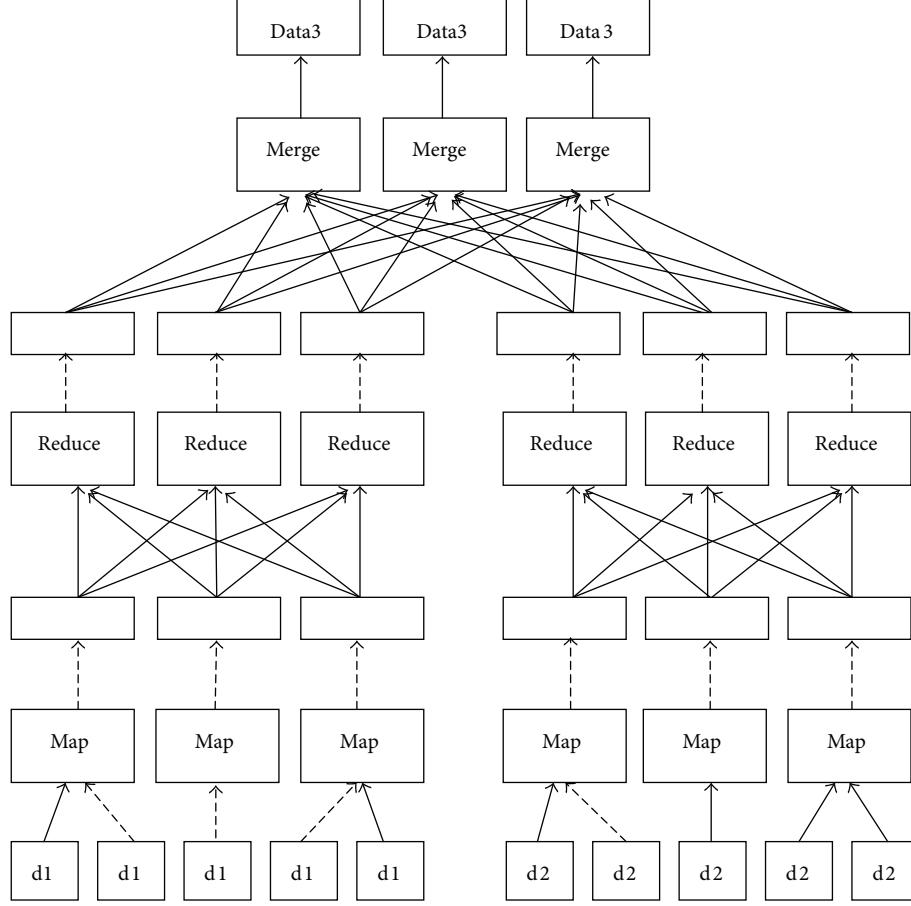


FIGURE 1: Data flow for the Map-Reduce-Merge framework.

Here, function composition, which is denoted by Haskell infix operator `(.)`, is used to compose three helper functions. A more detailed definition for `mapReduce` can be found in [18]. In this paper, we mainly focus on describing the Merge module with the `moduleMerge` function. By analyzing `mapReduceMerge` and `mapReduce`, we can discover the type for `moduleMerge` with its definition to be defined later (Algorithm 5).

Now, we will explain the types used in the definition of `mapReduceMerge`. The type `Map K1 V1` represents the input split type for `moduleMap`. Typically, every split corresponds to a map task. Hence, the list type `[Map K1 V1]` represents all the input data of a Map-Reduce job. Similarly, the type `Map K2 V3` represents the output result of a reduce task, and the list type `[Map K2 V3]` represents all the output data of a Map-Reduce job. Our types are compatible with the types defined in [18].

In parallel databases, a table is divided into different splits in order to store in large clusters. The splits form the unit of distribution and load balancing. In this paper, we use `Table` to represent a table, `Tablet` to represent a split of a table, and `Record` to represent a row in a table. According to our discussion above, `Table` has a type of `[Map K V]`, `Tablet` has a type of `Map K V`, and `Record` has a type of `(K, V)`. It happens that a table in Google's Bigtable [27] is a sparse, distributed, persistent multidimensional sorted `Map`.

4. Discovery of the Types in Merge

In this section, we design some helper functions for `moduleMerge` and discover the types of those functions. The `moduleMerge` function is defined to model the abstraction for the Merge module. According to Figure 2, the components of the Merge module can be divided into two parts, data transferring and data processing. First part includes `Partition Selector`, which can select and transfer the output of up-stream reducers. Second part is made up of `Processors`, `Configurable Iterators`, and `Merger`, which can merge two different datasets. Hence, we design two helper functions for `moduleMerge`, which are `getPartitionPair` and `mergeTwoPartition`. The `getPartitionPair` function selects the output `Tablets` from up-stream reducers and then delivers those Tablets to mergers. The `mergeTwoPartition` function takes two Tablets as input and returns a new `Tablet` as the merge result. The types of these two functions are shown as in Algorithm 6.

Here, `[Map K21 V31]` denotes the input type of `moduleMerge`. It coincides with the result type of `mapReduce1`, which reflects the fact that the output of `mapReduce1` is one input for `moduleMerge`. The type `(Map K21 V31, Map K22 V32)` represents two `Tablets` to be merged. The type `[(Map K21 V31, Map K22 V32)]` represents all the `Tablet` pairs that is a merger, process and the type `[[Map K21 V31, Map K22 V32]]` represents all the `Tablet` pairs that is all the mergers process.

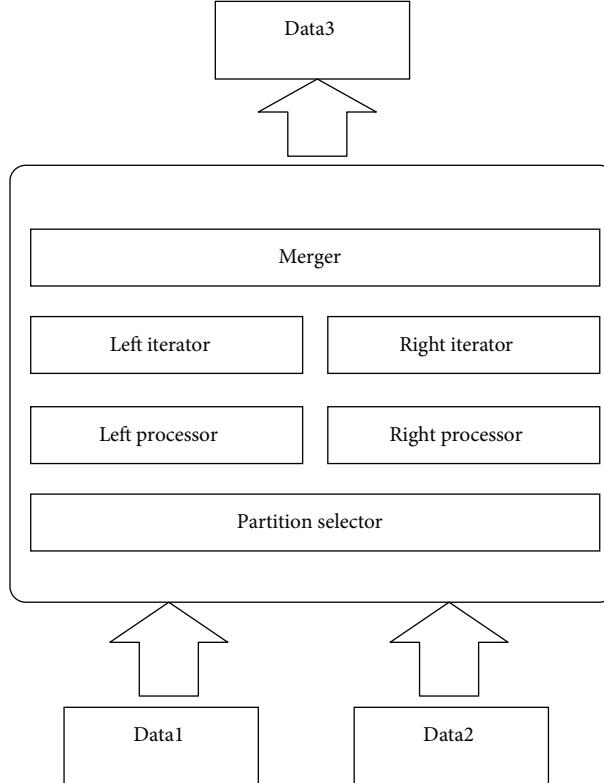


FIGURE 2: Data flow for the Merge module.

TABLE 1: The signatures of three Map-Reduce-Merge primitives.

	Source 1	Source 2
Map	$K11 \rightarrow V11 \rightarrow [(K21, V21)]$	$K12 \rightarrow V12 \rightarrow [(K22, V22)]$
Reduce	$K21 \rightarrow [V21] \rightarrow \text{Maybe } V31$	$K22 \rightarrow [V22] \rightarrow \text{Maybe } V32$
Merge	$(K21, V31) \rightarrow (K22, V32) \rightarrow (K23, V33)$	

```

mapReduceMerge :: [Map K11 V11] → [Map K12 V12] → [Map K23 V33]
mapReduceMerge lTable rTable = moduleMerge (mapReduce1 lTable)(mapReduce2 rTable)

```

ALGORITHM 3

```

mapReduce :: [Map K1 V1] → [Map K2 V3]
mapReduce = moduleReduce.moduleShuffle.moduleMap

```

ALGORITHM 4

```

moduleMerge :: [Map K21 V31] → [Map K22 V32] → [Map K23 V33]
moduleMerge = undefined

```

ALGORITHM 5

<i>getPartitionPair :: [Map K21 V31]</i>	--Table1 as distributed input data
→ [Map K22 V32]	--Table2 as distributed input data
→ [(Map K21 V31, Map K22 V32)]	--The tasks of all mergers
<i>mergeTwoPartition :: Map K21 V31</i>	--Tablet1 as input data
→ Map K22 V32	--Tablet2 as input data
→ Map K23 V33	--Tablet3 as output data

ALGORITHM 6

<i>getPair :: Map K21 V31</i>	--Tablet1 as input data
→ Map K22 V32	--Tablet2 as input data
→ [(KV1, KV2)]	--The task of one merger
<i>mergeTwoRecord :: KV1</i>	--Record1 as input data
→ KV2	--Record2 as input data
→ Maybe KV3	--Maybe Record3 or Nothing as output data

ALGORITHM 7

Every merger has a serial number in the Map-Reduce-Merge programming model. We implicitly express this characteristic by the merger position in the list.

We decompose the *mergeTwoPartition* function into two helper functions called *getPair* and *mergeTwoRecord*. The *getPair* function chooses and emits all the *Record* pairs that will be processed by the *mergeTwoRecord* function, where a *Record* pair is merged into a new *Record*. In fact, *mergeTwoPartition* is implemented by executing *mergeTwoRecord* many times. The types of these two functions are shown as in Algorithm 7.

Here, the type *KV1* is short for the type *(K21, V31)*, *KV2* is short for *(K22, V32)*, and *KV3* is short for *(K23, V33)*. The type *[(KV1, KV2)]* represents all the possible *Record* pairs that need to be merged. The type *Maybe* is a built-in type in Haskell, which is defined as “*data Maybe a = Just a | Nothing*.” Only those *Record* pairs that satisfy the merge condition can be merged into a new *Record* whose type is *Just KV3*. The type *(Map K21 V31, Int)* represents a *Tablet* and its number, which corresponds to the reducer number in Map-Reduce-Merge.

The *getPair* function has to decide whether two *Records* need to be merged or not, so the *iterationLogic* function is designed to implement this work. The same thing happened in the *getPartitionPair* function, where *partitionSelector* is designed to judge whether two *Tablets* need to be merged. The types for *iterationLogic* and *partitionSelector* are shown as in Algorithm 8.

We now summarize the components of the Merge module and the designed functions, as well as the relationship between them. In the Merge module, *Partition Selector* selects the input data for mergers from reducers. We design the *partitionSelector* function to model it. The *partitionSelector* function is invoked by *getPartitionPair* to transfer data from reducers to mergers. Hence, this phase is similar to the shuffle phase in Map-Reduce. *Configurable Iterator* includes two iterators corresponding to two datasets. All join algorithms, such as sort-merge join, nested-loop join, and hash join, have their own processing logic to control the relative

movements of two iterators. In our paper, we abstract it by the *iterationLogic* function according to the characteristics of Haskell. The *iterationLogic* function is called by *getPair* to generate all the wanted *Record* pairs. *Merger* that implements the user-defined logic is captured by the *mergeTwoRecord* function. *Processors* include two functions called *leftProcessor* and *rightProcessor*, which can implement hash join. For simplicity, we model it with the *iterationLogic* function. The relationship of our functions is summarized in Figure 3.

5. Definition of the Helper Functions in Merge

In this section, we describe how to implement the functions whose types have been determined in last section. After every function, some concepts of Haskell are explained when needed.

5.1. *mergeTwoRecord*. The *mergeTwoRecord* function is composed of two helper functions *match* and *mergeResult*. We use the *match* function to judge whether the join keys of two *Records* satisfy the given condition. If *K21* and *K22* satisfy a merge condition defined by the *match* function, these two *Records* will be processed by *mergeResult* where users can implement data processing logic. The definitions of *match* and *mergeResult* are related to concrete applications. In Section 6, we will illustrate how to use these two functions. The definition of *mergeTwoRecord* is as in Algorithm 9.

In the definition of *mergeTwoRecord*, the symbol “@” which is called as-pattern is a kind of pattern matching forms in Haskell. It denotes that *KV1* can be replaced by *(K21, V31)* and *KV2* can be replaced by *(K22, V32)*. The symbol “\$” denotes function application which can be substituted by parentheses. In this definition, “*Just \$ mergeResult KV1 KV2*” is equal to “*Just (mergeResult KV1 KV2)*.” This representation is much simpler than using parentheses, especially when nested parentheses are needed. The vertical pipe “|” represents guards in Haskell. It is used to judge whether the parameters satisfy some conditions. In this definition, if *K21* and *K22*

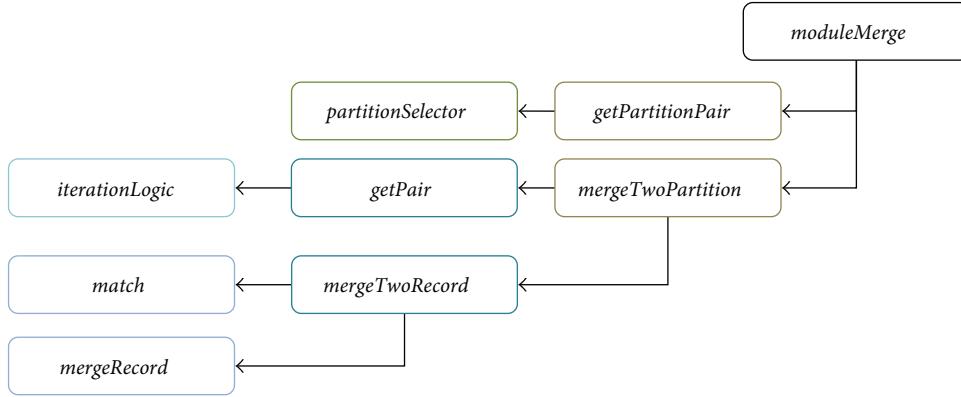


FIGURE 3: The relationship of the functions in Merge.

<i>iterationLogic :: KV1</i>	--Record1 as input data
→ KV2	--Record2 as input data
→ Bool	--Need to be merged or not
<i>partitionSelector :: (Map K21 V31, Int)</i>	--Tuple (Tablet1, Id1) as input data
→ (Map K22 V32, Int)	--Tuple (Tablet2, Id2) as input data
→ Bool	--Choose Tablet pair to be merged

ALGORITHM 8

```

mergeTwoRecord KV1@(K21, V31) KV2@(K22, V32)
| match K21 K22 = Just $ mergeResult KV1 KV2
| otherwise = Nothing
  
```

ALGORITHM 9

```

getPair lTablet rTablet =
  filter (uncurry iterationLogic)
  [(pairLeft, pairRight) | pairLeft ← toList lTablet, pairRight ← toList rTablet]   -- Step 2
                                         -- Step 1
  
```

ALGORITHM 10

satisfy the condition defined in the *match* function, *KV1* and *KV2* will be processed by the *mergeResult* function and then wrapped by the type *Maybe*. Otherwise, *mergeTwoRecord* will return *Nothing* as the result.

5.2. *getPair*. We implement *getPair* in two steps as follows.

Step 1. Get all possible combinations of *Records* from *lTablet* and *rTablet*; the result is the Cartesian product of these two *Tablets*.

Step 2. Filter the undesired *Record* pairs with *iterationLogic*.

In Haskell, list comprehensions are a handy way to produce lists. Their concepts are similar to set comprehensions in mathematics. Here in Step 1, the list comprehension is used for building a list out of two lists.

We could also do the same thing using set comprehensions, like this: $\{(pairLeft, pairRight) \mid pairLeft \in toList lTablet, pairRight \in toList rTablet\}$. The *filter* function takes a predicate and a list and returns the list whose elements satisfy that predicate. In Step 2, those elements which satisfy the conditions defined in *iterationLogic* will remain in the list, while others will be removed from the list. The definition of *getPair* is as in Algorithm 10.

5.3. *mergeTwoPartition*. The *mergeTwoPartition* function is implemented in the following four steps.

Step 1. Get the list of pairs from *lTablet* and *rTablet* by invoking *getPair*.

Step 2. Process every pair from the list with *mergeTwoRecord*. As a result; the return type is [*Maybe KV3*].

```

MergeTwoPartition lTablet rTablet =
fromList
$ map (\(Just x) → x) $ filter (\x → x/=Nothing)      -- Step 4
$ map (uncurry mergeTwoRecord)                         -- Step 3
$ getPair lTablet rTablet                             -- Step 2
$ getPair lTablet rTablet                             -- Step 1

```

ALGORITHM 11

```

getPartitionPair lTable rTable =
map (map (\((lTable, lNum), (rTable, rNum)) → (lTable, rTable)))           -- Step 4
$ map (filter (uncurry partitionSelctor))                                     -- Step 3
$ [[(pairLeft, pairRight) | pairRight ← rTableNum] | pairLeft ← lTableNum]   -- Step 2
where
lTableNum = zip lTable [1,2..]
rTableNum = zip rTable [1,2..]

```

ALGORITHM 12

```

moduleMerge lTable rTable =
map (foldl union empty)                                         -- Step 3
$ map (map (uncurry mergeTwoPartition))                         -- Step 2
$ getPartitionPair lTable rTable                                -- Step 1

```

ALGORITHM 13

Step 3. Remove the element equal to *Nothing* from the list with the function *filter*. Then change the element type from *Maybe KV3* to *KV3* with the *map* function.

Step 4. Change the list type to the type *Map* with the *fromList* function.

In the definition of the *mergeTwoPartition* function, the symbol “\” denotes λ expression. For example, the λ expression “ $\lambda x \rightarrow x/=Nothing$ ” represents a function whose parameter is x and function body is $x/=Nothing$. Hence, this function decides whether x is equal to *Nothing*. The definition of *mergeTwoPartition* is as in Algorithm 11.

5.4. *getPartitionPair*. We implement the *getPartitionPair* function in the following four steps.

Step 1. Use the *zip* function to identify each *Tablet* with a number. It simulates the situation that we use the reducer number to select the output data.

Step 2. Get all possible combinations of the elements from *lTable* and *rTable*, just like the Cartesian product of two *Tables*. In fact, we need all possible combinations of *lTablets* and *rTablets*.

Step 3. Filter the useless combinations with the *partitionSelector* function.

Step 4. Remove the *Num* part from pair (*Tablet*, *Num*).

The definition of *getPartitionPair* is as in Algorithm 12.

5.5. *moduleMerge*. We implement the *moduleMerge* function in the following three steps.

Step 1. Get all the combinations for *Tablets* from *lTable* and *rTable* with the *getPartitionPair* function.

Step 2. Merge two *Tablets* from different *Tables* with the *mergeTwoPartition* function. The outermost *map* application corresponds to the parallel merge tasks. The innermost *map* application corresponds to a merge task where the *mergeTwoPartition* function is executed.

Step 3. Concatenate all the partitions that produced by a merger with the *foldl* function.

In the definition of the *moduleMerge* function, we use some concepts about type *Map* and its operations. The function *foldl* has three parameters: a binary function, an initial value, and a *Map* type value. It returns a result that is the same type as the initial value. The *union* function combines two dictionaries into one dictionary. The value *empty* is an empty dictionary. The definition of *moduleMerge* is as in Algorithm 13.

5.6. *iterationLogic* and *partitionSelector*. The *iterationLogic* and *partitionSelector* functions are implemented according to different join algorithms as in Algorithm 14.

```

partitionSelector :: (Map K21 V31, Int) → (Map K22 V32, Int) → Bool
partitionSelctor (lTablet, left) (rTablet, right) = True
iterationLogic :: KV1 → KV2 → Bool
iterationLogic (K21, V31)(K22, V32) = True

```

ALGORITHM 14

type K21 = (Int, String) type V31 = Float type K22 = String type V32 = Float type K23 = Int type V33 = Float	--Employee key is (emp-id,dept-id) --Employee value is bonus --Department key is dept-id --Department value is bonus-adjustment --Bonus key is emp-id --Bonus value is bonus-final
---	---

ALGORITHM 15

```

lTablet1 = fromList [((2, "A"), 0), ((3, "A"), 250), ((1, "B"), 150)]      --Tablet1 of Table1
lTablet2 = fromList [((4, "B"), 0), ((7, "C"), 200), ((6, "D"), 100)]      --Tablet2 of Table1
lTablet3 = fromList [((5, "A"), 300), ((8, "C"), 150), ((9, "D"), 0)]      --Tablet3 of Table1
lTable = [lTablet1, lTablet2, lTablet3]                                         --Table1
rTablet1 = fromList [(("A", 0.95), ("B", 1.15))]                            --Tablet1 of Table2
rTablet2 = fromList [(("C", 1.00), ("D", 1.25))]                            --Tablet2 of Table2
rTable = [rTablet1, rTablet2]                                                 --Table2

```

ALGORITHM 16

The *iterationLogic* function judges whether two *Records* satisfy the merge condition. As we can see in this definition, we assume that it is always true no matter what the input data is. This is correct when we want to implement nested-loop join. The definition will change to $K21=K22$ if sort-merge join is implemented.

The *partitionSelector* function selects those *Tablets* that need to be merged. Just like the *iterationLogic* function, the return value is true when we want to implement nested-loop join. We will change the definition to $left==right$ when sort-merge join is needed, on condition that two Map-Reduce phases use the same partitioners.

As we can see in this section and last section, our method to describe the Merge module is “firstly design the types and then define the functions.” The types are designed from top to down, while the definitions are defined from bottom to up. At last, we get the abstract description of the Merge module by using Haskell as shown in Figure 4.

6. Case Study

In this section, an example of how to use our description is designed to demonstrate the proposed method. Then a brief analysis on two join algorithms that have been implemented in Map-Reduce-Merge is given, including nested-loop join and sort-merge join.

There are two tables in this example: Employee and Department. We use *Table1* and *Table2* to represent them, respectively. The primary keys of the tables are shown in bold

in Figures 5 and 6. One possible query is to compute the employee final bonus that is the product of bonus in *Table1* and bonus adjustment in *Table2*. This query result is stored in *Table3*. To accomplish this query, we take steps as follows.

Firstly, we divide the record in a table into two parts, corresponding to *Key* and *Value* in Map-Reduce, respectively, and assign every attribute with a Haskell type. In *Employee*, we chose the composition of *emp-id* and *dept-id* as *Key* and the others as the *Value*. As to *Department*, we chose *dept-id* as *Key* and the others as the *Value*. All the types of *Key* and *Value* emerging in the tables are defined as in Algorithm 15.

Notice that the *Key* we use in *Map-Reduce* is not the same as the key we use in databases. In our example, we use the same *Key* as the primary key in *Table1*, while in *Table2* we chose not to. In the subsequent steps, we implement the nested-loop join with our description. The way to process sort-merge join is similar except that we use other definitions of *iterationLogic* and *partitionSelector*, which will be discussed later.

Secondly, we construct the input data by modifying the table to the form of list of Maps as follows. *Table1* consists of three *Tablets*, and *Table2* consists of two *Tablets* (see Algorithm 16).

Thirdly, we define the functions *match* and *mergerResult*. In the *match* function, we guarantee that the employee *dept-id* is equal to the department *dept-id*. Only those *Record* pairs that satisfy this condition can be merged. In the *mergeResult* function, we implement the product of bonus and bonus adjustment to get the final bonus. The types and definitions of these two functions are shown as in Algorithm 17.

```

import Data.Map (Map, empty, union, fromList, toList)
moduleMerge :: [Map K21 V31] → [Map K22 V32] → [Map K23 V33]
moduleMerge lTable rTable =
    map (foldl union empty)
    $ map (map (uncurry mergeTwoPartition))
    $ getPartitionPair lTable rTable
    where
        mergeTwoPartition :: Map K21 V31 → Map K22 V32 → Map K23 V33
        mergeTwoPartition lTablet rTablet =
            fromList
            $ map ((Just x)→x) $ filter ((x→x/=Nothing))
            $ map (uncurry mergeTwoRecord) $ getPair lTablet rTablet
            where
                mergeTwoRecord :: KV1 → KV2 → Maybe KV3
                mergeTwoRecord kv1@(k21, v31) kv2@(k22, v32)
                    | match k21 k22 = Just $ mergeResult kv1 kv2
                    | otherwise = Nothing
                getPair :: Map K21 V31 → Map K22 V32 → [(KV1, KV2)]
                getPair lTablet rTablet =
                    filter (uncurry iterationLogic)
                    [(pairLeft, pairRight) | pairLeft ← toList lTablet,
                    pairRight ← toList rTablet]

        getPartitionPair :: [Map K21 V31] → [Map K22 V32]
                        → [[[Map K21 V31, Map K22 V32]]]
        getPartitionPair lTable rTable =
            map (map (\((lTablet, lNum), (rTablet, rNum))→(lTablet,
            rTablet)))
            $ map (filter (uncurry partitionSelcetor))
            $ [[(pairLeft, pairRight) | pairRight ← rTableNum] |
            pairLeft ← lTableNum]
            where
                lTableNum = zip lTable [1, 2..]
                rTableNum = zip rTable [1, 2..]
                partitionSelcetor :: (Map K21 V31, Int)
                                → (Map K22 V32, Int)
                                → Bool
                partitionSelcetor (lTablet, left) (rTablet, right) = True

```

FIGURE 4: The abstract description for *moduleMerge*.

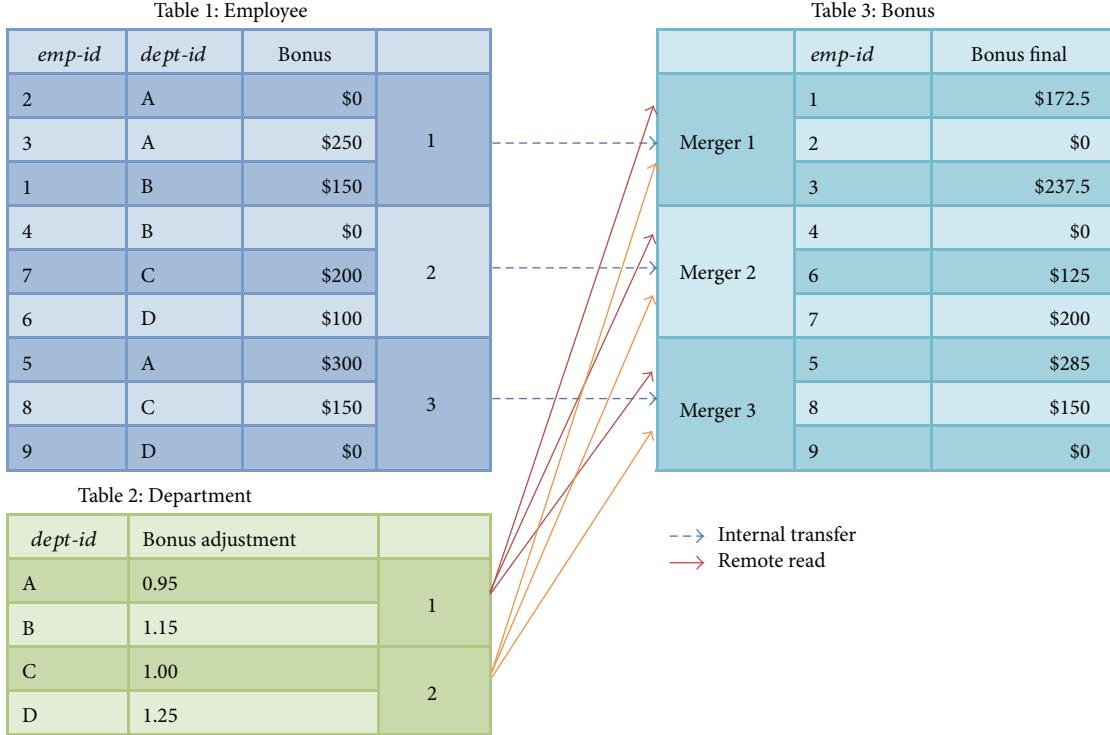


FIGURE 5: Data flow of nested-loop join.

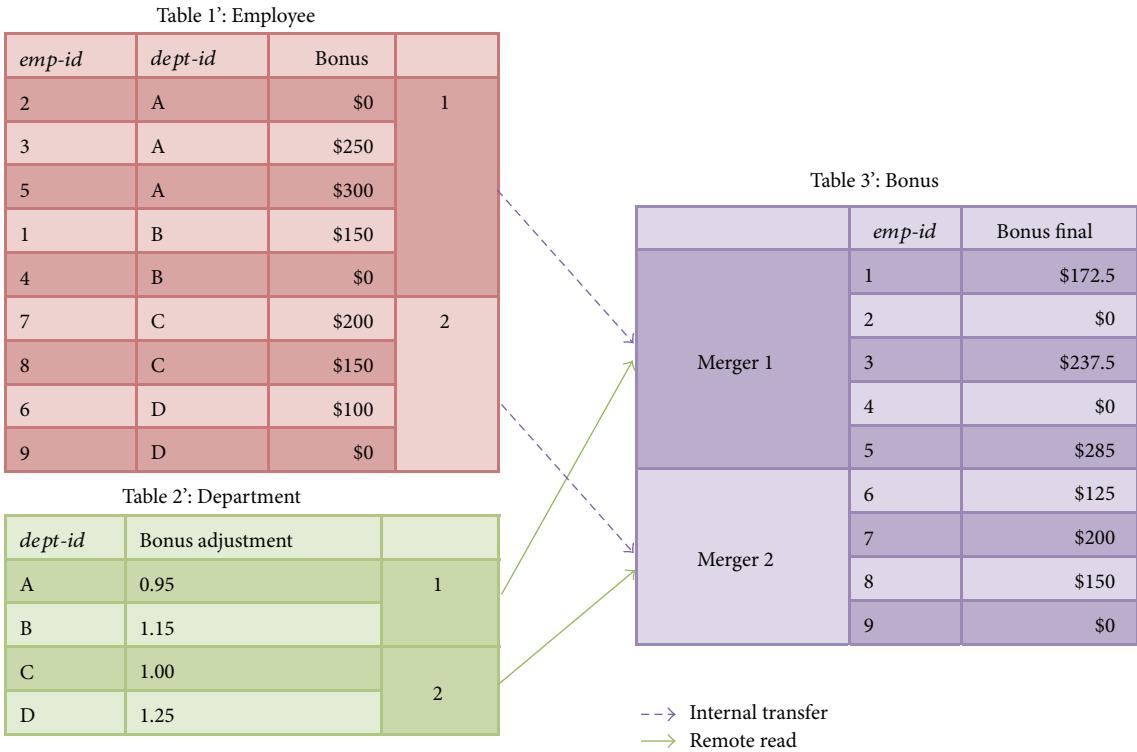


FIGURE 6: Data flow of sort-merge join.

```

match :: K21      --Key21 as input data
      → K22      --Key22 as input data
      → Bool     --Satisfy the merge condition or not
match K21 K22 = let (int, str) = K21 in if str == K22 then True else False
mergeResult :: KV1      --Record1 as input data
      → KV2      --Record2 as input data
      → KV3      --Record3 as output data
mergeResult (K21, V31) (K22, V32)= let (int, str)= K21 in (int, V31*V32)

```

ALGORITHM 17

Finally, we run the following command in Haskell compiler WinGHCi:

Prelude > moduleMerge lTable rTable.

The result is as follows. It corresponds to Table 3 in Figure 5:

[*fromList*[(1,172.5),(2,0.0),(3,237.5)],*fromList*[(4,0.0),
(6,125.0),(7,200.0)],*fromList*[(5,285.0),(8,150.0),(9,0.0)]].

It shows that our description can implement these two join algorithms. The corresponding dataflow graphs are Figures 5 and 6. We set the result of the *partitionSelector* function to be true in order to ensure that every *Tablet* from *Table1* can be merged with all tables in *Table2*. On the other hand, we set the function body of *partitionSelector* to be *left==right* so as to map reducers and mergers in a one-to-one relationship. We use the same strategy to define the *iterationLogic* function. The main difference between sort-merge join and nested-loop join is that the input of sort-merge join has been sorted, while the input of nested-loop join has not been. In this example, we take advantage of the combining phase strategy in Map-Reduce-Merge framework to reduce the remote read between reducers and mergers. Since the size of *Table1* is much bigger than the size of *Table2*, we combine the mergers with the reducers into same workers.

As we can see there are two processors and two iterators in the Merge module. Hence, Map-Reduce-Merge can implement two-way join algorithms. If we want to implement multiway join algorithms, a join tree (or a Map-Reduce-Merge workflow) is needed. According to the data flow of Figures 5 and 6, we can find that using sort-merge join can decrease the remote reads than nested-loop join. In Map-Reduce-Merge, sort-merge join is more efficient than nested-loop join when processing equal join. When processing more complicated join, the nested-loop join algorithm is in need.

7. Conclusions

Map-Reduce-Merge is an improved work based on Google's Map-Reduce programming model. It improves the ability to express and to process join operation among multiple heterogeneous datasets. At the same time, it increases the complexity to understand the execution flow of a job. This paper presents a rigorous description of Map-Reduce-Merge to abstract the fundamental functions for Map-Reduce-Merge using Haskell. Based on the abstract description, we analyze the characteristics of Map-Reduce-Merge programming model. On one hand, our work can help with

an unambiguous understanding of Map-Reduce-Merge and provide strong theoretical basis for designing more efficient parallel programming model to process join operation. On the other hand, programmers can use our description as a specification in software development. Our result can ensure the correctness and robustness of the software with Haskell strong type checking and type inference.

Since this paper mainly concentrates on describing the dataflow in Map-Reduce-Merge, an important future direction is to introduce some control parameters into our description to improve its flexibility and usability. In addition, cost information can cooperate with our description to estimate the performance of a Map-Reduce-Merge job.

Acknowledgments

This work was supported by the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant no. 20120061120059; the China Postdoctoral Science Foundation under Grant no. 2011M500612; the Key Program for Science and Technology Development of Jilin Province of China under Grant no. 20130206052GX; the National Natural Science Foundation of China under Grant nos. 61300049, 61070084, and 61100090; the Program for New Century Excellent Talents in University under Grant no. NCET-10-0436; the Science and Technology Development Program of Jilin Province of China under Grant no. 20101509; the Fundamental Research Funds for the Central Universities under Grant nos. 201103124, 201103133; and the "985 Project" of Jilin University of China.

References

- [1] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.
- [2] P. Wang, D. Meng, J. Zhan, and B. Tu, "Review of programming models for data-intensive computing," *Journal of Computer Research and Development*, vol. 47, no. 11, pp. 1993–2002, 2010.
- [3] J.-J. Li, J. Cui, D. Wang, L. Yan, and Y.-S. Huang, "Survey of MapReduce parallel programming model," *Chinese Journal of Electronics*, vol. 39, no. 11, pp. 2635–2642, 2011.
- [4] Y. Li, Y. Wang, and L. Bao, "FACC: a novel finite automaton based on cloud computing for the multiple longest common subsequences search," *Mathematical Problems in Engineering*, vol. 2012, Article ID 310328, 17 pages, 2012.

- [5] X. Wang, Y. Wang, and H. Zhu, “Energy-efficient multi-job scheduling model for cloud computing and its genetic algorithm,” *Mathematical Problems in Engineering*, vol. 2012, Article ID 589243, 16 pages, 2012.
- [6] H.-C. Yang, A. Dasdan, R.-L. Hsiao, and D. S. Parker, “Map-reduce-merge: simplified relational data processing on large clusters,” in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 1029–1040, Beijing, China, June 2007.
- [7] J. Wang, T. Wang, D. Yang, and H. Li, “A filter-based multi-join algorithm in cloud computing environment,” *Journal of Computer Research and Development*, vol. 48, supplement, pp. 245–253, 2011.
- [8] R. Vernica, M. J. Carey, and C. Li, “Efficient parallel set-similarity joins using MapReduce,” in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 495–506, Indianapolis, Ind, USA, June 2010.
- [9] D. Jiang, A. K. H. Tung, and G. Chen, “MAP-JOIN-REDUCE: toward scalable and efficient data analysis on large clusters,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 9, pp. 1299–1311, 2011.
- [10] A. Okcan and M. Riedewald, “Processing theta-joins using MapReduce,” in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 949–960, Athens, Greece, June 2011.
- [11] S. Blanas, J. M. Patel, V. Ercegovac, J. Rao, E. J. Shekita, and Y. Tian, “A comparison of join algorithms for log processing in MaReduce,” in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 975–986, Indianapolis, Ind, USA, June 2010.
- [12] X.-P. Qin, H.-J. Wang, X.-Y. Du, and S. Wang, “Big data analysis—competition and symbiosis of RDBMS and MapReduce,” *Journal of Software*, vol. 23, no. 1, pp. 32–45, 2012.
- [13] S. Wang, H.-J. Wang, X.-P. Qin, and X. Zhou, “Architecting big data: challenges, studies and forecasts,” *Chinese Journal of Computers*, vol. 34, no. 10, pp. 1741–1752, 2011.
- [14] S.-L. Gui, L. Luo, Y. Li, M. Yu, and J.-H. Xu, “Schedulability analysis tool for distributed real-time systems based on automata theory,” *Journal of Software*, vol. 22, no. 6, pp. 1236–1251, 2011.
- [15] J. Wang, M. Yin, and W. Gu, “Fuzzy multiset finite automata and their languages,” *Soft Computing*, vol. 17, no. 3, pp. 381–390, 2013.
- [16] X. Liu, L. Yang, M.-X. Pan, and L.-Z. Wang, “Scenario-driven service behavior manipulation,” *Journal of Software*, vol. 22, no. 6, pp. 1185–1198, 2011.
- [17] W. Gu, G. Li, and M. Yin, “Extending fuzzy soft sets with fuzzy description logics,” *ICIC Express Letters, Part B*, vol. 2, no. 5, pp. 1001–1007, 2011.
- [18] R. Lämmel, “Google’s MapReduce programming model—revisited,” *Science of Computer Programming*, vol. 70, no. 1, pp. 1–30, 2008.
- [19] R. Pike, S. Dorward, R. Griesemer, and S. Quinlan, “Interpreting the data: parallel analysis with Sawzall,” *Scientific Programming*, vol. 13, no. 4, pp. 277–298, 2005.
- [20] F. Palmieri, L. Buonanno, S. Venticinque, R. Aversa, and B. Di Martino, “A distributed scheduling framework based on selfish autonomous agents for federated cloud environments,” *Future Generation Computer Systems*, vol. 29, no. 6, pp. 1461–1472, 2013.
- [21] J. L. Lucas-Simarro, R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, “Scheduling strategies for optimal service deployment across multiple clouds,” *Future Generation Computer Systems*, vol. 29, no. 6, pp. 1431–1441, 2013.
- [22] D. Cai and M. Yin, “On the utility of landmarks in SAT based planning,” *Knowledge-Based Systems*, vol. 36, pp. 146–154, 2012.
- [23] D. Cai, J. Sun, and M. Yin, “Conformant planning heuristics based on plan reuse in belief states,” in *Proceedings of the 23rd AAAI Conference on Artificial Intelligence and the 20th Innovative Applications of Artificial Intelligence Conference*, pp. 1780–1781, Chicago, Ill, USA, July 2008.
- [24] C. Zhang and J. Sun, “An alternate two phases particle swarm optimization algorithm for flow shop scheduling problem,” *Expert Systems with Applications*, vol. 36, no. 3, pp. 5162–5167, 2009.
- [25] C. Zhang, J. Ning, and D. Ouyang, “A hybrid alternate two phases particle swarm optimization algorithm for flow shop scheduling problem,” *Computers and Industrial Engineering*, vol. 58, no. 1, pp. 1–11, 2010.
- [26] B. O’Sullivan, J. Goerzen, and E. Don Stewart, *Real World Haskell*, O’Reilly Media, 2008.
- [27] F. Chang, J. Dean, S. Ghemawat et al., “Bigtable: a distributed storage system for structured data,” *ACM Transactions on Computer Systems*, vol. 26, no. 2, article 4, 2008.

Research Article

Optimization of p -Cycle Placement for Differentiated Levels of Protection

Hong Hui Li and Xue Liang Fu

College of Computer and Information Engineering, Inner Mongolian Agricultural University, Hohhot 010018, Inner Mongolia, China

Correspondence should be addressed to Xue Liang Fu; fxliang@126.com

Received 18 May 2013; Accepted 29 July 2013

Academic Editor: William Guo

Copyright © 2013 H. H. Li and X. L. Fu. 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 develops a new scalable and efficient model for the design of p -cycles with the differentiated levels of node protection. The proposed model allows the indicated level of node survivability ranging from 0% to 100%, which could facilitate a carrier offer node-failure survivability (and hence availability) on a differentiated service basis. To design p -cycles, an integer linear program (ILP) is usually formulated with the prerequisite of a prior enumeration of all possible p -cycle candidates. A huge number of candidates may exist in a large-scale network. Thus, the resulting ILP becomes intractable. We propose a new design and solution method based on large-scale optimization techniques, known as column generation (CG). With CG, our design method generates p -cycle candidates dynamically when needed. Extensive experiments have been conducted for evaluation. The numerical results show that, with the spare capacity used only for link protection, up to 50% node-failure survivability can be achieved for free. Full node protection can be achieved at a marginal cost in comparison with those for link protection only.

1. Introduction

WDM (wavelength division multiplexing) mesh networks are preferred as next generation optical core networks [1]. The WDM technology enables a single fiber to carry up to hundreds of wavelength channels, each of which operates at a speed of several Gbps. Thus, a single network element failure may cause the failure of a number of light paths, which leads to the huge data as well as revenue loss. Survivability therefore is an important requirement in the design of WDM mesh networks.

Different kinds of protection approaches have been proposed for WDM optical mesh networks to ensure survivability from any single failure. Among them, p -cycles (short for preconfigured protection cycle) [2] have attracted extensive research interests due to their unique characteristics, that is, fast recovery speed and efficient capacity usage. Most of studies on p -cycle-based survivable network design [2–7] only consider protection against a single link failure with no consideration of a single node failure. A single node failure happens less frequently than a single link failure in WDM mesh networks. However, the resulting effects may be catastrophic as a single node failure disrupts all its incident links.

Moreover, according to service level agreements, customers may require differentiated levels of node protection. Thus, partial protection of paths from a single node failure may be cost effective and an attractive offering of a differentiated business.

So far, however, very few papers have explored p -cycle design with the differentiated levels of node protection. Node-encircling p -cycles (NEPC) are proposed in [8] for protection against any single node failure. An enhanced APS for p -cycles with node protection is proposed in [9]. Path-segment-protecting p -cycles are suggested in [10] for node protection. A very recent finding regarding p -cycles for node protection is reported in [11]. In order to keep the simplicity of p -cycle protection switching, the authors propose the two-hop-segment strategy for node protection [12] (see Section 2).

To design p -cycles with node protection, the conventional way (e.g., [9, 10]) is to formulate the design problem as integer linear program (ILP). The prerequisite for the ILP solution is a prior enumeration of all possible p -cycle candidates. The number of p -cycle candidates increases exponentially with the increase of network size (the number of links). Thus, the resulting ILP becomes less tractable in large-scale optical

networks. If only a subset of p -cycles candidates is considered, solution quality may not be guaranteed.

In this paper, we explore p -cycle design with the differentiated levels of node protection using the two-hop-segment strategy. The objective is to minimize spare capacity cost such that a network can survive from any single link failure as well as survive from the required level of any single node failure. (Note that for a single node failure, only the working paths passing through it are considered for protection.) In contrast with the protection approach in [12], the proposed approach [13, 14] enhances the p -cycle protection capability. A p -cycle can protect multiple affected paths upon an intermediate node failure. To deal with the scalability issues suffered in the conventional designs, we develop a new design method based on large-scale optimization tools, namely, column generation (CG). With CG, our proposed design generates p -cycle candidates dynamically when needed in the course of the optimization process.

The rest of the paper is organized as follows. Section 2 illustrates the principle of p -cycles for node protection. Section 3 presents the CG formulation developed for the design of p -cycles with differentiated levels of node protection. Section 4 evaluates the solution performance of the proposed design. Conclusions are made in Section 5.

2. Background

In this section, we illustrate the two-hop-segment strategy for node protection using p -cycles. p -Cycles are originally proposed for link protection. The very recent study [12] suggests the two-hop-segment strategy, with which p -cycles can protect from a single node failure. For an intermediate node of a working path, the related two-hop segment includes two links on the path which are adjacent to the node. If a p -cycle protects the two links of a two-hop segment, a p -cycle can protect the associated intermediate node in an on-cycle protection manner. In this on-cycle manner, a p -cycle provides one protection path for the recovery of one unit traffic through the failed node. If only two end nodes of a two-hop segment are sitting on a p -cycle, the p -cycle can offer two protection paths. This protection manner is named as a straddling protection manner.

Figure 1 illustrates the principal of node protection with p -cycles, where a demand $s-d$ is carried on working path $s-a-b-c-d$. For the intermediate node b , the associated two-hop segment consists in $a-b-c$. To protect node b , the associated p -cycle must protect the two-hop segment $a-b-c$ wholly. As shown in Figure 1(a) to Figure 1(d), each p -cycle, from c_1 to c_4 , can be used to protect the intermediate node b . Among them, Figure 1(a) to Figure 1(c) show the on-cycle protection manner, while Figure 1(d) shows the straddling protection manner.

3. A Column Generation Model

An optical network is denoted by a graph $G = (V, L)$, where V is the set of nodes and L is the set of links, indexed by v and ℓ , respectively.

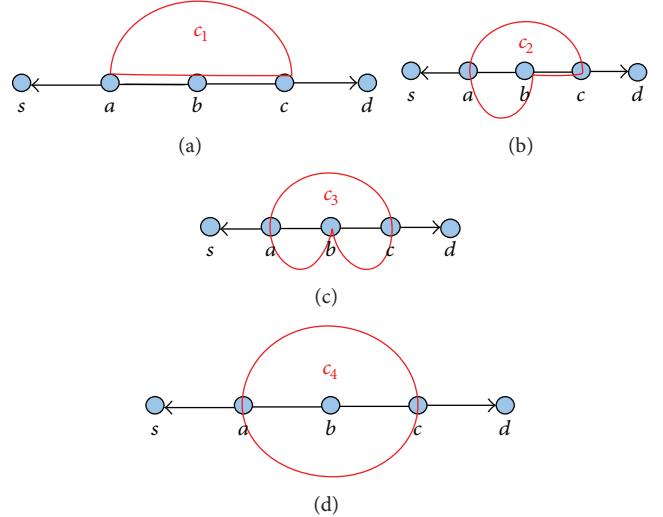


FIGURE 1: Two-hop-segment strategy for node protection with p -cycles.

In order to deal with the scalability issues in the classical design, we develop a column-generation- (CG-) based optimization model. Only a very few number of promising p -cycle candidates are generated online as needed in the course of the optimization process. With the CG technique (see, e.g., [15] for basic reference), the original design problem is decomposed into two problems: the master problem and the pricing problem. The master problem chooses p -cycles from candidates in order to protect all traffic against a single link failure or a single node failure according to the related quality of protection of node. A p -cycle candidate is generated dynamically by the pricing problem in each iteration of the CG algorithm.

3.1. The Master Problem. We denote the number of traffic units on link ℓ by ω_ℓ . For a given working path $p \in P$, let d_p be the number of connection requests carried on it and let V_p be its intermediate node set. Let Φ be the QoP of the intermediate nodes of the working path p .

Candidates are associated with the configuration set C . A cycle configuration c consists in a cycle that protects a set of links and a set of working paths against the failure of the associated intermediate nodes. A cycle configuration c is associated with a vector $(a_\ell^c)_{\ell \in L}$ and a matrix $(a_{pv}^c)_{p \in P, v \in V_p}$. The element $a_\ell^c \in \{2, 1, 0\}$ denotes the number of protection paths for link ℓ provided by the configuration c . The element $a_{pv}^c \in \{2, 1, 0\}$ represents the number of protection paths provided by configuration c for working path p against the failure of its relay node v . Let COST^c be the spare capacity cost of configuration c , which relies on the number of the on-cycle links.

Variables z^c denote the number of configuration c copies that are selected in the current solution. The design objective is to minimize the total spare capacity usage to protect all links as well as meet the required QoP for nodes.

The mathematical model can then be written as follows:

$$\min \sum_{c \in C} \text{COST}^c z^c \quad (1)$$

$$\text{subject to : } \sum_{c \in C} a_\ell^c z^c \geq \omega_\ell \quad \ell \in L \quad (2)$$

$$\sum_{c \in C} a_{pv}^c z^c \geq \Phi d_p \quad (3)$$

$$p \in P, \quad v \in V_p$$

$$z^c \in \mathbb{Z}^+ \quad c \in C. \quad (4)$$

Constraints (2) ensure that all traffic units are protected against any single link failure. Constraints (3) ensure that all demands are protected against a single node failure according to the associated level of QoP. Constraints (4) are variable domain constraints.

3.2. The Pricing Problem. The pricing problem corresponds to the optimization problem for generation of the promising p -cycle candidates. The pricing problem is the same as the one in [13, 14] as the same protection approach is adapted for node protection with p -cycles. The objective is to minimize the so-called reduced cost of the master problem and subject to sets of constraints.

The reduced cost can be written as follows:

$$\overline{\text{COST}}^c = \text{COST}^c - \sum_{\ell \in L} u_\ell a_\ell^c - \sum_{p \in P} \sum_{v \in V_p} u_{pv} a_{pv}^c, \quad (5)$$

where u_ℓ and u_{pv} are dual variables associated with constraints (2) and constraints (3), respectively.

Let us next define the following notations.

Sets and Parameters. $\omega(v)$ is the set of links adjacent to a node v . $\omega(V')$ is the set of links with one end node in the set V' but the other end node not. P_v is the set of working paths going through node v . $\tau_v^\ell = 1$ if link ℓ is adjacent to node v and on path p ; 0 otherwise. e_{pv}^1 and e_{pv}^2 are two end nodes of the 2-hop segment defined by working path p and its relay node v .

Variables. $b_\ell = 1$ if link ℓ sits on the current cycle; 0 otherwise. $s_\ell = 1$ if link ℓ straddles the current cycle; 0 otherwise. $y_v = 1$ if node v belongs to the current cycle; 0 otherwise. $x_{pv}^\ell = 1$ if link ℓ is used to protect a path p against the failure of its relay node v ; 0 otherwise.

With these notations, the objective function of the pricing problem can then be written as follows:

$$\min \sum_{\ell \in L} (\Lambda_\ell - u_\ell) b_\ell - 2 \sum_{\ell \in L} u_\ell s_\ell - \sum_{p \in P} \sum_{v \in V_p} u_{pv} \sum_{\ell \in \omega(e_{pv}^1)} x_{pv}^\ell. \quad (6)$$

Regarding constraints, there are two parts in the pricing problem. The first part defines a cycle, while the second part identifies the associated intermediate nodes of the working paths which can be protected by the current cycle.

In the following, the first part of constraints is presented:

$$\sum_{\ell \in \omega(v)} b_\ell = 2y_v \quad v \in V, \quad (7)$$

$$s_\ell \leq y_v - b_\ell \quad v \in V, \ell \in \omega(v), \quad (8)$$

$$s_\ell \geq y_v + y_{v'} - b_\ell - 1 \quad v, v' \in V, \ell = \{v, v'\} \in L, \quad (9)$$

$$\sum_{\ell \in \omega(V')} b_\ell \geq y_v + y_{v'} - 1 \quad V' \subset V, 3 \leq |V'| \leq |V| - 3 \quad (10)$$

$$v \in V', v' \in V \setminus V'.$$

Constraints (7) ensure that an on-cycle node holds two incident links on the cycle. Constraints (8) and (9) say that a link is a straddling link if the link is with two on-cycle end nodes but itself is not. Constraints (10) forbid generating a configuration with multiple cycles. Otherwise, it may complicate the determination of straddling links.

The second part of the constraints is next presented as follows:

$$x_{pv}^\ell \leq b_\ell \quad p \in P, v \in V_p, \ell \in L, \quad (11)$$

$$x_{pv}^\ell \leq 1 - \tau_v^\ell \quad p \in P, v \in V_p, \ell \in L, \quad (12)$$

$$\sum_{\ell \in \omega(e_{pv}^1)} x_{pv}^\ell = \sum_{\ell \in \omega(e_{pv}^2)} x_{pv}^\ell \quad p \in P, v \in V_p, \quad (13)$$

$$\sum_{\ell \in \omega(v')} x_{pv}^\ell \leq 2 \quad p \in P, v \in V_p, v' \in V, \quad (14)$$

$$\sum_{\ell \in \omega(v') \setminus \ell \neq \ell'} x_{pv}^\ell \geq x_{pv}^{\ell'} \quad p \in P, v \in V_p, \ell' \in \omega(v') \quad (15)$$

$$\sum_{p \in P_v} x_{pv}^\ell \leq 1 \quad \ell \in L, v \in V, \quad (16)$$

$$b_\ell, s_\ell, y_v, x_{pv}^\ell \in \{0, 1\} \quad \ell \in L, v \in V, p \in P. \quad (17)$$

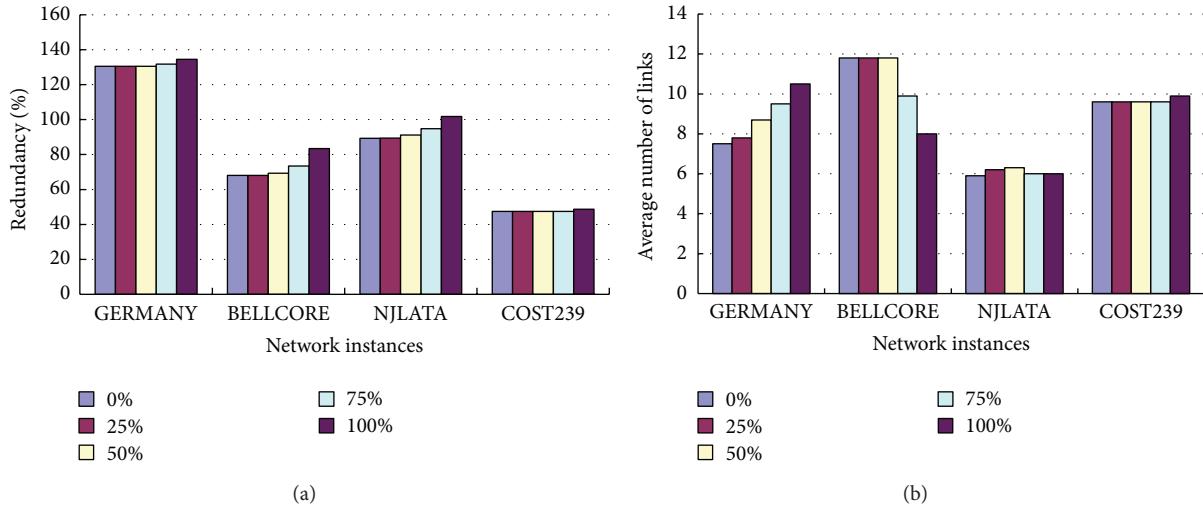
Constraints (11) say that only on-cycle links are eligible for protecting the intermediate nodes of the working path p . Constraints (12) say that a link cannot be used by the associated protection paths if the link is adjacent to the intermediate node v of the working path p . Constraints (13)–(15) are flow conservation constraints for defining the associated protection paths. The failure of a node affects all working paths through it. Constraints (16) say that, among them, only a disrupted working path can be recovered. The final set of constraints is variable domain constraints.

4. Computational Results

In this section, we evaluate the solution performances of our proposed design experimentally. Two metrics are exploited for evaluation, which include capacity redundancy (the ratio of spare capacity to working capacity [9]) and the average size

TABLE 1: Network instances.

Networks	Nodes	Links	Node degree	Num. demands	Working capacity
GERMANY [16]	17	26	3.1	136	4034
BELLCORE [17]	15	28	3.7	105	2610
NJLATA [17]	11	23	4.2	55	943
COST239 [18]	11	26	4.7	55	792

FIGURE 2: QoP p -cycle design.

(the average number of links) of the p -cycles selected in the final solution. Both design methods are implemented in C++ and solved by IBM CPLEX solver.

4.1. Date Instances. Four test networks are presented in Table 1. For each network, we provide network name, the number of nodes, the number of links, and the node degree for approximately representing connectivity. Also, we list the number of demands carried on each network and the associated working capacity cost. Each element in the associated traffic matrices represents the number of requests between each node pair, which is uniformly distributed on the interval $[1, \dots, 20]$.

4.2. Experimental Comparison. Experimental results regarding capacity redundancy and the average size of the p -cycles are shown in Figure 2. In Figure 2(a), for the indicated network instance, each bar corresponds to the capacity redundancy of an optimal solution for the indicated level of node-failure recovery ratio (i.e., Φ values). Experimental results reveal that, with our proposed QoP design, p -cycles can provide 100% node-failure survivability (i.e., $\Phi = 1$) at marginal additional redundant spare capacity compared with those only for link protection ($\Phi = 0$), which ranges from $\sim 1\%$ to $\sim 15\%$. The denser the network is, the less the redundant spare capacity is required for full node protection compared with that only for link protection. In addition, we can observe that, with the same spare capacity as the p -cycles designed for link protection, 25% node-failure survivability

can be achieved for free for each network instance. For GERMANY and COST239, up to 50% node-failure survivability can be achieved with the spare capacity used for the provision of link-protecting p -cycles.

Figure 2(b) shows the solution structure of the QoP p -cycle design, where each bar indicates the average number of links of optimally selected p -cycles for the indicated network instance and the indicated level of node-failure survivability. We can observe that, for network instance GERMANY, the higher the level of node-failure survivability is, the larger the average size of p -cycles is. However, the trends do not appear in network instances NJLATA and BELLCORE. For the instance COST239, due to its high density of the network topology, there are many chances existing to build the same size (the same number) of p -cycles while holding a different node-protection capability. As a result, the average size of optimally designed p -cycles keeps invariable upon the QoP level (Φ value) no more than 75%.

5. Conclusion

In this paper, we investigated the design of p -cycles with the differentiated levels of the node-failure survivability in WDM mesh networks. The main goal is to develop a capacity-efficient and scalable design method. Our proposed design method enhances the p -cycle protection capabilities. A p -cycle can protect multiple affected paths upon a node failure. In order to manage the scalability issue suffered in the conventional design method, this paper exploited the large-scale optimization techniques, that is, column generation

(CG). Based on CG, our proposed method generates the p -cycle candidates dynamically as needed in the course of the optimization process.

Extensive experiments have been conducted. Experimental results show clearly that, with our proposed design method, p -cycles can provide full node protection at a marginal additional redundant capacity compared with those for link protection only. For some instances, up to 50% node-failure survivability can be achieved for free with spare capacity used only for link protection.

References

- [1] T. E. Stern and K. Bala, *Multiwavelength Optical Networks: A Layered Approach*, Prentice-Hall, 1999.
- [2] W. D. Grover and D. Stamatelakis, "Cycle-oriented distributed preconfiguration: ring-like speed with mesh-like capacity for self-planning network restoration," in *Proceedings of the IEEE International Conference on Communications (ICC '98)*, vol. 1, pp. 537–543, June 1998.
- [3] D. A. Schupke, C. G. Gruber, and A. Autenrieth, "Optimal configuration of p -cycles in WDM networks," in *Proceedings of the International Conference on Communications (ICC '02)*, vol. 5, pp. 2761–2765, May 2002.
- [4] L. Chang and R. Lu, "Finding good candidate cycles for efficient p -cycle network design," in *Proceedings of the 13th International Conference on Computer Communications and Networks (ICCCN '04)*, pp. 321–326, October 2004.
- [5] C. Rocha and B. Jaumard, "Revisiting p -cycles / FIPP p -cycles vs. shared link / path protection," in *Proceedings of the 17th International Conference on Computer Communications and Networks (ICCCN '08)*, pp. 447–452, August 2008.
- [6] K. Lo, D. Habibi, Q. V. Phung, A. Rassau, and H. N. Nguyen, "Efficient p -cycles design by heuristic p -cycle selection and refinement for survivable WDM mesh networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '06)*, pp. 1–5, December 2006.
- [7] B. Wu, K. L. Yeung, and P. Ho, "ILP formulations for p -cycle design without candidate cycle enumeration," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 284–295, 2010.
- [8] D. Stamatelakis and W. D. Grover, "IP layer restoration and network planning based on virtual protection cycles," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1938–1949, 2000.
- [9] D. A. Schupke, "Automatic protection switching for p -cycles in WDM networks," *Optical Switching and Networking*, vol. 2, no. 1, pp. 35–48, 2005.
- [10] G. Shen and W. D. Grover, "Extending the p -cycle concept to path segment protection for span and node failure recovery," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 8, pp. 1306–1319, 2003.
- [11] D. P. Onguetou and W. D. Grover, "A new insight and approach to node failure protection with ordinary p -cycles," in *Proceedings of the IEEE International Conference on Communications (ICC '08)*, pp. 5145–5149, May 2008.
- [12] W. D. Grover and D. P. Onguetou, "A new approach to node-failure protection with span-protecting p -cycles," in *Proceedings of the 11th International Conference on Transparent Optical Networks (ICTON '09)*, pp. 1–5, July 2009.
- [13] B. Jaumard and H. Li, "Design of p -cycles for full node protection in WDM Mesh networks," in *Proceedings of the IEEE International Conference on Communications (ICC '11)*, pp. 1–5, June 2011.
- [14] H. Li, *p-cycle based protection in WDM mesh networks [Ph.D. dissertation]*, Concordia University, 2013.
- [15] V. Chvátal, *Linear Programming*, Freeman, New York, NY, USA, 1983.
- [16] R. Hlsermann, "A set of typical transport network scenarios for network modelling," Tech. Rep., Beitrge zur 5. ITG-Fachtagung Photonische Netze, Leipzig, Germany, 2004.
- [17] K. Murakami and H. S. Kim, "Comparative study on restoration schemes of survivable ATM networks," in *Proceedings of the 16th IEEE Annual Conference on Computer Communications (INFOCOM '97)*, pp. 345–352, IEEE Computer Society, Washington, DC, USA, April 1997.
- [18] P. Batchelor et al., "Ultra high-capacity optical transmission networks: final report of action COST 239," Tech. Rep., Faculty of Electrical Engineering and Computing, University of Zagreb, 1999.

Research Article

An Accurate and Efficient Method to Predict Y-NO Bond Homolysis Bond Dissociation Energies

Hong Zhi Li,^{1,2} Lin Li,¹ Zi Yan Zhong,¹ Yi Han,¹ LiHong Hu,¹ and Ying Hua Lu^{1,3}

¹ School of Computer Science and Information Technology, Northeast Normal University, Changchun 130017, China

² School of Life Sciences, Northeast Normal University, Changchun 130024, China

³ Institute of Functional Material Chemistry, Faculty of Chemistry, Northeast Normal University, Changchun 130024, China

Correspondence should be addressed to LiHong Hu; lhhu@nenu.edu.cn and Ying Hua Lu; luyh@nenu.edu.cn

Received 6 June 2013; Accepted 23 July 2013

Academic Editor: William Guo

Copyright © 2013 Hong Zhi 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.

The paper suggests a new method that combines the Kennard and Stone algorithm (Kenstone, KS), hierarchical clustering (HC), and ant colony optimization (ACO)-based extreme learning machine (ELM) (KS-HC/ACO-ELM) with the density functional theory (DFT) B3LYP/6-31G(d) method to improve the accuracy of DFT calculations for the Y-NO homolysis bond dissociation energies (BDE). In this method, Kenstone divides the whole data set into two parts, the training set and the test set; HC and ACO are used to perform the cluster analysis on molecular descriptors; correlation analysis is applied for selecting the most correlated molecular descriptors in the classes, and ELM is the nonlinear model for establishing the relationship between DFT calculations and homolysis BDE experimental values. The results show that the standard deviation of homolysis BDE in the molecular test set is reduced from 4.03 kcal mol⁻¹ calculated by the DFT B3LYP/6-31G(d) method to 0.30, 0.28, 0.29, and 0.32 kcal mol⁻¹ by the KS-ELM, KS-HC-ELM, and KS-ACO-ELM methods and the artificial neural network (ANN) combined with KS-HC, respectively. This method predicts accurate values with much higher efficiency when compared to the larger basis set DFT calculation and may also achieve similarly accurate calculation results for larger molecules.

1. Introduction

In the past few decades, quantum chemistry has attracted remarkable attention and made significant improvements along with the corresponding fields of physics, mathematics, and computer science [1–7]. Quantum chemical methods have captured the physical essence of molecules through numerous research efforts, to the point where various molecular properties (including geometry, dipole moment, polarizability, thermodynamics, and excited states) can be obtained without experiments. However, quantum chemical calculations for large molecules can be quite demanding, in particular for high-accuracy calculations that need to solve the Schrödinger equation thoroughly. Therefore, accuracy often has to be sacrificed for practical reasons, for example, introducing approximations when solving the wave function for large molecular systems. Despite these approximations, the size of molecule is still limited. For a large system with a regular structure, the quantum chemical methods can only give a qualitative description. For complex systems, it is hard

to calculate with existing quantum chemical methods; even if the calculation can be carried out, the accuracy might be very low and the error is far beyond the specifications in scientific research [8, 9]. Therefore, alternative methods for large or complex systems need to be developed to meet the challenges of molecular calculations.

Artificial intelligence methods can resolve the accuracy problem of quantum chemical calculations for molecular bond properties by establishing the relationship between the calculated and experimental values instead of strictly solving wave functions with very high cost. By this method, high-precision calculation results can be obtained without wasting time on advanced quantum chemical methods and ultra-large basis sets because this combination-type calculation method can take mutual advantages of two methods to reduce the systematic errors induced by defects of theories and functions with simple physical parameters adopted in artificial intelligence methods. Thus quantum chemical calculations with low-level methods and small basis sets are sufficient for obtaining valid data from artificial intelligence methods.

Meanwhile, small basis sets enable quantum chemical calculations to apply to larger molecules, further increasing the chance to design novel lead molecules.

Although the artificial intelligence method is a relatively new multidisciplinary research field, in the past decade, it has received a lot of attention and thus has developed rapidly. The combination strategy consists of the quantum chemical method first being used to obtain the molecular properties and then take the quantum chemical calculation results as inputs for the statistical methods to establish the relationship between the experimental and calculation values. There are many choices on statistical methods including linear methods such as linear regression [10–14], nonlinear methods such as neural networks [15–37], and support vector machines [38–40]. Usually, the linear regression method is simple and intuitive, but problems in molecular systems are mainly nonlinear; for the same descriptor inputs, neural networks can better solve complicated nonlinear problems that might be too difficult to be modeled by mathematical equations [27]. However, the traditional neural networks have the following major drawbacks. (1) The speed of training is slow. The gradient descent method generally takes multiple iterations to converge, so the training process takes a long time. (2) Because it is easy to fall into a local minimum, the global minimum can be missed during the calculation. (3) The selection of learning rate η is sensitive. The learning rate η has a great impact on the performance of the neural network, so the proper η must be chosen to obtain a stable network. If η is too small, the rate of algorithm convergence is very slow and the training process takes a long time; on the other hand, if it is too large, the training process might be unstable (divergent). Therefore, it is important to explore a training algorithm where the learning rate is comparably high, which increases the possibility of acquiring the global optimum solution. This leads to better generalization for the prediction. The extreme learning machine (ELM) method avoids these defects, thus exhibiting a better performance than neural networks [3, 4].

Nitric oxide (NO) plays an important physiological role in the human life cycle [41–50]. It is difficult to accurately obtain homolysis BDE values in NO carrier molecules due to the complexity of the experiments. Professor Cheng's group recently devoted considerable efforts to measuring Y-NO bond homolysis BDE in solvents [51–61], which provides a great opportunity for designing NO molecule carriers in silico. Hence, we herein use artificial intelligence and DFT to achieve high-accuracy homolysis BDE calculation results to better design potentially novel NO carriers.

The content of this paper is arranged as follows. First, it describes the HC, ACO, and ELM methods. Second, it illustrates the technology roadmap of the KS-HC/ACO-ELM method. Third, it discusses the calculation results of the Kenstone, HC, ACO, and ELM methods (i.e., the appropriate classification of the training set and the test set, clustering analyses for molecular descriptors with both the known and unknown class numbers, and the establishment of the ELM model, resp.). Finally, the results and the discussions are summarized.

2. Methods

2.1. Hierarchical Clustering (the Known Number). The basic idea of the hierarchical clustering (HC) method is to classify samples in terms of distances. Firstly, it defines the distance between samples and classes and combines the nearest two classes into classes of samples, and then it recalculates the distance between the new class and other classes and classifies them according to the minimum distance. By reducing one class at a time, this process is repeated until all of the samples come into one class. The chart showing the clustering process is called a clustering chart. One of the features of the HC method is that when samples are classified into a class in a certain level, they will always belong to the same class in the subsequent divisions.

There are a variety of methods to define the distance between classes. Different definitions generate different HC analysis methods, which include the following: the shortest-distance method, the longest-distance method, the middle-distance method, the center of gravity method, the group-average method, the variable group-average method, the variable method, and the variance and sum method. The recurrence formula for these methods is

$$D_{ir}^2 = \alpha_p D_{ip}^2 + \alpha_q D_{iq}^2 + \beta D_{pq}^2 + r |D_{ip}^2 - D_{iq}^2|, \quad (1)$$

where D_{ij} is the distance between classes. The values of parameters α , β , and r vary according to the different methods and are adopted in the program. The default values of the parameters are used in the calculations.

2.2. Ant Colony Optimization (the Unknown Number). One of the clustering problems is that the number of clusters is unknown. In the ant colony optimization (ACO) clustering method, the samples are thought of as ants with different attributes, and the cluster center is the food sources that the ants seek. Therefore, the sample clustering is the mechanism of ants looking for food sources.

If X is a data set to be classified and N is the feature set of the samples, the clustering algorithm is as follows.

- (1) Initially, allocate N samples into N classes because one sample belongs to one class.
- (2) Calculate Euclidean distances between classes w_i and w_j :

$$d_{ij} = \sqrt{\sum_{k=1}^n (\overline{X_k^{w_i}} - \overline{X_k^{w_j}})^2}, \quad (2)$$

$$\overline{X^{w_i}} = \frac{1}{N} \sum_{k=1}^{N_i} X_k, \quad (3)$$

where d_{ij} represents the Euclidean distance between classes i and j , $\overline{X^{w_i}}$ is the cluster center vector, and N_i is the number of samples in class w_i .

- (3) Calculate the amount of pheromone on each path. Let r be the radius of the cluster. $\tau_{ij}(t)$ is the residual

pheromone at time t on the path from w_i to w_j . The pheromone on the path (i, j) is

$$\tau_{ij}(t) = \begin{cases} 1, & d_{ij} \leq r, \\ 0, & d_{ij} > r, \end{cases} \quad (4)$$

where $r = A + d_{\min} + (d_{\max} - d_{\min}) \cdot B$ and A and B are constants.

(4) Calculate the probability of w_i merging into w_j :

$$p_{ij}(t) = \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{s \in S} \tau_{is}^\alpha(t) \eta_{is}^\beta(t)}, \quad (5)$$

where $S = \{s \mid d_{ij} \leq r, s = 1, 2, \dots, j-1, j+1, \dots, M\}$, s stands for a number of a certain class, S represents a collection of all numbers of classes, where the distance is less than or equal to r in the class j , M is the total number of classes, and $\eta_{is}(t)$ is the weight parameters.

(5) If $p_{ij}(t) \geq p_0$, then w_i is subsumed in the w_j , the number of classes minus 1. p_0 is a given probability value. Recalculate the merged cluster center.

(6) Determine whether there is merging or not. If there is no merging, stop the cycle; otherwise, go to step (2) and continue the iteration.

2.3. Extreme Learning Machine. The extreme learning machine (ELM) was proposed by Huang et al. in 2006 [62]. The core principle of ELM is a single-hidden layer feed-forward neural network (SLFN), a typical structure of which is shown in Figure 1. This network consists of an input layer, a hidden layer, and an output layer. The neurons in the former two and latter two are fully connected. There are n neurons corresponding to the input variables in the input layer; there are l neurons in hidden layer; there are m neurons corresponding to m output variables in the output layer.

Briefly, suppose the connection weight ω between the input layer and the hidden layer is given by

$$\omega = \begin{bmatrix} \omega_{11} & \omega_{12} & \cdots & \omega_{1n} \\ \omega_{21} & \omega_{22} & \cdots & \omega_{2n} \\ \vdots & \vdots & & \vdots \\ \omega_{l1} & \omega_{l2} & \cdots & \omega_{ln} \end{bmatrix}_{l \times n}, \quad (6)$$

where ω_{ij} represents the weight between neuron i in the input layer and neuron j in the hidden layer.

The connection weight matrix β between the hidden layer and the output layer is given by

$$\beta = \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1m} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2m} \\ \vdots & \vdots & & \vdots \\ \beta_{l1} & \beta_{l2} & \cdots & \beta_{lm} \end{bmatrix}_{l \times m}, \quad (7)$$

where β_{jk} represents the weight value between the j neuron in the hidden layer and the k neuron in the output layer.

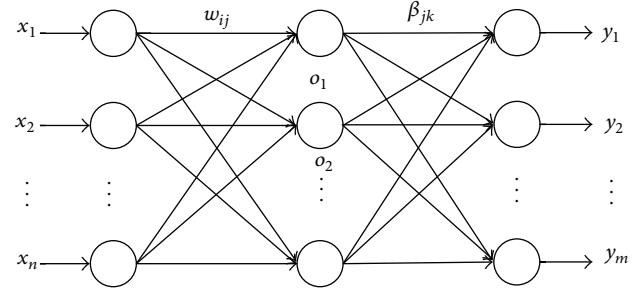


FIGURE 1: A typical structure of a single-hidden layer feed-forward neural network (SLFN).

The threshold b of the neuron in the hidden layer is given by

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_l \end{bmatrix}_{l \times 1}. \quad (8)$$

The input matrix X and the output matrix Y in the training set with Q samples are given by

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1Q} \\ x_{21} & x_{22} & \cdots & x_{2Q} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nQ} \end{bmatrix}_{n \times Q},$$

$$Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1Q} \\ y_{21} & y_{22} & \cdots & y_{2Q} \\ \vdots & \vdots & & \vdots \\ y_{m1} & y_{m2} & \cdots & y_{mQ} \end{bmatrix}_{m \times Q}. \quad (9)$$

The activation function of the neuron in the hidden layer is $g(x)$; the output of net T can be seen in Figure 1. Consider

$$T = [t_1, t_2, \dots, t_Q]_{m \times Q},$$

$$t_j = \begin{bmatrix} t_{1j} \\ t_{2j} \\ \vdots \\ t_{mj} \end{bmatrix}_{m \times 1} = \begin{bmatrix} \sum_{i=1}^l \beta_{1i} g(\omega_{1i} x_j + b_1) \\ \sum_{i=1}^l \beta_{2i} g(\omega_{2i} x_j + b_2) \\ \vdots \\ \sum_{i=1}^l \beta_{im} g(\omega_{im} x_j + b_m) \end{bmatrix}_{m \times l}, \quad (10)$$

$$j = 1, 2, \dots, Q,$$

where $\omega_i = [\omega_{i1}, \omega_{i2}, \dots, \omega_{in}]$, $x_j = [x_{1j}, x_{2j}, \dots, x_{nj}]^T$. Equation (10) can be expressed as

$$H\beta = T', \quad (11)$$

where T' is the transpose of the matrix T and H is the output matrix of the hidden layer of the neural network. Consider

$$H(w_1, w_2, \dots, w_l, b_1, b_2, \dots, b_l, x_1, x_2, \dots, x_Q)$$

$$= \begin{bmatrix} g(w_1x_1 + b_1) & g(w_2x_1 + b_2) & \cdots & g(w_lx_1 + b_l) \\ g(w_1x_2 + b_1) & g(w_2x_2 + b_2) & \cdots & g(w_lx_2 + b_l) \\ \vdots & \vdots & & \vdots \\ g(w_1x_Q + b_1) & g(w_2x_Q + b_2) & \cdots & g(w_lx_Q + b_l) \end{bmatrix}_{Q \times l}. \quad (12)$$

Because the number of neurons in the hidden layer is equal to the number of samples in the training set, SLFN can approach the training samples with zero error for any w and b ; that is,

$$\sum_{j=1}^Q \|t_j - y_j\| = 0, \quad (13)$$

where $y_j = [y_{1j}, y_{2j}, \dots, y_{mj}]^T$, $j = 1, 2, \dots, Q$

However, when the number of the training set Q is relatively large, the number of the hidden layer neurons K is usually smaller than Q to reduce the computational complexity. Meanwhile, the training error of SLFN can approach any $\varepsilon > 0$; that is,

$$\sum_{j=1}^Q \|t_j - y_j\| < \varepsilon. \quad (14)$$

Therefore, when the activation function $g(x)$ can be infinitely differentiable, not all of the parameters of SLFN need to be adjusted. The values of w and b can be chosen randomly before training and remain the same in the rest of the selection process. The connection weights β between the hidden layer and output layer can be obtained by solving the equations to obtain the least squares solution:

$$\min_{\beta} \|H\beta - T'\|, \quad (15)$$

the estimated solution is $\hat{\beta} = H^+T'$, where H^+ is the Moore-Penrose generalized inverse of the output matrix of the hidden layer H .

From the above analysis, ELM can randomly generate w and b before training. It is only needed to determine the number of hidden layer neurons and the activation function of the hidden layer neuron (infinitely differentiable), and then β can be calculated. Generally, ELM learning algorithms follow the next steps.

- (1) Determine the number of the neurons in the hidden layer and randomly set the connection weight w between the input layer and the hidden layer and the bias b of the neurons in the hidden layer.
- (2) Select an infinitely differentiable function as the activation function of the neurons in the hidden layer, and then calculate the matrix H of the hidden layer output.
- (3) Calculate the output layer weight $\hat{\beta}$: $\hat{\beta} = H^+T'$.

3. Technology Roadmap

The technology roadmap of the ELM method based on KS-HC/ACO to improve the accuracy of the molecular bond energies calculated by DFT is shown in Figure 2. It consists of three parts. Part one is the collection of the data, including molecular bond data and the molecular descriptors. The second part describes the theoretical calculation. Ninety-two important organic NO carrier molecules are studied, 12 molecular descriptors are calculated by DFT B3LYP/6-31G(d) (ΔH_{homo} , Q_Y , Q_N , Q_O , N_X , μ , α , E_{HOMO} , E_{LUMO} , $E_{\text{HOMO}-1}$, $E_{\text{LUMO}+1}$, and ΔE), and homolysis BDE experimental values are included [34]. The Kenstone method is used to divide the training set and the test set. HC and ACO are taken as different classification methods; the former requires a predefined number of the classes to select the appropriate molecular descriptors that will be modeled by the ELM method, while the latter does not need to specify the number of classes. The third part aims at testing and analyzing the models. When the calculation results are not satisfied, the algorithm should be optimized repeatedly up to a reasonable accuracy. At the same time, it is suggested to determine the comparison test.

4. Results and Discussion

4.1. Kenstone Calculation Results. The basic idea of the Kenstone method [63] is to divide the training and test sets according to the distance between two data points in the data set. First, the furthest points are put into the training set; the distances between the remaining points and the training data are then calculated. The minimum distances are retained while the maximum distance is selected and put into the training data one by one until the training set population reaches the desired number. While random assignment of the training and test sets can occasionally yield a better calculation result, it cannot guarantee the best result or an outlier-free outcome. For the Kenstone method, these problems are avoided, which means that the KS-HC/ACO-ELM model always yields good results and generalizability. In this study, 80 molecules selected with the Kenstone method are regarded as the training set and 12 molecules represent the test set. Molecules 13, 19, 21, 27, 31, 35, 36, 43, 55, 70, 78, and 81 are in the test set, and the rest are in the training set. It can be seen from the Kenstone calculation results that molecules 13, 19, 21, 27, 31, 35, 36, and 43 contain N-NO; molecule 55 contains O-NO; molecules 70 and 78 contain S-NO; and molecule 81 contains C-NO. This shows that the KS method can achieve diversity and balance for the training and test sets.

4.2. Hierarchical Clustering Calculation Results. It is critical for the HC to adopt the appropriate distance and clustering method. It is best if the method can exclude the dimensional effect, and usually the standard Euclidean distance is a reasonable choice. As for the clustering method, the shortest-distance method is too contracted and the longest-distance method is too expanded. These two methods are simple, but they belong to extreme classification methods. Therefore, the more appropriate methods are the average-distance method,

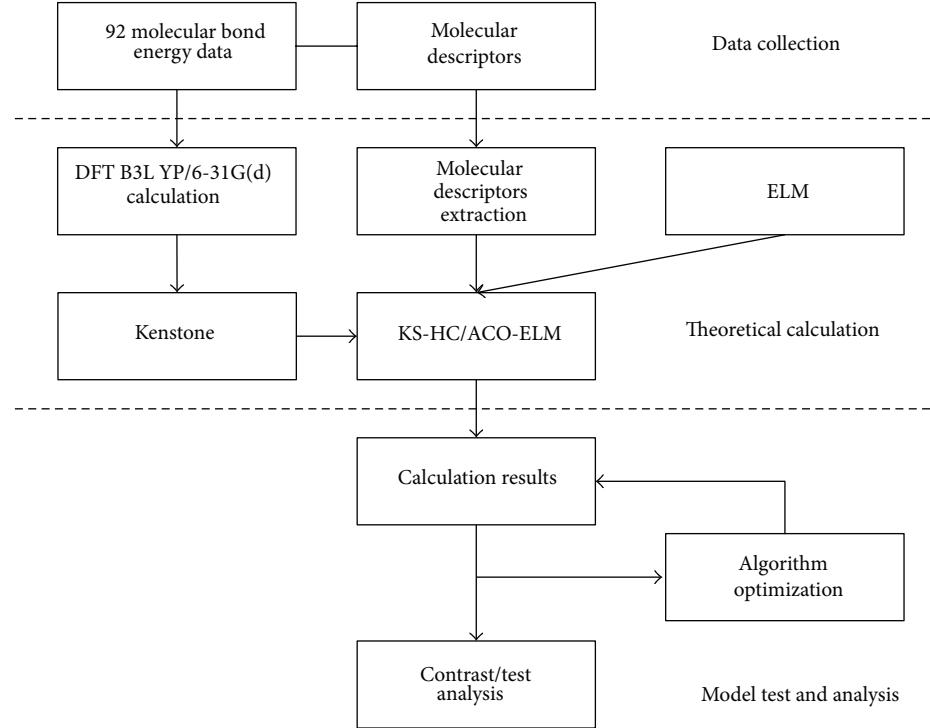


FIGURE 2: The flow chart of the KS-HC/ACO-ELM model.

TABLE 1: Cophenetic correlation coefficients of various clustering methods based on the Euclidean distance.

Clustering methods	Longest-distance method	Shortest-distance method	Average-distance method	Weighted-average method	Center of gravity method	Middle-distance method	Sum of squares
Cophenetic correlation coefficients based on the Euclidean distance	0.7726	0.6685	0.8381	0.8246	0.8113	0.7739	0.7995

the weighted average method, and the center of gravity method.

There is no existing standard for evaluating the quality of clustering results, and the application of the clustering method depends on the researchers' application skills and experiences with the classification objects. If the clustering method is applied properly, the critical scale and the distance matrix could be highly correlated. The correlation is calculated by cophenetic correlation coefficients, which can be used to evaluate the clustering results. The value of the cophenetic correlation coefficient is between zero and one. If the clustering is effective, the value is large. However, the high value is only statistically significant; it does not mean that the result is physically effective. Sometimes the result is not meaningful in practice.

The cophenetic correlation coefficients are shown in Table 1. The clustering methods based on the Euclidean distance include the longest-distance method, the shortest-distance method, the average-distance method, the weighted-average method, the center of gravity method, the middle-distance method, and the sum of squares.

Based on the average-distance method of the Euclidean distance, 12 molecular descriptors are clustered into classes, and the numbers of classes are given from 2 to 11. This

paper only takes for an example the 7 classes. The clustering pedigree chart of the average-distance method is shown in Figure 3. Numbers 1 through 12 on the horizontal axis in Figure 3 correspond to twelve molecular descriptors (ΔH_{homo} , Q_Y , Q_N , Q_O , N_X , μ , α , E_{HOMO} , E_{LUMO} , $E_{\text{HOMO}-1}$, $E_{\text{LUMO}+1}$, and ΔE). The clustering calculation results are (1, 7, 2, 7, 4, 6, 4, 5, 5, 5, 5, and 3), which indicate that ΔH_{homo} , Q_N , μ , and ΔE are clustered into classes 1, 2, 6, and 3, respectively; Q_Y and Q_O are clustered into class 7; N_X and α into class 4; and E_{HOMO} , E_{LUMO} , $E_{\text{HOMO}-1}$, and $E_{\text{LUMO}+1}$ into class 5. When the given numbers of the clustering classes are different, the analysis and principle are the same as those used in class 7.

4.3. Ant Colony Optimization Calculation Results. The HC method needs a predefined number of clusters, which includes every type of classification. The advantage is obvious when the sample (i.e., the number of molecule descriptors) is small, but when the sample is large, it is neither necessary nor practical to list every classification using an exhaustive method. Therefore, in this paper, an ACO without a predefined number of clusters is also applied to the screening of molecular descriptors to arrive at a generalized model of the ELM method for either a small or large number of descriptors.

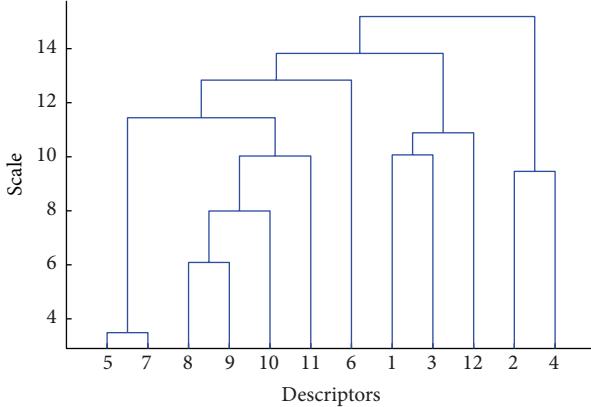


FIGURE 3: Clustering pedigree chart of the average-distance method.

According to the definition of the ACO, the optimization function is the minimum ratio of the within-class distance to interclass distance. This means the distance among the classes should be large, but the distance between two samples within a class should be small:

$$\min J(w, c) = \frac{\sum_{i=1}^{N_j} \sum_{p=1}^n w_{ij} \|x_{ip} - c_{jp}\|^2}{\sum_{j=1}^{M-1} \sum_{p=1}^n \|c_{j+1,p} - c_{jp}\|^2}, \quad (16)$$

where

$$c_{jp} = \frac{\sum_{i=1}^{N_j} w_{ij} x_{ip}}{\sum_{i=1}^{N_j} w_{ij}}, \quad j = 1, 2, \dots, M; \quad p = 1, 2, \dots, n, \quad (17)$$

$$w_{ij} = \begin{cases} 1, & \text{sample } i \text{ belongs to class } j, \\ 0, & \text{otherwise,} \end{cases}$$

where x_{ip} is the p factor of sample i and c_{jp} is the p factor of the center of the class j .

The results of the ACO calculation of the clustering molecular descriptors (ΔH_{homo} , Q_Y , Q_N , Q_O , N_X , μ , α , E_{HOMO} , E_{LUMO} , $E_{\text{HOMO}-1}$, $E_{\text{LUMO}+1}$, and ΔE) are (1, 2, 2, 2, 3, 4, 5, 6, 6, 6, 6, and 6), where the numbers in the bracket correspond to the classes of the molecular descriptors. From the calculation results it can be observed that twelve molecular descriptors can be divided into six classes: ΔH_{homo} is class 1, Q_Y , Q_N , and Q_O are class 2, N_X , μ , and α are classes 3, 4, and 5, respectively, and E_{HOMO} , E_{LUMO} , $E_{\text{HOMO}-1}$, $E_{\text{LUMO}+1}$, and ΔE are class 6.

4.4. ELM Calculation Results. The correlation coefficients between the 12 molecular descriptors and homolysis BDE experimental values were calculated, and they are 0.64, 0.46, 0.49, 0.02, 0.12, 0.18, 0.28, 0.51, 0.17, 0.43, 0.05, and 0.27. According to the magnitude of the correlation coefficients (e.g., by HC calculation) when Q_Y and Q_O cluster into a class, Q_Y is correlated higher with homolysis BDE than Q_O and will therefore be selected as the input of ELM. Similarly, when N_X and α are classified into one class, α will be selected as the input of ELM. In the cluster consisting of E_{HOMO} , E_{LUMO} , $E_{\text{HOMO}-1}$, and $E_{\text{LUMO}+1}$, E_{HOMO} is selected as the input of

ELM. ΔH_{homo} , Q_N , μ , and ΔE are clustered into one class. Therefore, ΔH_{homo} , Q_N , μ , ΔE , Q_Y , α , and E_{HOMO} are selected as the final inputs of ELM by using the HC method. In the same way, ΔH_{homo} , Q_N , N_X , μ , α , and E_{HOMO} are selected as the final inputs of ELM after using the ACO clustering method. The errors of the test set in different methods are shown in Figure 4(a).

To assess the KS-HC/ACO-ELM method, the results calculated by the molecular descriptors without screening KS-ELM (using all descriptors), KS-HC-ELM (best results are obtained when the molecular descriptors cluster into 7 classes), and KS-ACO-ELM are compared with the results calculated by B3LYP/6-31G(d). A three layered artificial neural network (KS-HC-ANN) calculation was also performed for comparison. The ANN structure uses the same number of inputs and 6 hidden neurons (shown in Figure 4(b)). The corrected results show that the systematic errors of the calculation value are eliminated, that the homolysis BDE are significantly improved by the KS-ELM, KS-HC-ELM, and KS-ACO-ELM methods, the range of error is narrowed down, the STD calculated by the KS-ELM, KS-HC-ELM, and KS-ACO-ELM methods is much smaller than the calculation error by DFT B3LYP/6-31G(d), the errors are in a Gaussian distribution, and the values of most of the errors are approximately 0. As is clearly seen in Figure 4(b), the results from KS-HC-ELM approach the homolysis BDE experimental values better than those obtained from the ANN, and the calculation time is also much less than that in the ANN because the ANN learning algorithm requires a number of network training parameters set by manual work. ANN also easily leads to a local optimal solution, and its calculated results are very unstable. The ELM method only requires setting the number of hidden layer nodes of the network. It does not need to adjust the network input weights and hidden bias in the implementation process of the algorithm, and it can generate a unique optimal solution, so it has both high learning speed and good generalization performance.

The STD corrected by the KS-ELM method is much smaller than the error of the B3LYP/6-31G(d), and the STD calculated by KS-HC-ELM and KS-ACO-ELM is smaller than that by KS-ELM. That means that it is necessary to screen molecular descriptors to eliminate redundant descriptors. The STD is reduced from 4.03 to 0.30, 0.28, 0.29, and 0.32 kcal mol⁻¹ after correction by the KS-ELM, KS-HC-ELM, KS-ACO-ELM, and KS-HC-ANN methods, respectively. It is quite simple to see that the result corrected by the KS-HC-ELM method is much closer to the experimental results. If some trivial features are introduced into ELM, the accuracy of the model might decrease, and if the parameters are selected inappropriately, overfitting could also appear. In this experiment, although the calculation results using the KS-HC-ELM method are superior to those by the KS-ACO-ELM method, it cannot be assumed that this will be the case under every circumstance because each method has its own advantages. The clustering methods can be divided into two categories, where one requires a predefined number of categories, while the other does not. Because the ELM inputs affect its performance and the best combination of molecular descriptors is difficult to determine, it is recommended

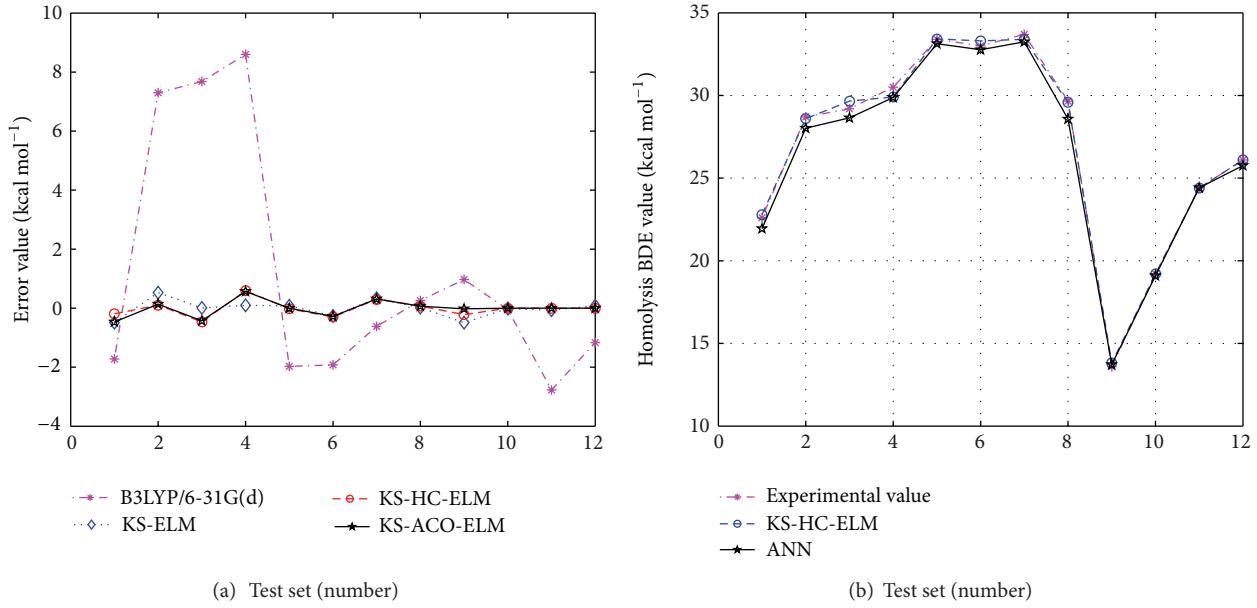


FIGURE 4: Errors of the calculated homolysis BDE and experimental values for the test set. (a) Errors calculated by the B3LYP/6-31G(d), KS-ELM (including all the molecular descriptors), KS-HC-ELM, and KS-ACO-ELM methods for twelve molecules in the test set. (\star), (\diamond), (\circ), and ($*$) represent methods B3LYP/6-31G(d), KS-ELM, KS-HC-ELM, and KS-ACO-ELM, respectively. (b) Experimental homolysis BDE and the calculated homolysis BDE values determined using the KS-HC-ELM and KS-HC-ANN methods (kcal mol^{-1}).

that when the number of clustering molecular descriptors is smaller, both the HC method and the ACO method are suitable and that when the number of the molecular descriptors is larger, the ACO method is more appropriate.

5. Conclusions

Nitric oxide (NO) is a very important signaling molecule in mammalian systems. When there are problems resulting in the release of NO molecules in the body, NO carrier molecules may be taken as a drug to treat the disease. It is complicated to measure NO molecular bond homolysis BDE in various experimental methods, and it is rather difficult to achieve chemical accuracy. The DFT method has been very popular in computational chemistry for the past two decades because many computational studies have shown that DFT calculations can capture the physical essence of the molecules. For the calculation of small molecules, the accuracy can reach a high level, but for large molecules, the cost of calculation is too expensive and the calculation accuracy is quite poor.

In this paper, the combination of KS-HC/ACO-ELM and DFT calculation methods succeeds in improving the DFT calculation accuracy of NO bond homolysis BDE. The results show that for three KS-HC/ACO-ELM methods, the STD of homolysis BDE of 92 organic molecules in the test set decreases from 4.03 to 0.30, 0.28, and 0.29 kcal mol^{-1} . This proves that the KS-HC/ACO-ELM method based on B3LYP/6-31G(d) can remarkably improve homolysis BDE calculations, and its result may be used as the reference value to experimental results with high accuracy. The adoption of Kenstone makes the model more generalized. Although the model is ambiguous and the space of the descriptors

is complex, the reproducibility of the calculation is very important. The selection of the training set and test set using the Kenstone method makes the model more robust and reproducible and makes the calculation results more convincing. The selection of appropriate molecular descriptors using the HC/ACO can eliminate the subjective bias on some input descriptors. Compared to the ANN method, the nonlinear model created by ELM can not only avoid sensitive parameters and local minimum problems, but it can also achieve more accurate calculation results with less calculation time and computing resources.

The KS-HC/ACO-ELM method expands the feasibility and applicability of the B3LYP/6-31G(d) method, and the more experimental data and molecular descriptors there are, the higher calculation accuracy the KS-HC/ACO-ELM will achieve, leading to more obvious advantages. It is important that such a high-precision method can be used to design novel NO-releasing drug molecule. We believe that it is significantly effective for the KS-HC/ACO-ELM method to improve the accuracy of DFT B3LYP/6-31G(d) calculations, even with a smaller basis set (e.g., STO-3G, etc.). Meanwhile, this method can be applied to correct other properties such as Heterolytic Bond Dissociation Energies, Absorbed Energies, Ionization Energies, and heat of formation. Further studies are ongoing. This study provides an effective tool (the combination of the KS-HC/ACO-ELM and DFT methods) to improve and predict highly accurate homolysis BDE of the molecular NO carrier systems.

Acknowledgments

The authors gratefully acknowledge the financial support from the Science and Technology Development Planning of

Jilin Province (20130522109JH, 20110364, and 20125002) and the Fundamental Research Funds for the Central Universities (11QNJJ008 and 11QNJJ027).

References

- [1] X. W. Zhao, W. Y. Zhang, X. Xu, Z. Q. Ma, and M. H. Yin, "Prediction of protein phosphorylation sites by using the composition of k-spaced amino acid pairs," *PLOS ONE*, vol. 7, no. 10, Article ID e46302, 2012.
- [2] T. Zou, M. Yin, and W. Gu, "Reverse bridge theorem under constraint partition," *Mathematical Problems in Engineering*, vol. 2010, Article ID 617398, 18 pages, 2010.
- [3] J. N. Wang, J. L. Jin, Y. Geng et al., "An accurate and efficient method to predict the electronic excitation energies of BODIPY fluorescent dyes," *Journal of Computational Chemistry*, vol. 34, no. 7, pp. 566–575, 2013.
- [4] J.-N. Wang, H.-L. Xu, S.-L. Sun et al., "An effective method for accurate prediction of the first hyperpolarizability of alkalides," *Journal of Computational Chemistry*, vol. 33, no. 2, pp. 231–236, 2012.
- [5] M. Yin, Y. Hu, F. Yang, X. Li, and W. Gu, "A novel hybrid K-harmonic means and gravitational search algorithm approach for clustering," *Expert Systems with Applications*, vol. 38, no. 8, pp. 9319–9324, 2011.
- [6] X. T. Li, J. N. Wang, and M. H. Yin, "Enhancing the performance of cuckoo search algorithm using orthogonal learning method," *Neural Computing and Applications*, 2013.
- [7] X. Zhao, X. Li, Z. Ma, and M. Yin, "Prediction of lysine ubiquitylation with ensemble classifier and feature selection," *International Journal of Molecular Sciences*, vol. 12, no. 12, pp. 8347–8361, 2011.
- [8] K. K. Irikura and D. J. Frurip, *Prediction and Estimation of Molecular Thermodynamics*, American Chemical Society, Washington, DC, USA, 1998.
- [9] C. J. Cramer, *Essentials of Computational Chemistry: Theories and Models*, Wiley, West Sussex, UK, 2002.
- [10] X.-M. Duan, G.-L. Song, Z.-H. Li, X.-J. Wang, G.-H. Chen, and K.-N. Fan, "Accurate prediction of heat of formation by combining Hartree-Fock/density functional theory calculation with linear regression correction approach," *Journal of Chemical Physics*, vol. 121, no. 15, pp. 7086–7095, 2004.
- [11] X.-M. Duan, Z.-H. Li, H.-R. Hu et al., "Linear regression correction to first principle theoretical calculations—improved descriptors and enlarged training set," *Chemical Physics Letters*, vol. 409, no. 4–6, pp. 315–321, 2005.
- [12] D. Jacquemin, V. Wathelet, and E. A. Perpète, "Ab initio investigation of the $n \rightarrow \pi^*$ transitions in thiocarbonyl dyes," *The Journal of Physical Chemistry A*, vol. 110, no. 29, pp. 9145–9152, 2006.
- [13] D. Jacquemin, E. A. Perpète, O. A. Vydrov, G. E. Scuseria, and C. Adamo, "Assessment of long-range corrected functionals performance for $n \rightarrow \pi^*$ transitions in organic dyes," *Journal of Chemical Physics*, vol. 127, no. 9, Article ID 094102, 2007.
- [14] R. Khosrokhavar, J. B. Ghasemi, and F. Shiri, "2D Quantitative Structure-property relationship study of mycotoxins by multiple linear regression and support vector machine," *International Journal of Molecular Sciences*, vol. 11, no. 9, pp. 3052–3068, 2010.
- [15] J. Homer, S. C. Generalis, and J. H. Robson, "Artificial neural networks for the prediction of liquid viscosity, density, heat of vaporization, boiling point and Pitzer's acentric factor. Part I. Hydrocarbons," *Physical Chemistry Chemical Physics*, vol. 1, no. 17, pp. 4075–4081, 1999.
- [16] L. E. S. De Souza and S. Canuto, "Efficient estimation of second virial coefficients of fused hard-sphere molecules by an artificial neural network," *Physical Chemistry Chemical Physics*, vol. 3, no. 21, pp. 4762–4768, 2001.
- [17] L. Hu, X. Wang, L. Wong, and G. Chen, "Combined first-principles calculation and neural-network correction approach for heat of formation," *Journal of Chemical Physics*, vol. 119, no. 22, pp. 11501–11507, 2003.
- [18] X. Wang, L. Hu, L. Wong, and G. Chen, "A combined first-principles calculation and neural networks correction approach for evaluating Gibbs energy of formation," *Molecular Simulation*, vol. 30, no. 1, pp. 9–15, 2004.
- [19] J. R. Votano, M. Parham, L. H. Hall, L. B. Kier, and L. M. Hall, "Prediction of aqueous solubility based on large datasets using several QSPR models utilizing topological structure representation," *Chemistry & Biodiversity*, vol. 1, no. 11, pp. 1829–1841, 2004.
- [20] X.-M. Duan, Z.-H. Li, G.-L. Song, W.-N. Wang, G.-H. Chen, and K.-N. Fan, "Neural network correction for heats of formation with a larger experimental training set and new descriptors," *Chemical Physics Letters*, vol. 410, no. 1–3, pp. 125–130, 2005.
- [21] A. Alexandridis, P. Patrinos, H. Sarimveis, and G. Tsekouras, "A two-stage evolutionary algorithm for variable selection in the development of RBF neural network models," *Chemometrics and Intelligent Laboratory Systems*, vol. 75, no. 2, pp. 149–162, 2005.
- [22] M. Arakawa, K. Hasegawa, and K. Funatsu, "QSAR study of anti-HIV HEPT analogues based on multi-objective genetic programming and counter-propagation neural network," *Chemometrics and Intelligent Laboratory Systems*, vol. 83, no. 2, pp. 91–98, 2006.
- [23] M. Fernández and J. Caballero, "Ensembles of Bayesian-regularized genetic neural networks for modeling of acetylcholinesterase inhibition by huperzines," *Chemical Biology and Drug Design*, vol. 68, no. 4, pp. 201–212, 2006.
- [24] R. K. H. Galvão, M. C. U. Araújo, M. D. N. Martins et al., "An application of subagging for the improvement of prediction accuracy of multivariate calibration models," *Chemometrics and Intelligent Laboratory Systems*, vol. 81, no. 1, pp. 60–67, 2006.
- [25] K. T. Lee, S. Bhatia, A. R. Mohamed, and K. H. Chu, "Optimizing the specific surface area of fly ash-based sorbents for flue gas desulfurization," *Chemosphere*, vol. 62, no. 1, pp. 89–96, 2006.
- [26] P. Zhou, F. Tian, and Z. Li, "A structure-based, quantitative structure-activity relationship approach for predicting HLA-A*0201-restricted cytotoxic T lymphocyte epitopes," *Chemical Biology and Drug Design*, vol. 69, no. 1, pp. 56–67, 2007.
- [27] J. M. Wu and X. Xu, "The XI method for accurate and efficient prediction of heats of formation," *The Journal of Chemical Physics*, vol. 127, no. 21, Article ID 214105, 8 pages, 2007.
- [28] M. Sattari and F. Gharagheizi, "Prediction of molecular diffusivity of pure components into air: a QSPR approach," *Chemosphere*, vol. 72, no. 9, pp. 1298–1302, 2008.
- [29] N. Qu, L. Wang, M. Zhu, Y. Dou, and Y. Ren, "Radial basis function networks combined with genetic algorithm applied to nondestructive determination of compound erythromycin ethylsuccinate powder," *Chemometrics and Intelligent Laboratory Systems*, vol. 90, no. 2, pp. 145–152, 2008.
- [30] W. Xu, X. Liu, S. Luan, Q. Zhang, and P. Zhang, "An ab initio potential energy surface of the $\text{He} + \text{H}_2 \rightarrow \text{HeH}^+ + \text{H}$ reaction," *Chemical Physics Letters*, vol. 464, no. 1–3, pp. 92–96, 2008.

- [31] J. Wu and X. Xu, "Improving the B3LYP bond energies by using the X1 method," *Journal of Chemical Physics*, vol. 129, no. 16, Article ID 164103, 11 pages, 2008.
- [32] M. D. Wodrich and C. Corminboeuf, "Reaction enthalpies using the neural-network-based X1 approach: the important choice of input descriptors," *The Journal of Physical Chemistry A*, vol. 113, no. 13, pp. 3285–3290, 2009.
- [33] T. Gao, L.-L. Shi, H.-B. Li et al., "Improving the accuracy of low level quantum chemical calculation for absorption energies: the genetic algorithm and neural network approach," *Physical Chemistry Chemical Physics*, vol. 11, no. 25, pp. 5124–5129, 2009.
- [34] H. Z. Li, W. Tao, T. Gao, H. Li, Y. H. Lu, and Z. M. Su, "Improving the accuracy of density functional theory (DFT) calculation for homolysis bond dissociation energies of Y-NO bond: generalized regression neural network based on grey relational analysis and principal component analysis," *International Journal of Molecular Sciences*, vol. 12, no. 4, pp. 2242–2261, 2011.
- [35] H. Z. Li, L. H. Hu, W. Tao et al., "a promising tool to achieve chemical accuracy for density functional theory calculations on Y-NO homolysis bond dissociation energies," *International Journal of Molecular Sciences*, vol. 13, no. 7, pp. 8051–8070, 2012.
- [36] H.-Z. Li, W. Tao, T. Gao, H. Li, Y.-H. Lü, and Z.-M. Su, "Improving the accuracy of DFT calculation for homolysis bond dissociation energies of Y-NO bond via back propagation neural network based on mean impact value," *Chemical Journal of Chinese Universities*, vol. 33, no. 2, pp. 346–352, 2012.
- [37] Y. E. Shao, "Prediction of currency volume issued in Taiwan using a hybrid artificial neural network and multiple regression approach," *Mathematical Problems in Engineering*, vol. 2013, Article ID 676742, 9 pages, 2013.
- [38] W. Ma, X. Zhang, F. Luan et al., "Support vector machine and the heuristic method to predict the solubility of hydrocarbons in electrolyte," *The Journal of Physical Chemistry A*, vol. 109, no. 15, pp. 3485–3492, 2005.
- [39] X. Zhao, Z. Ma, and M. Yin, "Using support vector machine and evolutionary profiles to predict antifreeze protein sequences," *International Journal of Molecular Sciences*, vol. 13, no. 2, pp. 2196–2207, 2012.
- [40] A. R. Lima, A. J. Cannon, and W. W. Hsieh, "Nonlinear regression in environmental sciences by support vector machines combined with evolutionary strategy," *Computers & Geosciences*, vol. 50, no. 4, pp. 136–144, 2013.
- [41] A. R. Butler and D. L. H. Williams, "The physiological role of nitric oxide," *Chemical Society Reviews*, vol. 22, no. 4, pp. 233–241, 1993.
- [42] B. A. Averill, "Dissimilatory nitrite and nitric oxide reductases," *Chemical Reviews*, vol. 96, no. 7, pp. 2951–2964, 1996.
- [43] R. M. J. Palmer, A. G. Ferrige, and S. Moncada, "Nitric oxide release accounts for the biological activity of endothelium-derived relaxing factor," *Nature*, vol. 327, no. 6122, pp. 524–526, 1987.
- [44] L. J. Ignarro, "Biosynthesis and metabolism of endothelium-derived nitric oxide," *Annual Review of Pharmacology and Toxicology*, vol. 30, pp. 535–560, 1990.
- [45] P. L. Feldman, O. W. Griffith, and D. J. Stuehr, "The surprising life of nitric oxide," *Chemical and Engineering News*, vol. 71, no. 51, pp. 26–38, 1993.
- [46] J. M. Fukuto and L. J. Ignarro, "In vivo aspects of nitric oxide (NO) chemistry: does peroxynitrite (OONO) play a major role in cytotoxicity?" *Accounts of Chemical Research*, vol. 30, no. 4, pp. 149–152, 1997.
- [47] S. Moncada, R. M. J. Palmer, and E. A. Higgs, "Nitric oxide: physiology, pathophysiology, and pharmacology," *Pharmacological Reviews*, vol. 43, no. 2, pp. 109–142, 1991.
- [48] L. J. Ignarro, "Signal transduction mechanisms involving nitric oxide," *Biochemical Pharmacology*, vol. 41, no. 4, pp. 485–490, 1991.
- [49] C. T. Gnewuch and G. Sosnovsky, "A critical appraisal of the evolution of N-nitrosoureas as anticancer drugs," *Chemical Reviews*, vol. 97, no. 3, pp. 829–1013, 1997.
- [50] C. A. Whited, J. J. Warren, K. D. Lavoie et al., "Gating NO release from nitric oxide synthase," *Journal of the American Chemical Society*, vol. 134, no. 1, pp. 27–30, 2012.
- [51] J.-P. Cheng, K. Wang, Z. Yin, X. Zhu, and Y. Lu, "NO affinity. The driving force of nitric oxide (NO) transfer in biomimetic N-nitrosoacetanilide and N-nitrososulfoanilide systems," *Tetrahedron Letters*, vol. 39, no. 43, pp. 7925–7928, 1998.
- [52] J.-P. Cheng, M. Xian, K. Wang, X. Zhu, Z. Yin, and P. G. W. Peng George Wang, "Heterolytic and homolytic Y-NO bond energy scales of nitroso-containing compounds: chemical origin of NO release and NO capture," *Journal of the American Chemical Society*, vol. 120, no. 39, pp. 10266–10267, 1998.
- [53] M. Xian, X.-Q. Zhu, J. Lu, Z. Wen, and J.-P. Cheng, "The first O-NO bond energy scale in solution: heterolytic and homolytic cleavage enthalpies of O-nitrosyl carboxylate compounds," *Organic Letters*, vol. 2, no. 3, pp. 265–268, 2000.
- [54] X.-Q. Zhu, J.-Q. He, Q. Li, M. Xian, J. Lu, and J.-P. Cheng, "N-NO bond dissociation energies of N-nitroso diphenylamine derivatives (or analogues) and their radical anions: implications for the effect of reductive electron transfer on N-NO bond activation and for the mechanisms of NO transfer to nitraniions," *Journal of Organic Chemistry*, vol. 65, no. 20, pp. 6729–6735, 2000.
- [55] J.-M. Lü, J. M. Wittbrodt, K. Wang et al., "NO affinities of S-nitrosothiols: a direct experimental and computational investigation of RS-NO bond dissociation energies," *Journal of the American Chemical Society*, vol. 123, no. 12, pp. 2903–2904, 2001.
- [56] X.-Q. Zhu, W.-F. Hao, H. Tang, C.-H. Wang, and J.-P. Cheng, "Determination of N-NO bond dissociation energies of N-Methyl-N-nitrosobenzenesulfonamides in acetonitrile and application in the mechanism analyses on NO transfer," *Journal of the American Chemical Society*, vol. 127, no. 8, pp. 2696–2708, 2005.
- [57] X.-Q. Zhu, J.-Y. Zhang, and J.-P. Cheng, "Mechanism and driving force of NO transfer from S-nitrosothiol to cobalt(II) porphyrin: a detailed thermodynamic and kinetic study," *Inorganic Chemistry*, vol. 46, no. 2, pp. 592–600, 2007.
- [58] X. Li, X.-Q. Zhu, X.-X. Wang, and J.-P. Cheng, "Determination of N-NO bond dissociation energies of N-nitrosoindoles and their radical anions in acetonitrile," *Chemical Journal of Chinese Universities*, vol. 28, no. 12, pp. 2295–2298, 2007.
- [59] X. Li, X.-Q. Zhu, and J.-P. Cheng, "Determination of NO chemical affinities of benzyl nitrite in acetonitrile," *Chemical Journal of Chinese Universities*, vol. 28, no. 12, pp. 2327–2329, 2007.
- [60] X. Li and J.-P. Cheng, "Determination of S-NO bond dissociation energies of S-Nitroso-N-acetyl-D, L-penicillamine dipeptides," *Chemical Journal of Chinese Universities*, vol. 29, no. 8, pp. 1569–1572, 2008.
- [61] X. Li, H. Deng, X.-Q. Zhu, X. Wang, H. Liang, and J.-P. Cheng, "Establishment of the C-NO bond dissociation energy scale in solution and its application in analyzing the trend of NO transfer from C-nitroso compound to thiols," *Journal of Organic Chemistry*, vol. 74, no. 12, pp. 4472–4478, 2009.

- [62] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.
- [63] R. W. Kennard and L. A. Stone, "Computer aided design of experiments," *Technometrics*, vol. 11, no. 1, pp. 137–148, 1969.

Research Article

Infrared Target Detection and Location for Visual Surveillance Using Fusion Scheme of Visible and Infrared Images

Zi-Jun Feng,^{1,2} Xiao-Ling Zhang,¹ Li-Yong Yuan,³ and Jia-Nan Wang²

¹ School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

² School of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, China

³ College of Mathematics, Physics and Information Engineering, Zhejiang Normal University, Jinhua 321000, China

Correspondence should be addressed to Xiao-Ling Zhang; cs@nenu.edu.cn

Received 21 May 2013; Accepted 16 July 2013

Academic Editor: William Guo

Copyright © 2013 Zi-Jun Feng 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 main goal of image fusion is to combine substantial information from different images of the same scene into a single image that is suitable for human and machine perception or for further image-processing tasks. In this study, a simple and efficient image fusion approach based on the application of the histogram of infrared images is proposed. A fusion scheme to select adaptively weighted coefficients for preserving salient infrared targets from the infrared image and for obtaining most spatial detailed information from the visible image is presented. Moving and static infrared targets in the fused image are labeled with different colors. This technique enhances perception of the image for the human visual system. In view of the modalities of infrared images, low resolution, and low signal-to-noise ratio, an anisotropic diffusion equation model is adopted to remove noise and to effectively preserve edge information before the fusion stage. By using the proposed method, relevant spatial information is preserved and infrared targets are clearly identified in the resulting fused images.

1. Introduction

With the rapid improvement of sensor technology, many surveillance systems have been developed in recent years. Infrared sensors, which are useful tools because of their individual modalities, have been employed in fields that include military surveillance, medical imaging, and machine vision. Infrared sensors can detect relative differences in the amount of thermal energy reflected from objects in the scene. They are thus more effective than visible cameras under poor lighting conditions. Current studies have focused on object detection and tracking from infrared images, and infrared sensor-based methods have shown good performance [1–4]. However, sensors often differ in modalities, considering physical and technological limitations. The image data acquired using different sensors exhibit diverse modalities, such as degradation and thermal and visual characteristics. By combining two or more different sensor data sets, a surveillance system can perform better than a system that uses only a single sensor. This technology is called image fusion. Image fusion is defined as the process of combining substantial information from

several sensors using mathematical techniques to create a single composite image that is highly comprehensive and thus extremely useful for a human operator or in the execution of other computer vision tasks [5].

In this study, we focus on the fusion process of visible and infrared imagery. Visible imagery has high resolution and can provide spatial details of the scene. Infrared imagery aids in the detection and recognition of heat-based targets under poor lighting conditions or in cases in which the target and the background are of the same color [6, 7]. Relevant information in both visible and infrared images should be preserved in the resulting fused images. A surveillance system can benefit from an efficient fusion to improve targeting and identification performance. In recent years, image fusion has become an important topic. Many fusion methods for visible and infrared images have been proposed [4, 6, 8, 9]. However, these techniques often lead to spatial distortions in the visible image because of the mixing of irrelevant infrared information in the nontarget area of the infrared image. In [8], the experiments on visible and infrared images show the contamination of the background regions by the

infrared information. Considering the modalities of visible and infrared images, the following three basic requirements must be met to achieve a good fusion method: (1) infrared targets must be perfectly preserved in the fused image; (2) spatial detail information from the visible image must not contaminate the nontarget regions of the infrared image; and (3) the fused image should be enhanced to be easily understood by the human visual system.

In the current work, a novel fusion algorithm is proposed to meet these requirements. The method is aimed at preserving high spatial detail information and showing infrared targets clearly in the resulting fused image. An adaptive threshold can be determined by the histogram of the infrared image. If the gray value (pixel by pixel) of the infrared image is greater than or equal to the threshold value, then the pixel value of the fused image is calculated by a weight determination function. Otherwise, the new value of the fused image would be directly obtained from the visible image. The proposed fusion algorithm is highly suitable for the image fusion of visible and infrared images because it displays salient infrared targets and avoids spatial distortion. Furthermore, the idea of labeling moving targets with red contours and marking static targets (or the highlighted nontarget areas of the infrared image) with green contours is very effective for the human visual system.

The rest of the paper is organized as follows. Section 2 gives a brief introduction of nonlinear diffusion filtering. Section 3 describes in detail the image fusion scheme. Section 4 provides the experimental results and comparisons. Section 5 discusses infrared target detection. Section 6 presents the conclusions.

2. Image Denoising

Given the physical limitations of sensors, many infrared imaging systems produce images with low signal-to-noise ratio, contrast, and resolution; these features reduce the detectability of targets and impede further investigation of infrared images [1, 10, 11]. Thus, image denoising has become an important issue in addressing the shortcomings of infrared images. Several methods, such as Gaussian filtering [12] and wavelet-based denoising, have been developed to address these issues [10, 13]. A disadvantage of some common methods is that they do not only smooth noise, but they also blur edges that contain significant image features. In the current study, an anisotropic diffusion [14–16] algorithm is adopted in infrared image denoising to filter noise and effectively preserve edge information.

After more than two decades of research, partial differential equation- (PDE-) based nonlinear diffusion processes have been widely applied in signal and image processing. PDE-based and improved PDE approaches have been successfully applied in image denoising, segmentation, enhancement, and restoration [15–18]. The first PDE-based nonlinear anisotropic diffusion technique was reported by Perona and Malik [14]. Different from the isotropic heat conduction equation, the anisotropic diffusion equation filters an image depending on its local properties. The anisotropic diffusion

equation removes image noise and simultaneously preserves its edges.

The diffusion equation in two dimensions can be expressed as

$$\frac{\partial I(x, y, t)}{\partial t} = \operatorname{div}(g(|\nabla I|) \nabla I), \quad (1)$$

$$I(x, y, 0) = I_0(x, y),$$

where div is the divergence operator, ∇ is a gradient operator, and $g(|\nabla I|)$ denotes the diffusion conductance function chosen as a decreasing function to ensure high diffusion in a homogeneous region and weak diffusion near the edges. A typical choice for the conductance function is

$$g(|\nabla I|) = \frac{1}{1 + (|\nabla I|/K)^2}, \quad (2)$$

where K is a gradient threshold that can be fixed by an empirical constant.

3. Image Fusion Scheme

First, a common weighted combination rule is presented

$$I_f(x, y) = \begin{cases} \omega_2 I_v(x, y) + \omega_1 I_i(x, y) & \text{if } I_i(x, y) \geq \text{th}, \\ I_v(x, y) & \text{otherwise,} \end{cases} \quad (3)$$

where $I_v(x, y)$ is the intensity value of the pixel coordinate (x, y) in the visible image; $I_i(x, y)$ and $I_f(x, y)$ are the intensity values of the corresponding pixel (x, y) in the infrared image and the fused image, respectively; and th is used to distinguish between the dark background and the bright objects determined by image histogram-based method. The fused image is produced pixel by pixel according to the gray value of the infrared image. If the gray value is greater than or equal to th , then the new gray value of the fused image is obtained at the corresponding pixel location using (3). Otherwise, the gray value of the fused image is directly obtained from the visible image. The next step of the fusion algorithm is the production of the weighted coefficients.

Second, a weight determination function is defined as

$$\omega_1 = \frac{1}{1 + \exp(-k((x - sp)/(G_{\max} - sp) - \sigma))}, \quad (4)$$

$$\omega_2 = 1 - \omega_1,$$

where x represents the intensity value of the infrared image, G_{\max} is equal to 255, k controls the slope of the function, σ determines the translation of the curve, and sp is the starting position of the curve. Figure 1 depicts the function defined in (4), considering the conditions wherein $k = 10$, $\sigma = 0.5$, and $sp = 120$. An increase in k results in a steep curve. Therefore, k is usually limited to the range in [8, 12], depending on the intensity levels of each image. The quantity σ is a constant equal to 0.5; this constant specifies that the values of ω_1 and ω_2 must be 0.5 when x is the middle point in the range $[sp, G_{\max}]$.

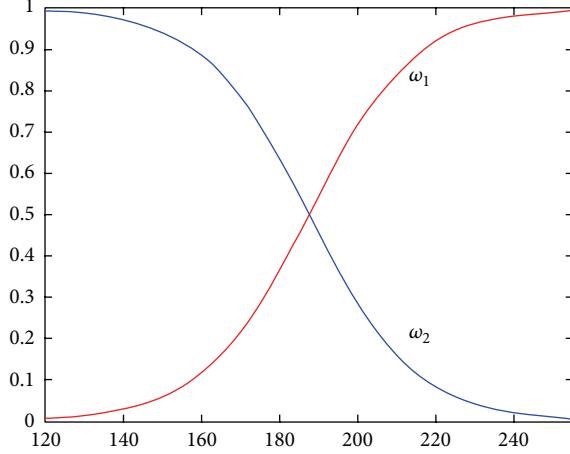


FIGURE 1: Example of a weight determination function.

In this study, the parameters $k = 10$ and $\sigma = 0.5$ are chosen, but the sp value is an adaptive selection based on the intensity histogram of the infrared image.

Third, the sp value is defined as

$$sp = G_{\max} - 2(G_{\max} - th), \quad (5)$$

where G_{\max} is equal to 255 and th is the same threshold described in (3) that distinguishes between the targets and the background according to the histogram of the infrared image. The value of th is in the range [40, 80], as determined by considering the intensity level distribution of the infrared image in the experiments.

4. Experimental Results

4.1. Image Data Sets. Two sets of visible and infrared images were chosen. They are publicly available through the ImageFusion.org website [19]. These image sets were chosen because (1) the grayscale levels of the visible image are abundant and the light objects of the scene are clearly distinguishable in the infrared image (Figures 2(a) and 2(b)); and (2) the infrared image is corrupted by noise and the intensities of targets are low (Figure 3(b)). Figures 3(a) and 3(b), respectively, show a visible frame and the corresponding infrared frame from the sequences extracted from MPEG files. The image in Figure 3(b) was warped using planar homography to align it with the visible spectrum image [20]. All input images were assumed to be registered, and each pair of images contains exactly the same scene.

The first image set (UN Camp, frame 1815) comprises a terrain scene characterized by a path, trees, fences, and a house roof (Figure 2(a)). A person standing behind the trees and closing the fence is shown in the infrared image (thermal 3–5 μm , Figure 2(b)). Although the two images exhibit the same scene, the visible image clearly provides spatial details of the scene, but the human figure is invisible.

On the contrary, the human figure is highlighted in the infrared image, but other objects are hard to recognize correctly. The second data set (AIC) was chosen from the AIC thermal and visible nighttime data set sequence [19, 20]. Frame 3853 (Figure 3(a)) was extracted from the visible MPEG format file, and the corresponding frame (Figure 3(b)) was obtained from the infrared (thermal: 7 μm to 14 μm) sequence. To ensure consistency in the procedure, the color image was converted to a grayscale image. The white margin of the infrared image was manually filled with black color to achieve good image fusion. The scene contains buildings, bright windows, roads, and pedestrians. Further details and descriptions of the data acquisition procedure can be found in [20].

4.2. Infrared Image Denoising. The method for reducing noise in this case is nonlinear diffusion filtering using techniques similar to those discussed in Section 2. An additive operator splitting (AOS) algorithm [21], a type of efficient nonlinear diffusion filtering, was applied to the infrared images. The AOS algorithm performs anisotropic diffusion to remove image noise while preserving its edges.

4.3. Fusion Results and Comparisons. To illustrate the performance of the proposed image fusion approach, image fusion was performed using conventional methods, namely, weighted averaging (WAV-based), discrete wavelet transform (DWT-based), and nonsubsampled contourlet transform (NSCT-based). The DWT-based and NSCT-based methods are performed by simply merging the low-pass and high-pass subband coefficients using the averaging scheme and the choose-max selection scheme, respectively. The DWT-based fusion algorithm is performed using five-level decomposition. The NSCT-based method is performed by using db3 wavelets in scale decomposition; 9–7 wavelets are used in a nonsubsampled directional filter bank, in which the number of directions is 4.

Figures 2 and 3 illustrate the fusion results using the aforementioned methods. The proposed scheme captures most of the salient areas of the infrared images and preserves the spatial detail information from the visible images. The infrared targets are obvious, and the nontarget regions are seldom contaminated in the fused images. For clear comparisons, the difference images between the fused images and the source visible images are given in Figure 4. The fused images obtained using the proposed method have the best visual quality.

Objective evaluation criteria, namely, fusion root-mean-square error (FRMSE) and correlation coefficient (CC), were used to evaluate the fused images. Considering that the size of the target in the infrared image is small compared with that in the scene, we suppose that I_v is the ideal reference image. $I_v(x, y)$ and $I_f(x, y)$ denote the pixel value of the visible image and the fused image at points (x, y) , respectively. The size of the images is calculated as $M \times N$.

- (1) FRMSE can effectively reflect the similarity between two images. Small FRMSE values provide satisfactory

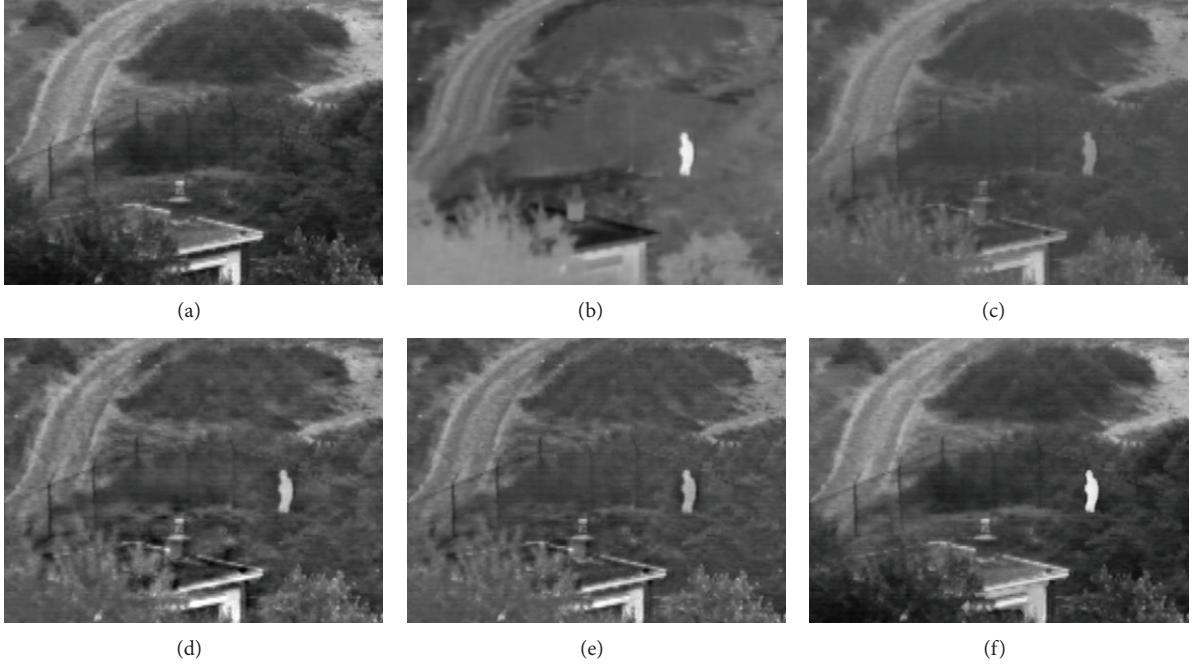


FIGURE 2: Source images and fusion results of different fusion algorithms from the UN Camp sequence set. (a) Visible image; (b) denoised infrared image; and (c)–(f) images obtained by the WAV-based, DWT-based, NSCT-based, and proposed methods, respectively.

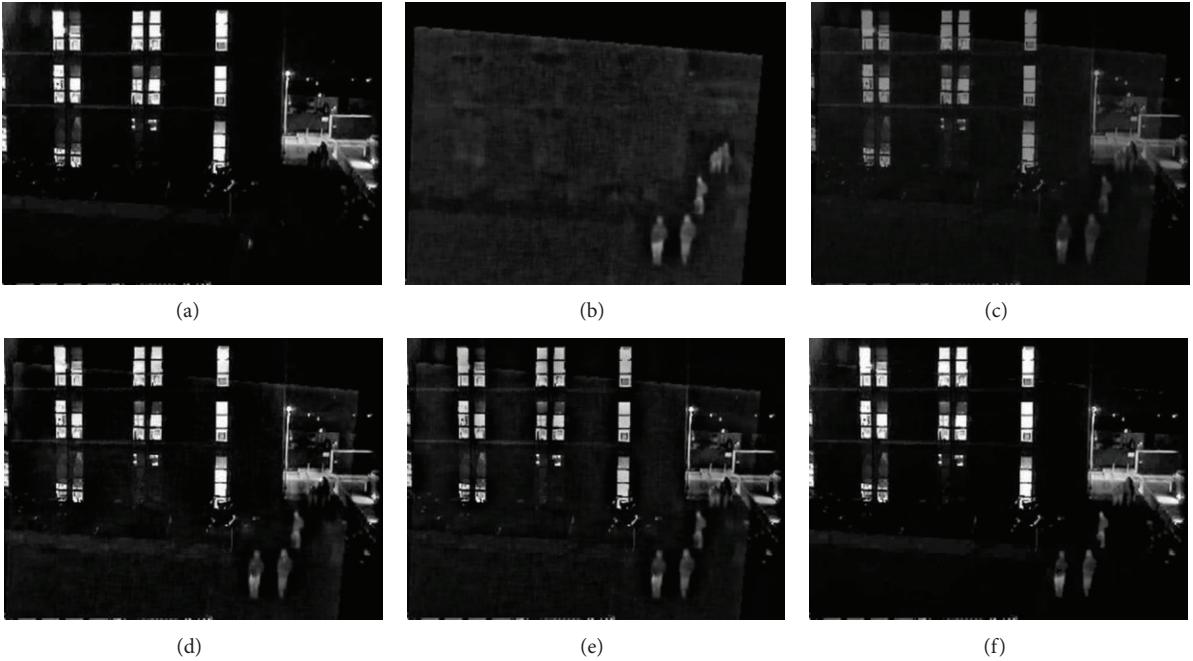


FIGURE 3: Source images and fusion results of different fusion algorithms from the AIC sequence set. (a) Visible image; (b) denoised infrared image; and (c)–(f) images obtained by the WAV-based, DWT-based, NSCT-based, and proposed methods, respectively.

fusion results:

(2) CC can be evaluated to compare $I_v(x, y)$ with $I_f(x, y)$:

$$\text{FRMSE} = \sqrt{\frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N (I_v(x, y) - I_f(x, y))^2}. \quad (6)$$

$$\text{CC} = \left(\sum_{x=1}^M \sum_{y=1}^N [(I_v(x, y) - \bar{I}_v) \times (I_f(x, y) - \bar{I}_f)] \right)$$

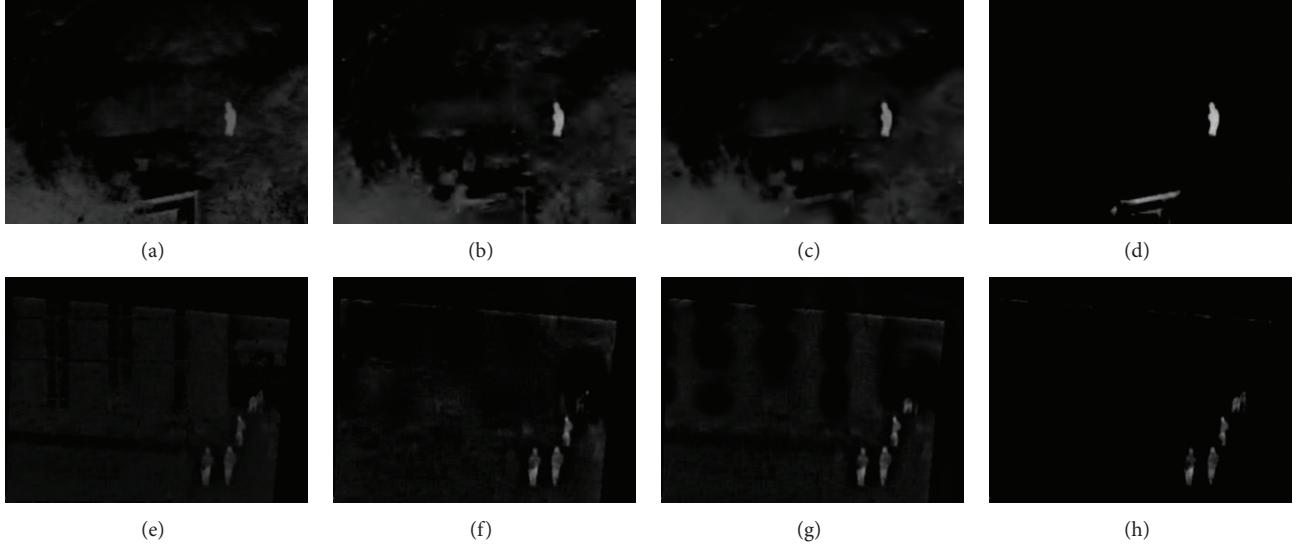


FIGURE 4: Difference images between the fused images and the corresponding visible images. (a)–(d) For the UN Camp sequence, fusion was performed using the WAV-based, DWT-based, NSCT-based, and proposed methods, respectively. (e)–(h) For the AIC sequence, fusion was performed using the WAV-based, DWT-based, NSCT-based, and proposed methods, respectively.

$$\times \left(\sum_{x=1}^M \sum_{y=1}^N \left[(I_v(x, y) - \bar{I}_v)^2 \right] \right)^{-1/2}, \quad (7)$$

where \bar{I}_v and \bar{I}_f are the means of the visible image and the fused image, respectively. The maximum coefficient corresponds to the optimum fusion.

Table 1 shows the results of the quantitative evaluation using the two evaluation methods. The proposed method achieves superior results. The values of the fusion results of the first image set are not entirely consistent with those of the second image set because the visible image from the UN Camp sequence contains abundant texture information. By contrast, the AIC visible sequence contains structural details of buildings, such as edges. These objective criteria prove that the images fused using the proposed method are strongly correlated with the corresponding visible images; that is, the proposed scheme ensures that useful spatial detailed information of the visible images is preserved in the fused images.

5. Infrared Target Detection

Automatic object detection remains difficult to be undertaken. An object detector needs to cope with the diversity of visual imagery that exists in the world at large. Different detection methods can be used for different environmental conditions, so numerous approaches for automatic object detection have been investigated [22–26]. A common method is to extract targets from the image sequence through background subtraction when the video is captured by a stationary

TABLE 1: Comparison of fusion results.

Method	UN Camp set		AIC set	
	FRMSE	CC	FRMSE	CC
WAV-based	26.0935	0.7913	26.1056	0.9512
DWT-based	26.9110	0.7394	14.1720	0.9605
NSCT-based	25.4268	0.7813	18.4112	0.9439
Proposed	13.0508	0.9443	7.3073	0.9892

camera. The simplest way is to calculate an average image of the scene or to smoothen the pixels of the background with a Kalman filter [23] under the assumption that the background consists of stationary objects. A preferable way to tolerate background variation in a video is to employ a Gaussian function that describes the distribution of each pixel belonging to a stable background object. Among these background subtraction techniques, the mixture of Gaussians (MoG) has been widely utilized to model scene backgrounds at the pixel level [24–26].

In MoG, the distribution of recently observed values of each pixel in the scene is characterized. A new pixel value is represented by one of the major components of the mixture model and is used to update the model. One of the significant advantages of this method is that when something is allowed to become part of the background the existing model of the background is not destroyed. The original background color remains in the mixture until it becomes the most probable distribution and a new color is observed [26]. Good results of foreground object detection by applying MoG to outdoor scenes have been reported.

In surveillance applications, moving and static targets need to be correctly detected (for military surveillance systems, static targets must not be arbitrarily ignored), and the locations of the targets need to be identified. In Section 4, the

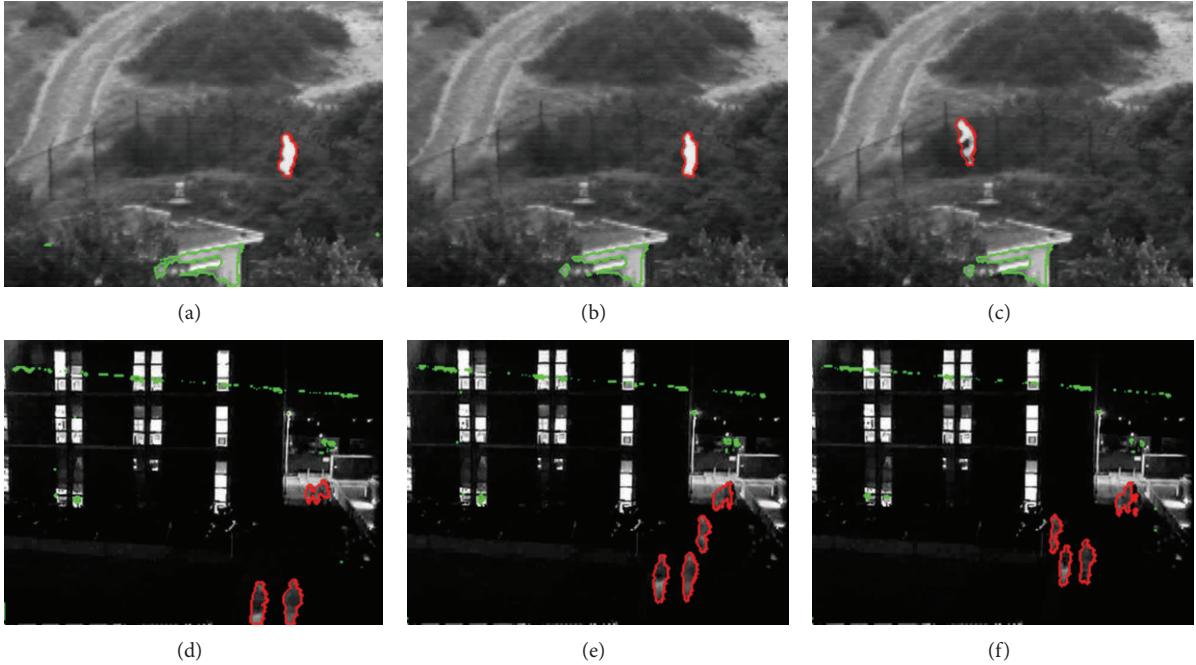


FIGURE 5: Moving and static objects were detected and labeled with different colors. (a)–(c) UN Camp sequence: frames 1813, 1815, and 1830 are shown. (d)–(f) AIC sequence: frames 3796, 3853, and 3890 are shown.

resulting fused image is obtained according to the proposed fusion rule of visible and infrared images. However, the fused image in the implementation was not computed to determine whether a moving object is present in the scene.

To separate moving targets from the static targets, infrared images easily address the issues of background modeling based on the MoG method. The chromatic contours of infrared targets in the fused images are highly suitable for the human visual system. Visual effect is enhanced in the natural scene when moving, and static targets are labeled with different colors. The data sets are still UN Camp and AIC images, as described in Section 4. In Figure 5, the moving and static targets are marked in red and green, respectively. These colors help humans clearly distinguish targets from the background in the fused images. All AIC infrared frames were warped to align them with the corresponding visible spectrum images. Green-dotted borders can be observed on top of the frames of the AIC sequence (Figure 5).

6. Conclusions

An efficient fusion method for the image fusion of infrared and visible images is proposed. To obtain good results using the proposed fusion scheme, a weight determination function by a suitable coefficient that enhances infrared targets and preserves spatial detail information from the infrared and visible images, respectively, is described. The method is appropriate for the image fusion of visible and infrared images. The infrared targets in the natural scene can be clearly distinguished in the resulting fused images. The infrared targets are highlighted by marking them in red (or green) in the fused image. This technique is useful for visual

surveillance. Moving targets are also detected and marked in red in this study. Future work will be focused on determining ways to track certain moving targets effectively.

Acknowledgments

The authors would like to acknowledge TNO Human Factors in The Netherlands and Dr. C. Ó Conaire for providing source images and video sequences. This work is supported by the Science Foundation for Research Fund for the Doctoral Program of Higher Education of China (no. 20100043120012), National Natural Science Foundation of China for Young Scholars (no. 41101434), and Opening Fund of Top Key Discipline of Computer Software and Theory in Zhejiang Provincial Colleges at Zhejiang Normal University under Grant no. ZSDZZZZXK37.

References

- [1] J. W. Davis and V. Sharma, "Background-subtraction using contour-based fusion of thermal and visible imagery," *Computer Vision and Image Understanding*, vol. 106, no. 2-3, pp. 162–182, 2007.
- [2] Y. Li, X. Mao, D. Feng, and Y. Zhang, "Fast and accuracy extraction of infrared target based on Markov random field," *Signal Processing*, vol. 91, no. 5, pp. 1216–1223, 2011.
- [3] J. Shaik and K. M. Iftekharuddin, "Detection and tracking of targets in infrared images using Bayesian techniques," *Optics and Laser Technology*, vol. 41, no. 6, pp. 832–842, 2009.
- [4] S. Colantonio, M. Benvenuti, M. G. di Bono, G. Pieri, and O. Salvetti, "Object tracking in a stereo and infrared vision system," *Infrared Physics and Technology*, vol. 49, no. 3, pp. 266–271, 2007.

- [5] T. Stathaki, *Image Fusion: Algorithms and Applications*, Academic Press, 2008.
- [6] A. Toet, M. A. Hogervorst, S. G. Nikolov et al., "Towards cognitive image fusion," *Information Fusion*, vol. 11, no. 2, pp. 95–113, 2010.
- [7] M. Leviner and M. Maltz, "A new multi-spectral feature level image fusion method for human interpretation," *Infrared Physics and Technology*, vol. 52, no. 2-3, pp. 79–88, 2009.
- [8] J. J. Lewis, R. J. O' Callaghan, S. G. Nikolov, D. R. Bull, and N. Canagarajah, "Pixel- and region-based image fusion with complex wavelets," *Information Fusion*, vol. 8, no. 2, pp. 119–130, 2007.
- [9] P. G. Wang, H. Tian, and W. Zheng, "A novel image fusion method based on FRFT-NSCT," *Mathematical Problems in Engineering*, vol. 2013, Article ID 408232, 9 pages, 2013.
- [10] C. Ni, Q. Li, and L. Z. Xia, "A novel method of infrared image denoising and edge enhancement," *IEEE Transactions on Signal Processing*, vol. 88, no. 6, pp. 1606–1614, 2008.
- [11] V. S. Petrović and C. S. Xydeas, "Sensor noise effects on signal-level image fusion performance," *Information Fusion*, vol. 4, no. 3, pp. 167–183, 2003.
- [12] A. P. Witkin, "Scale-space filtering: a new approach to multi-scale description," in *Proceedings of the IEEE International conference on Acoustic, Speech, and Signal Processing (ICASSP '84)*, vol. 9, pp. 150–153, March 1984.
- [13] S. Mallat and W. L. Hwang, "Singularity detection and processing with wavelets," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 617–643, 1992.
- [14] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.
- [15] J. Montiel and A. Beghdadi, "A new interpretation and improvement of the nonlinear anisotropic diffusion for image enhancement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 940–946, 1999.
- [16] H. Singh, V. Kumar, and S. Bhooshan, "Anisotropic diffusion for details enhancement in multiexposure image fusion," *ISRN Signal Processing*, vol. 2013, Article ID 928971, 18 pages, 2013.
- [17] X. Li and M. Yin, "An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure," *Advances in Engineering Software*, vol. 55, pp. 10–31, 2013.
- [18] Y. Hu, M. Yin, and X. Li, "A novel objective function for job-shop scheduling problem with fuzzy processing time and fuzzy due date using differential evolution algorithm," *International Journal of Advanced Manufacturing Technology*, vol. 56, no. 9–12, pp. 1125–1138, 2011.
- [19] ImageFusion.Org, The Online Resource for Research in Image Fusion, <http://www.imagefusion.org/>.
- [20] C. Ó. Conaire, N. E. O'Connor, E. Cooke, and A. F. Smeaton, "Comparison of fusion methods for thermo-visual surveillance tracking," in *Proceedings of the 9th International Conference on Information Fusion (FUSION '06)*, pp. 1–7, July 2006.
- [21] J. Weickert, B. M. Romeny, and M. A. Viergever, "Efficient and reliable schemes for nonlinear diffusion filtering," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 398–410, 1998.
- [22] L. Li, W. Huang, I. Y. H. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," in *Proceedings of the 11th ACM International Conference on Multimedia (MM '03)*, pp. 2–10, November 2003.
- [23] D. Koller, J. Weber, T. Huang et al., "Towards robust automatic traffic scene analysis in real-time," in *Proceedings of the 12th IAPR International conference on Pattern Recognition*, pp. 126–131, October 1994.
- [24] M. Harville, "A framework for high-level feedback to adaptive, per-pixel, mixture-of-Gaussian background models," in *Proceedings of the 7th European conference on Computer Vision (ECCV '02)*, pp. 543–560, May 2002.
- [25] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proceedings of the 6th European conference on Computer Vision (ECCV '00)*, pp. 751–767, June 2000.
- [26] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

Research Article

Phase Transitions of Contingent Planning Problem

Junping Zhou, Weihua Su, Zhiqiang Ma, and Minghao Yin

College of Computer, Northeast Normal University, Changchun 130117, China

Correspondence should be addressed to Zhiqiang Ma; mazq@nenu.edu.cn and Minghao Yin; ymh@nenu.edu.cn

Received 19 June 2013; Accepted 18 July 2013

Academic Editor: William Guo

Copyright © 2013 Junping Zhou 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 explores the phase transitions of the contingent planning problems. We present CONTINGENT PLAN-EXISTENCE algorithm and CONTINGENT PLAN-NONEEXISTENCE algorithm for quickly proving that the contingent planning instances have solutions and have no solutions, respectively. By analyzing the two algorithms, the phase transition area of the contingent planning problems is obtained. If the number of the actions is not greater than θ_{ub} , the CONTINGENT PLAN-NONEEXISTENCE algorithm can prove that nearly all the contingent planning instances have no solution. If the number of the actions is not lower than θ_{lb} , the CONTINGENT PLAN-EXISTENCE algorithm can prove that nearly all the contingent planning instances have solutions. The results of the experiments show that there exist phase transitions from a region where almost all the contingent planning instances have no solution to a region where almost all the contingent planning instances have solutions.

1. Introduction

There are many artificial intelligence (AI) problems that have been shown to be inherently intractable. For example, the propositional satisfiability problem, or k -SAT problem for short, is a prototypical NP-complete problem [1], which means that k -SAT problem cannot be solved in polynomial time if $P \neq NP$. It is significant to investigate the phase transitions of the intractable problems. That is because by working on it the researchers cannot only analyze the structure of intractable problems, but also understand the average-case performance of the solvers that solve these AI problems.

In recent years, much attention has been given to the phase transitions of the random k -SAT problems [2–8]. The researchers found that the phase transitions of the random k -SAT problems did exist, where instances changed from being almost all satisfiable to being almost all unsatisfiable, and the hard instances only occurred in the phase transition area. The same phenomena were also observed for some other NP-complete problems, such as the traveling salesman problems [9], the constraint satisfaction problems [10], and the Max-SAT problems [11]. Some other researchers made tremendous progress towards the phase transitions of the artificial planning problems, for instance, STRIPS planning problems [12], conformant planning problems [13], and plan modification of the conformant planning problems [14]. In consequence,

one interesting question was put forward whether there were phase transitions in other artificial planning problems.

The aim of this paper is to explore the phase transitions of the contingent planning problems. In order to investigate the phenomena, CONTINGENT PLAN-EXISTENCE algorithm and CONTINGENT PLAN-NONEEXISTENCE algorithm are presented. The phase transition point of the contingent planning problems is obtained by providing probabilistic analyses of the CONTINGENT PLAN-EXISTENCE algorithm and the CONTINGENT PLAN-NONEEXISTENCE algorithm. The results of the experiments show that the empirical results match the phase transitions from the theoretic analyses.

2. Background

2.1. Definitions and Notations. This section describes some notations about contingent planning problem relevant to our work. Further details, such as the exclusive actions and the consistent actions, can be found in [15–19].

Definition 1 (contingent planning problem). A contingent planning problem is a 5-tuple $P = \langle S, A, O, I, G \rangle$ with the following components:

- (i) S is a finite set of states and the subset of S is called belief state;

- (ii) A is a finite set of actions. An action is a pair $\langle \text{pre}, \text{eff} \rangle$, respectively, called preconditions and effects. Effects are a triple $\langle \text{cond}, \text{del}, \text{add} \rangle$, where cond is the effect conditions, del is the deleting effects and add is the adding effects;
- (iii) O is a finite set of observations;
- (iv) $I \subseteq S$ is a set of states over S , which is called the initial belief state;
- (v) $G \neq \emptyset$ is the goal, which is actually a partial state composed of the goal conditions.

More specifically, contingent planning problem is the task of generating a conditional plan given uncertainty about the initial state and action effects, but with the ability to observe some aspects of the current state [20]. In this paper we only focus on the contingent planning problem with uncertainty about the initial state and determinacy about the actions effects, but with the ability to sense the current state.

Definition 2 (action applicability). Let Bs be a belief state. The action a is applicable in Bs if $\forall s \in Bs : \text{pre}(a) \subseteq s$.

If the preconditions of an action are satisfied by all states of the belief state, then the action can be applicable. On the contrary, if the preconditions of an action are not satisfied, the action cannot be carried out, which means that the belief state is changeless. The resulting belief state by executing an action is determined by the adding effects and the deleting effects.

Definition 3 (observation function). Let O be a finite set of observations and Bs an arbitrary belief state. The observation function over Bs and O is a function $\varphi : Bs \rightarrow 2^O \setminus \emptyset$ such that $\forall s \in Bs : \varphi(s) \subseteq O$.

In the following, we will use $\varphi^-(o)$ to denote the set of states that are compatible with the observation o ; $\varphi^-(o^-)$ to denote the set of states that are compatible with other observations rather than o .

Definition 4 (conditional plan). The set of conditional plans Π is the minimal set such that

- (i) $\emptyset \in \Pi$;
- (ii) for an action a , if $a \in A$ and $\pi \in \Pi$, then $a \circ \pi \in \Pi$;
- (iii) if $o \in O$ and $\pi_1, \pi_2 \in \Pi$, then if o then π_1 else $\pi_2 \in \Pi$.

Definition 5 (contingent plan). Given a contingent planning problem $P = \langle S, A, O, I, G \rangle$, a contingent plan or solution is a conditional plan π to the problem P if and only if

- (i) π is applicable in the initial belief state I ;
- (ii) every run of π from the initial belief state I ends in G .

A random contingent planning model, called variable model, is used to carry out an investigation on the phase transitions, which we will address in the following.

Definition 6 (variable model). A variable model M_v is a 5-tuple $\langle S, A, O, I, G \rangle$, where

- (i) S is a finite set of states and every state is made up of some propositions;
- (ii) A is a finite set of actions. For $\forall a_1, a_2 \in A$, $\text{Num}(\text{pre}(a_1)) = \text{Num}(\text{pre}(a_2))$ and $\text{Num}(\text{eff}(a_1)) = \text{Num}(\text{eff}(a_2))$, where $\text{Num}(\text{pre}(a_1))$ and $\text{Num}(\text{pre}(a_2))$ are the numbers of preconditions of a_1 and a_2 respectively, $\text{Num}(\text{eff}(a_1))$ and $\text{Num}(\text{eff}(a_2))$ are the numbers of postconditions of a_1 and a_2 , respectively;
- (iii) O is a finite set of observations;
- (iv) $I \subseteq S$ is a set of initial states over S , which is also called the initial belief state;
- (v) $G \neq \emptyset$ is the goal, which is actually a partial state composed of the goal conditions.

2.2. The Fundamental Analysis of the Variable Model. In this section, we present the distributions of instances under the variable model. Before addressing the analysis of the variable model, two assumptions of the random contingent planning instances generated in this paper are provided. The first one is that each precondition of an action is selected independently of other precondition and postcondition. Similar rule is hold for the postconditions. The second one is that each action has a fixed number of preconditions. And the number of postconditions also agrees with the rule.

Because of the assumptions that each precondition (or postcondition) of an action is selected independently of other preconditions and postconditions, and the precondition or the postcondition is either a proposition or its negation, the random contingent planning instances under the variable model are distributed as follows.

Given n propositions, m actions, r preconditions, c postconditions, and g goal conditions, for a random proposition p and an action a , p is a precondition of the action with the probability $r/(2n)$; alternatively $\neg p$ is a precondition with the probability $r/(2n)$. And $c/(2n)$ is the probability for postconditions.

Because the initial state of the contingent planning problem is partially known, there exists a set of initial states, where the number of the initial states is k in this paper. Every initial state is made up of conditions, each of which is selected at random from the $2n$ conditions since there are n propositions. For the goal conditions, the g goal conditions are also selected at random from the $2n$ conditions.

Furthermore, in the rest of paper, three inequalities are also used to analyze the phase transitions. The detailed proofs can be found in [21]

$$e^{-x/(1-x)} \leq 1 - x, \quad x \in [0, 1] \quad (1)$$

$$1 - x \leq e^{-x} \quad (2)$$

$$\frac{xy}{1+xy} \leq 1 - (1-x)^y, \quad x \in [0, 1]. \quad (3)$$

```

Algorithm CONTINGENT PLAN-NONEXISTENCE ( $S, A, O, I, G$ )
  for each  $s \in I$ 
    if  $pre(A) \notin s$ ;
    then return failure;
  endfor
  for each  $g \in G$ 
    if  $g \notin eff(A)$ 
    then return failure;
  endfor
  for each  $o \in O$ 
     $Bs_T = I \cap \varphi^-(o);$ 
     $Bs_F = I \cap \varphi^-(o^-);$ 
    for each  $s \in Bs_T$ 
      if  $pre(A) \subset s$ ;
      then return don't know;
    endfor
    for each  $s' \in Bs_F$ 
      if  $pre(A) \subset s'$ ;
      then return don't know;
    endfor
  endfor
  return failure;
end

```

ALGORITHM 1: Contingent plan-nonexistence algorithm.

Actually, the following analyses for the phase transitions of the contingent planning problems are based on the fundamental analysis of the variable model. Because it is very simple to solve the sort of contingent planning problems that the goal conditions are satisfied when there is not any actions, in this paper we only deal with the sort of contingent planning problems that the goal conditions are not satisfied by the initial belief state.

3. Contingent Plan Nonexistence Algorithm

In this section, we present the CONTINGENT PLAN-NONEXISTENCE algorithm for quickly proving that a contingent planning instance has no solutions. By analyzing the algorithm, the phase transition area is obtained. The framework of the CONTINGENT PLAN-NONEXISTENCE algorithm is presented in Algorithm 1.

The CONTINGENT PLAN-NONEXISTENCE algorithm firstly considers whether the preconditions of all actions are not in any state s in the initial belief state. If a state in the initial belief state does not satisfy the preconditions of all actions, then the instance has no solutions. Then the algorithm checks out whether a goal condition is not a postcondition of any action. If the postconditions of the actions do not cover all the goal conditions, then the instance has no solutions. The algorithm finally applies an observation to split the initial belief state into two belief states and checks out whether the preconditions of all actions are in any state in the two belief states, respectively. If there is a state in one of the belief states that satisfies the preconditions of all actions, then we cannot decide whether the instance has solutions or not. In addition, if there is not a state in all the split belief states that satisfies the preconditions of all actions, the instance has

no solutions. For a contingent planning instance with k initial states, m actions, b observations, and g goal conditions, the time complexity of the algorithm is $O((k + g + 2kb)m)$. Therefore, the CONTINGENT PLAN-NONEXISTENCE algorithm can decide that the instance has no solutions efficiently.

Theorem 7. *The contingent planning instance has no solution if CONTINGENT PLAN-NONEXISTENCE algorithm returns failure; the contingent planning instance cannot be determined whether it has solutions if CONTINGENT PLAN-NONEXISTENCE algorithm returns do not know.*

Proof. Let us first list the cases that the CONTINGENT PLAN-NONEXISTENCE algorithm returns *failure*. (1) The initial belief state involves a state that does not contain the preconditions of all the actions. As we know, if the preconditions of all the actions are not in the state, any action cannot be executed. Because in this paper we only deal with the sort of contingent planning problems that no goal conditions are satisfied by the initial belief state, a sequence of empty actions cannot be the solution of the instances. Therefore, the contingent planning instances do not have solutions. (2) There is a goal condition that is not an effect of any action, which means that the goal condition cannot be obtained by executing any action. Therefore, the contingent planning instances do not have solutions. (3) There is not a state in all the split belief states that satisfies the preconditions of all actions, which means that any action cannot be executed. Therefore, the contingent planning instances have no solutions. Then let us discuss the case that the CONTINGENT PLAN-NONEXISTENCE algorithm returns *do not know*. There is a state in one of the belief states that satisfies the preconditions of all actions, which means

that at least an action can be executed from the belief state. However, we cannot guarantee that all the goal conditions are satisfied by executing any conditional plan from each state in the initial belief state. Therefore, the contingent planning instance cannot be determined whether it has solutions.

By Theorem 7, we know that the CONTINGENT PLAN-NONEXISTENCE algorithm is correct, but it cannot guarantee the completeness. In other words, if a contingent planning instance has solutions, the CONTINGENT PLAN-NONEXISTENCE algorithm could not determine that the instance has solutions. However, we can prove in Theorem 8 that nearly all contingent planning instances have no solutions when σ is small enough (σ is a constant and $0 < \sigma < 1$). \square

Theorem 8. *For random contingent planning instances under the variable model, if the number of actions satisfies*

$$m \leq \theta_{ub}, \quad \theta_{ub} = \min \left(\frac{(1 - (1 - r/2n)^n)(-\ln \ln 1/\sigma)}{(1 - r/2n)^n}, \left(\frac{(2n - c)}{c} \right) \left(\ln g - \ln \ln \frac{1}{\sigma} \right) \right), \quad (4)$$

where n is the number of propositions; m is the number of actions; r and c , respectively, are the expected numbers of preconditions and postconditions within an action; g is the number of goal conditions; k is the number of initial states; σ ($0 < \sigma < 1$) is a constant, then the CONTINGENT PLAN-NONEXISTENCE algorithm proves that no solution exists for at least $1 - \sigma$ of the instances.

Proof. If we are to prove that the algorithm determines that no solution exists for at least $1 - \sigma$ of the instances, we only need to prove that the probability that every initial state satisfies the preconditions of any action, the probability that a goal condition is not a postcondition of any action, and the probability that every initial state in the belief state Bs_T (or Bs_F) satisfies the preconditions of any action are not more than σ . Suppose (4) is true, we have

$$m \leq \min \left(\frac{(1 - (1 - r/2n)^n)(-\ln \ln 1/\sigma)}{(1 - r/2n)^n}, \left(\frac{(2n - c)}{c} \right) \left(\ln g - \ln \ln \frac{1}{\sigma} \right) \right). \quad (5)$$

Because the initial state of the contingent planning problem is partially known, there exists a set of initial states, where the number of the initial states is k in this paper. Every initial state is made up of conditions, each of which is selected at random from the $2n$ conditions since there are n propositions. At first, the probability that every initial state satisfies the preconditions of any action is given as follows:

$(1 - r/2n)^n$: probability that a state satisfies the preconditions of an action;

$1 - (1 - r/2n)^n$: probability that a state does not satisfy the preconditions of an action;

$(1 - (1 - r/2n)^n)^m$: probability that a state does not satisfy the preconditions of any action;

$1 - (1 - (1 - r/2n)^n)^m$: probability that a state satisfies the preconditions of any action;

$(1 - (1 - (1 - r/2n)^n)^m)^k$: probability that every initial state satisfies the preconditions of any action.

Then, the probability that every initial state in the belief state Bs_T satisfies the preconditions of any action is given as follows, where the number of the states in Bs_T (or Bs_F) is x ($1 \leq x \leq k$),

$(1 - r/2n)^n$: probability that a state satisfies the preconditions of an action;

$1 - (1 - r/2n)^n$: probability that a state does not satisfy the preconditions of an action;

$(1 - (1 - r/2n)^n)^m$: probability that a state does not satisfy the preconditions of any action;

$1 - (1 - (1 - r/2n)^n)^m$: probability that a state satisfies the preconditions of any action;

$(1 - (1 - (1 - r/2n)^n)^m)^x$: probability that every initial state in the belief state Bs_T (or Bs_F) satisfies the preconditions of any action.

For the goal conditions, the g goal conditions are also selected at random from the $2n$ conditions. Then, the probability that a goal condition is not a postcondition of any action is as follows:

$c/2n$: probability that the goal condition is a postconditions of the action;

$1 - c/2n$: probability that the goal condition is not a postconditions of the action;

$(1 - c/2n)^m$: probability that the goal condition is not a postconditions of any action;

$1 - (1 - c/2n)^m$: probability that the goal condition is a postconditions of some action;

$(1 - (1 - c/2n)^m)^g$: probability that every goal condition is a postconditions of some action.

In the following, we will prove that the probability that every initial state satisfies the preconditions of any action and the probability that every initial state in the belief state Bs_T (or Bs_F) satisfies the preconditions of any action both are not more than σ . Because the probability that every initial state satisfies the preconditions of any action is less than the probability that every initial state in the belief state Bs_T (or Bs_F) satisfies the preconditions of any action, we only prove that the probability that every initial state in the belief state Bs_T (or Bs_F) satisfies the preconditions of any action is not more than σ .

By (5), we get

$$m \leq \frac{(1 - (1 - r/2n)^n)(-\ln \ln 1/\sigma)}{(1 - r/2n)^n}. \quad (6)$$

This is equivalent to

$$\ln\left(\frac{1}{\ln(1/\sigma)}\right) \geq \frac{m(1-r/2n)^n}{1-(1-r/2n)^n}, \quad (7)$$

$$\ln\left(\ln\left(\frac{1}{\sigma}\right)\right) \leq -\frac{m(1-r/2n)^n}{1-(1-r/2n)^n}. \quad (8)$$

Let $x = (1-r/2n)^n$. By (1), we have

$$e^{-(1-r/2n)^n/(1-(1-r/2n)^n)} \leq 1 - \left(1 - \frac{r}{2n}\right)^n. \quad (9)$$

From (8) and (9), we have

$$\ln\left(\ln\left(\frac{1}{\sigma}\right)\right) \leq m \ln\left(1 - \left(1 - \frac{r}{2n}\right)^n\right). \quad (10)$$

This is equivalent to

$$-\left(1 - \left(1 - \frac{r}{2n}\right)^n\right)^m \leq (\ln \sigma). \quad (11)$$

Let $x = (1 - (1 - r/2n)^n)^m$. By (2), we have

$$\ln\left(1 - \left(1 - \left(1 - \frac{r}{2n}\right)^n\right)^m\right) \leq -\left(1 - \left(1 - \frac{r}{2n}\right)^n\right)^m. \quad (12)$$

From (11) and (12), we have

$$\ln\left(1 - \left(1 - \left(1 - \frac{r}{2n}\right)^n\right)^m\right) \leq (\ln \sigma). \quad (13)$$

Simplify (13), and we get

$$1 - \left(1 - \left(1 - \frac{r}{2n}\right)^n\right)^m \leq \sigma. \quad (14)$$

In addition, for

$$\left(1 - \left(1 - \left(1 - \frac{r}{2n}\right)^n\right)^m\right)^x \leq 1 - \left(1 - \left(1 - \frac{r}{2n}\right)^n\right)^m; \quad (15)$$

therefore, we obtain

$$\left(1 - \left(1 - \left(1 - \frac{r}{2n}\right)^n\right)^m\right)^x \leq \sigma. \quad (16)$$

Thus, if the inequality of the theorem is satisfied, then the probability that every initial state in the belief state Bs_T (or Bs_F) does not satisfy the preconditions of any action and the probability that every initial state satisfies the preconditions of any action are at least $1 - \sigma$.

Finally, we prove that the probability that a goal condition is not a postcondition of any action is not more than σ .

From (5), we get

$$m \leq \left(\frac{(2n-c)}{c}\right) \left(\ln g - \ln \ln \frac{1}{\sigma}\right). \quad (17)$$

This is equivalent to

$$\frac{-mc}{(2n-c)} \geq \ln\left(\frac{\ln(1/\sigma)}{g}\right). \quad (18)$$

By (1), we get

$$\frac{-c}{(2n-c)} \leq \ln\left(1 - \frac{c}{2n}\right). \quad (19)$$

From (18) and (19), we have

$$\ln\left(\frac{\ln(1/\sigma)}{g}\right) \leq m \ln\left(1 - \frac{c}{2n}\right). \quad (20)$$

This is equivalent to

$$-\left(1 - \frac{c}{2n}\right)^m \leq \frac{(\ln \sigma)}{g}. \quad (21)$$

By (2), we get

$$\ln\left(1 - \left(1 - \frac{c}{2n}\right)^m\right) \leq -\left(1 - \frac{c}{2n}\right)^m. \quad (22)$$

From (21) and (22), we have

$$\ln\left(1 - \left(1 - \frac{c}{2n}\right)^m\right) \leq \frac{(\ln \sigma)}{g}. \quad (23)$$

Simplify (23), and we get

$$\left(1 - \left(1 - \frac{c}{2n}\right)^m\right)^g \leq \sigma. \quad (24)$$

Thus, if the inequality of the theorem is satisfied, then the probability that some goal condition is not a postcondition of any action is at least $1 - \sigma$.

Therefore, if the inequality of the theorem is satisfied, then the algorithm will determine that no solution exists for at least $1 - \sigma$ of the instances. \square

Theorem 8 presents an upper bound of the phase transition. If we can obtain a lower bound, the phase transition area of the contingent planning problems will be acquired. Therefore, in the next section, we will present the CONTINGENT PLAN-EXISTENCE algorithm and discuss the lower bound of the contingent planning problems by analyzing the algorithm.

4. Contingent Plan Existence Algorithm

In this section, we present the CONTINGENT PLAN-EXISTENCE algorithm for powerfully determining that a contingent planning instance has solutions. By analyzing the algorithm, the lower bound of the phase transition is obtained. The framework of the CONTINGENT PLAN-EXISTENCE algorithm is presented in Algorithm 2.

In the algorithm, N denotes the set of unsatisfied goal conditions, Y denotes the set of satisfied goal conditions, Bs denotes the considered belief state; $\text{add}(a)$ denotes the adding effects of the action a , $\text{del}(a)$ denotes the deleting effects of the action a , and $\text{applicable}(\text{Bs}, a)$ denotes the action a applicable in the belief state Bs . The algorithm firstly initializes N , Y , and Bs , that is, $N = G$, $Y = \text{NULL}$, and $\text{Bs} = I$. Then it checks whether all states in the considered belief states satisfy

```

Algorithm CONTINGENT PLAN-EXISTENCE ( $N, Y, Bs, A, O$ )
  flag = 0;
  for each  $s \in Bs$ 
    if  $N \notin s$ 
      then flag = 1;
    endfor
    if flag == 0
      then return true
    for each  $a \in A$ 
      if applicable( $Bs, a$ ) and  $N \cap add(a) \neq \emptyset$  and  $Y \cap del(a) = \emptyset$ 
        then  $Bs' = Bs + add(a) - del(a);$ 
         $N = N - (N \cap add(a));$ 
         $Y = Y + (N \cap add(a));$ 
        CONTINGENT PLAN-EXISTENCE ( $N, Y, Bs', A, O$ );
      endfor
    for each  $o \in O$ 
       $Bs_T = Bs \cap \varphi^-(o);$ 
       $Bs_F = Bs \cap \varphi^-(o^-);$ 
      if  $Bs_T \neq \emptyset$  and  $Bs_F \neq \emptyset$ 
        then CONTINGENT PLAN-EXISTENCE ( $N, Y, Bs_T, A, O$ ) and
          CONTINGENT PLAN-EXISTENCE ( $N, Y, Bs_F, A, O$ )
      endfor
    return don't know
  end

```

ALGORITHM 2: Contingent plan-existence algorithm.

the goal. If each state in the considered belief state satisfies the goal, the value of $flag$ will not change, which means that the instance has solutions. Otherwise, the algorithm considers whether an action can be applied to the considered belief state to ensure that the postconditions of the actions satisfy more goal conditions. If there exists such actions, the process is repeated. The algorithm finally checks whether there is an observation that can split the considered belief state into two nonnull belief states. If the observation does exist, the similar process is repeated in the two branches, and the instance has solutions only when the two branches return *true*. Except the above three cases, the algorithm cannot determine whether the contingent planning instance has solutions.

Theorem 9. *The contingent planning instance has solutions if the CONTINGENT PLAN-EXISTENCE algorithm returns true; the contingent planning instance cannot be determined whether it has solutions if the CONTINGENT PLAN-EXISTENCE algorithm returns do not know.*

Proof. Let us first discuss the case that the CONTINGENT PLAN-EXISTENCE algorithm returns *true*. When all the goal conditions are satisfied by a conditional plan from each state in the initial belief state, we can determine that the instance has solutions. Then we list the cases that the CONTINGENT PLAN-EXISTENCE algorithm returns *do not know*. (1) The considered belief state has no applicable actions. (2) The number of unsatisfied goal conditions does not decrease by executing all applicable actions. (3) One of branches that the observation splits returns *do not know*. The above three cases imply that we cannot determine whether the instance has solutions or not.

The CONTINGENT PLAN-EXISTENCE algorithm is correct, but we also cannot guarantee the completeness. In the following, we prove that almost all the contingent planning instances have solutions by analyzing the algorithm when σ is small enough (σ is a constant and $0 < \sigma < 1$). \square

Lemma 10. *Consider random planning instances under the variable model expecting that d of the g goal conditions are not satisfied, if*

$$m \geq e^{(1-(1-r/2n)^n)^k / (1-(1-(1-r/2n)^n)^k)} e^{c(g-d)/n} \times \left(\left(\frac{2n}{cd} \right) + 1 \right) \left(\ln \frac{1}{\sigma} \right), \quad (25)$$

where n is the number of propositions; m is the number of actions; r and c , respectively, are the expected numbers of preconditions and postconditions within an action; g is the number of goal conditions; k is the number of initial states; x is the number of states of an arbitrary belief state; σ ($0 < \sigma < 1$) is a constant, then the CONTINGENT PLAN-EXISTENCE algorithm applying some action will increase the number of satisfied goal conditions for at least $1 - \sigma$ of the instances.

Proof. At first, the probability that some action will increase the number of satisfied goal conditions is given as follows:

$(1 - r/2n)^n$: probability that a state satisfies the preconditions of an action;

$1 - (1 - r/2n)^n$: probability that a state does not satisfy the preconditions of an action;

$(1 - (1 - r/2n)^n)^x$: the least probability that every state in an arbitrary belief state does not satisfy the preconditions of an action;

$1 - (1 - (1 - r/2n)^n)^x$: probability that every state in the arbitrary belief state satisfies the preconditions of an action;

$1 - (1 - (1 - r/2n)^n)^k$: the most probability that every state in the arbitrary belief state satisfies the preconditions of an action. Because in this paper we research on the contingent planning problem whose actions have deterministic effects, the belief state during the execution has most k states;

$(1 - c/2n)^{g-d}$: probability that the postconditions of an action are consistent with the $g-d$ goal conditions already achieved;

$1 - (1 - c/2n)^d$: probability that the postconditions of an action achieve at least one of d remaining goal conditions.

Thus, the probability p that a particular action can be applied will not exclude any satisfied goals and will achieve at least one more goal condition which is

$$p = \left(1 - \left(1 - \left(1 - \frac{r}{2n}\right)^n\right)^k\right) \left(1 - \frac{c}{2n}\right)^{g-d} \left(1 - \left(1 - \frac{c}{2n}\right)^d\right). \quad (26)$$

$1-p$ is the probability that the action is missing one or more of these properties, and $(1-p)^m$ is the probability that m actions are unsatisfactory.

Because $(1-p)^m \leq e^{-pm} \leq e^{-\ln(1/\sigma)} = \sigma$, then there will be some satisfactory action with probability at least $1-\sigma$. The inequality is satisfied if $m \geq (1/p)(\ln 1/\sigma)$.

By (1), we get

$$\begin{aligned} 1 - \left(1 - \left(1 - \frac{r}{2n}\right)^n\right)^k &\geq e^{-(1-(1-r/2n)^n)^k/(1-(1-r/2n)^n)^k} \\ \left(1 - \frac{c}{2n}\right)^{g-d} &\geq e^{-c(g-d)/(2n-r)} \geq e^{-c(g-d)/n}. \end{aligned} \quad (27)$$

By (3), we get

$$1 - \left(1 - \frac{c}{2n}\right)^d \geq \frac{cd}{2n + cd}. \quad (28)$$

Then, finally,

$$\begin{aligned} \left(\frac{1}{p}\right) \left(\ln \frac{1}{\sigma}\right) &= e^{(1-(1-r/2n)^n)^k/(1-(1-r/2n)^n)^k} e^{c(g-d)/n} \\ &\times \left(\left(\frac{2n}{cd}\right) + 1\right) \left(\ln \frac{1}{\sigma}\right). \end{aligned} \quad (29)$$

Thus, if the inequality is satisfied by the number of the actions, the CONTINGENT PLAN-EXISTENCE algorithm applying some action will increase the number of satisfied goal conditions for at least $1-\sigma$ of the instances. \square

Theorem 11. Consider random planning instances under the variable model if:

$$\begin{aligned} m \geq \theta_{lb}, \quad \theta_{lb} &= e^{1/(1-(1-r/2n)^n)^k} e^{cg/n} \\ &\times \left(\left(\frac{2n}{c}\right) + 1\right) \left(\ln \frac{g}{\sigma}\right), \end{aligned} \quad (30)$$

where n is the number of propositions; m is the number of actions; r and c , respectively, are the expected numbers of preconditions and postconditions within an action; g is the number of goal conditions; k is the number of initial states; σ ($0 < \sigma < 1$) is a constant, the CONTINGENT PLAN-EXISTENCE algorithm will find solutions for at least $1-\sigma$ of the instances.

Proof. For a contingent planning instance with g goal conditions, the number of satisfied goal conditions will be increased at most g times. If each increase occurs with probability at least $(1-\sigma)/g$, g increase (the most possible) will occur with probability at least $1-\sigma$.

Thus, Lemma 10 can be applied using σ/g instead of σ . Maximizing over the g goal conditions leads to

$$\begin{aligned} \max_{d=1}^g &e^{(1-(1-r/2n)^n)^k/(1-(1-r/2n)^n)^k} e^{c(g-d)/n} \\ &\times \left(\left(\frac{2n}{cd}\right) + 1\right) \left(\ln \frac{g}{\sigma}\right) \\ &\leq e^{1/(1-(1-r/2n)^n)^k} e^{cg/n} \left(\left(\frac{2n}{c}\right) + 1\right) \left(\ln \frac{g}{\sigma}\right). \end{aligned} \quad (31)$$

Therefore, if the inequality is satisfied by the number of the actions, the algorithm will find solutions for at least $1-\sigma$ of the instances. \square

Theorem 11 shows that when the number of actions is not lower than θ_{lb} , the CONTINGENT PLAN-EXISTENCE algorithm can prove that nearly all the contingent planning instances have solutions.

5. Experimental Results

In this section, we take experiments on the contingent planning instances under the variable model in order to investigate the relationship between the densities (ratio of the number of actions to the number of propositions) and the effectiveness of the contingent planning instances. In the experiment, we generate a large collection of instances with three preconditions and two postconditions within an action, 2^i (i is a constant) initial states, one observation, and two goal conditions. In addition, the propositions are 6, 8, 10, 12, 14, 16, and 18, respectively. All experiments are run on a cluster of 2.4 GHz Intel Xeon machines with 2 GB memory running Linux CentOS 5.4. Figure 1 illustrates the relationship between the densities and the effectiveness of the contingent planning instances. From the results we can see that the effectiveness grows when the density increases and the numbers of propositions and goal conditions are constant. Furthermore, the experiments show that there is a transition from insoluble

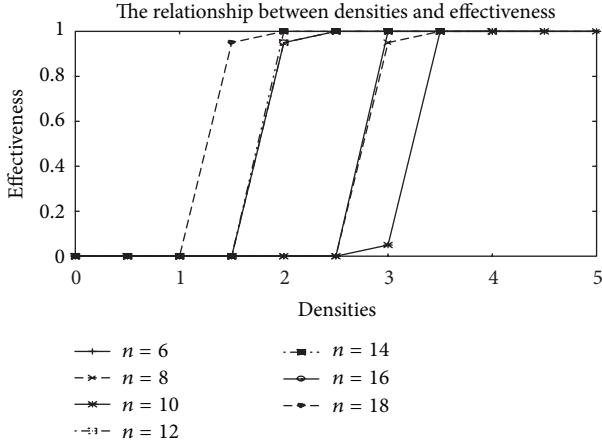


FIGURE 1: The Relationship between densities and effectiveness of contingent planning problems.

to soluble instances as the density grows and the transition points drop with the propositions growing.

6. Conclusions

In this paper, we present analyses for the phase transitions of the contingent planning problems. By analyzing the CONTINGENT PLAN-NONEXISTENCE algorithm and the CONTINGENT PLAN-EXISTENCE algorithm, quantitative results are obtained. If the number of actions is not greater than θ_{ub} , the CONTINGENT PLAN-NONEXISTENCE algorithm can prove that almost all the contingent planning instances have no solution; if the number of actions is not lower than θ_{lb} , the CONTINGENT PLAN-EXISTENCE algorithm can prove that almost all the contingent planning instances have solutions. The results of the experiments also show that there exist the phase transition phenomena.

Acknowledgments

This work was fully supported by the National Natural Science Foundation of China (Grants nos. 11226275 and 61070084), the Fundamental Research Funds for the Central Universities (Grant no. 11QNJJ006), the Ministry of Education (Grant no. 20120043120017), the Postdoctoral Science Foundation of China (2012M520658), and the opening fund of top key discipline of computer software and theory in Zhejiang Provincial Colleges at Zhejiang Normal University (Grant no. ZSDZZZZXK37).

References

- [1] S. A. Cook, "The complexity of theorem proving procedures," in *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing*, M. A. Harrison, R. B. Banerji, and J. D. Ullman, Eds., pp. 151–158, New York, NY, USA, 1971.
- [2] J. M. Crawford and L. D. Auton, "Experimental results on the crossover point in satisfiability problems," in *Proceedings of the 11th National Conference on Artificial Intelligence*, W. R. Swartout, Ed., pp. 21–27, Washington, DC, USA, July 1993.
- [3] B. Selman, D. G. Mitchell, and H. J. Levesque, "Generating hard satisfiability problems," *Artificial Intelligence*, vol. 81, no. 1-2, pp. 17–29, 1996.
- [4] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky, "Determining computational complexity from characteristic 'phase transitions,'" *Nature*, vol. 400, no. 6740, pp. 133–137, 1999.
- [5] D. Mitchell, B. Selman, and H. Levesque, "Hard and easy distributions of SAT problems," in *Proceedings of the 10th National Conference on Artificial Intelligence (AAAI '92)*, W. R. Swartout, Ed., pp. 459–465, San Jose, Calif, USA, July 1992.
- [6] E. Friedgut, "Sharp thresholds of graph properties, and the k-sat problem," *Journal of the American Mathematical Society*, vol. 12, no. 4, pp. 1017–1054, 1999.
- [7] A. Frieze and S. Suen, "Analysis of two simple heuristics on a random instance of k-SAT," *Journal of Algorithms*, vol. 20, no. 2, pp. 312–355, 1996.
- [8] L. M. Kirousis, E. Kranakis, D. Krizanc, and Y. C. Stamatiou, "Approximating the unsatisfiability threshold of random formulas," *Random Structures and Algorithms*, vol. 12, no. 3, pp. 253–269, 1998.
- [9] I. P. Gent and T. Walsh, "The TSP phase transition," *Artificial Intelligence*, vol. 88, no. 1-2, pp. 349–358, 1996.
- [10] K. Xu and W. Li, "Exact phase transitions in random constraint satisfaction problems," *Journal of Artificial Intelligence Research*, vol. 12, pp. 93–103, 2000.
- [11] D. Coppersmith, D. Gamarnik, M. Hajiaghayi, and G. B. Sorkin, "Random MAX SAT, random MAX CUT, and their phase transitions," *Random Structures and Algorithms*, vol. 24, no. 4, pp. 502–545, 2004.
- [12] J. Rintanen, "Phase transitions in classical planning: an experimental study," in *Proceedings of the 14th International Conference on Automated Planning and Scheduling*, S. Zilberstein, J. Koehler, and S. Koenig, Eds., pp. 101–110, British Columbia, Canada, 2004.
- [13] J. Zhou, P. Huang, M. Yin, and C. Zhou, "Phase transitions of expspace-complete problems," *International Journal of Foundations of Computer Science*, vol. 21, no. 6, pp. 1073–1088, 2010.
- [14] J. Zhou, M. Yin, X. Li, and J. Wang, "Phase transitions of expspace-complete problems: a further step," *International Journal of Foundations of Computer Science*, vol. 23, no. 1, pp. 173–184, 2012.
- [15] H. Palacios and H. Geffner, "From conformant into classical planning: efficient translations that may be complete too," in *Proceedings of the 17th International Conference on Automated Planning and Scheduling (ICAPS '07)*, M. S. Boddy, M. Fox, and S. Thiebaux, Eds., pp. 264–271, Providence, RI, USA, September 2007.
- [16] D. Bryce, S. Kambhampati, and D. E. Smith, "Planning graph heuristics for belief space search," *Journal of Artificial Intelligence Research*, vol. 26, pp. 35–99, 2006.
- [17] M. Helmert, "The fast downward planning system," *Journal of Artificial Intelligence Research*, vol. 26, pp. 191–246, 2006.
- [18] U. Kuter, D. Nau, M. Pistore, and P. Traverso, "Task decomposition on abstract states, for planning under nondeterminism," *Artificial Intelligence*, vol. 173, no. 5-6, pp. 669–695, 2009.
- [19] J.-P. Zhou, M.-H. Yin, W.-X. Gu, and J.-G. Sun, "Research on decreasing observation variables for strong planning under partial observation," *Journal of Software*, vol. 20, no. 2, pp. 290–304, 2009.

- [20] R. P. Goldman and M. S. Boddy, "Expressive planning and explicit knowledge," in *Proceedings of the 3rd International Conference on Artificial Intelligence Planning Systems*, B. Drabble, Ed., pp. 110–117, Edinburgh, Scotland, 1996.
- [21] J. Hoffmann and R. I. Brafman, "Contingent planning via heuristic forward search with implicit belief states," in *Proceedings of the 15th International Conference on Automated Planning and Scheduling*, S. Biundo, K. L. Myers, and K. Rajan, Eds., pp. 71–80, Monterey, Calif, USA, 2005.

Research Article

Reconfigurable Architecture for Elliptic Curve Cryptography Using FPGA

A. Kaleel Rahuman and G. Athisha

Department of Electronics and Communication Engineering, PSNA College of Engineering and Technology, Dindigul-624 622, Tamil Nadu, India

Correspondence should be addressed to A. Kaleel Rahuman; kaleel23@gmail.com

Received 5 April 2013; Accepted 4 July 2013

Academic Editor: William Guo

Copyright © 2013 A. Kaleel Rahuman and G. Athisha. 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 high performance of an elliptic curve (EC) crypto system depends efficiently on the arithmetic in the underlying finite field. We have to propose and compare three levels of Galois Field $GF(2^{163})$, $GF(2^{193})$, and $GF(2^{256})$. The proposed architecture is based on Lopez-Dahab elliptic curve point multiplication algorithm, which uses Gaussian normal basis for $GF(2^{163})$ field arithmetic. The proposed $GF(2^{193})$ is based on an efficient Montgomery add and double algorithm, also the Karatsuba-Ofman multiplier and Itoh-Tsujii algorithm are used as the inverse component. The hardware design is based on optimized finite state machine (FSM), with a single cycle 193 bits multiplier, field adder, and field squarer. The another proposed architecture $GF(2^{256})$ is based on applications for which compactness is more important than speed. The FPGA's dedicated multipliers and carry-chain logic are used to obtain the small data path. The different optimization at the hardware level improves the acceleration of the ECC scalar multiplication, increases frequency and the speed of operation such as key generation, encryption, and decryption. Finally, we have to implement our design using Xilinx XC4VLX200 FPGA device.

1. Introduction

Many hardware designs of elliptic curve cryptography have been developed, aiming to accelerate the scalar multiplication processes, mainly those based on the field programmable gate arrays (FPGAs). The application field of FPGAs has clearly outgrown the prototype-only use. More and more FPGA implementations are in an environment which used to be ASIC-only territory. When these applications are implemented on an FPGA, they need secure data communication. In this rapidly changing environment, the reconfigurability of an FPGA is a very useful feature which is not available on an ASIC. Secure public key authentication and digital signatures are increasingly important for electronic communications and coerce, and they are required not only on high powered desktop computers, but also on smart cards and wireless devices with severely constrained memory and processing capabilities. Cryptography offers a robust solution for IT security services in terms of confidentiality, data integrity, authenticity, and nonrepudiation. In fact, security deals

mainly with the ability to face counterattacks [1], while speed and area, which represent the eternal trade-off, concern the ability to make intensive cryptographic processes, while keeping used hardware as low as possible. In other words, it is the ability to embed a strategic and strong algorithm in very few hardware, that is, finding an optimal solution to the one to many problem: portability against power consumption, speed against area, but the main issue in cryptography is security.

In the last decade, the approach of hardware implementing elliptic curve cryptography (ECC) algorithm knew a very concentrated contest, due essentially to the requirements of security, speed, and area constraints. Cryptography has become one of the most important fields in our life, due essentially to two factors: increase in secrecy and increase in breaking code or hackers in the other side. Organizations tend to increase their benefits by keeping their information system as transparent as possible. On the other hand, hackers and code or key breakers are being organized in kind of unofficial groups; this leads to being a step ahead before getting the codes breakdown. Scientists are tending to complicate the

```

Input:  $A, B \in GF(2^m)$ 
Output:  $D = (D_0, D_1, D_2, \dots, D_{m-1}) \in GF(2^m)$ ,  $D = A \cdot B$ 
(1) For  $0 \leq t \leq m - 1$ 
(2) For  $0 \leq s \leq m - 1$ 
(3)  $D_{s+t+1} \leftarrow y_{s,s+1}$ 
(4) end for
(5) end for
(6) return  $D$ 

```

ALGORITHM 1: Bit-level multiplication algorithm for $GF(2^m)$.

reverse engineering process of the encryption system, at the same time, keeping encryption keys as low as possible. This issue is being tackled by many mathematicians, mainly those working on elliptic curves [2].

Elliptic curve cryptosystems possess a number of degrees of freedom like Galois field characteristic, extension degree, elliptic curve parameters, or the fixed point generating the working subgroup on the curve. The beauty of this new field is potentially related to the simplicity of the operators used in the encryption process, to the nonsecure transmission constraints used in the exchange of the keys and to the enhanced complexity that might face hackers when unwanted information goes out of the organization. This paper focuses on the high-performance comparisons of hardware design of ECC over $GF(2^{163})$, $GF(2^{193})$, and $GF(2^{256})$ (Table 4).

2. Elliptic Curve Cryptography

In 1985, Koblitz and Miller introduced the use of elliptic curves in public key cryptography called Elliptic curve Cryptography (ECC). Basically, the main operation of elliptic curves consists of multiplying a point by a scalar in order to get a second point; the complexity arises from the fact that, given the initial point and the final point, the scalar could not be deduced, leading to a very difficult problem of reversibility, or cryptoanalysis, called also the elliptic curve discrete logarithm problem [1].

The ECC algorithms with their small key sizes present nowadays the best challenge for cryptanalysis problems compared to RSA or AES, thus dealing with ECC will lead to smaller area hardware, less bandwidth use, and more secure transactions.

The attractiveness of ECC algorithms is that they operate on a Galois Field (GF), by means of two simple operations, known as the field addition and field multiplication, which define a ring over $GF(P^m)$ where P and m are primes. In the particular case, where we deal with hardware designs, a binary field is preferred, where the couple, (P, m) , defines the set of elliptic curves. In this work, $P = 2$ and $m = 163, 193$, and 256 .

In this paper, we propose a high-performance elliptic curve cryptographic processor over $GF(2^m)$, that is, $GF(2^{163})$, $GF(2^{193})$, and $GF(2^{256})$. The proposed architecture is based on a modified Lopez-Dahab elliptic curve point multiplication algorithm and uses GNB for $GF(2^m)$ field arithmetic. Three major characteristics of the proposed architecture use

fast arithmetic units based on a word-level multiplier which adopts a parallelized point doubling and point addition unit with uniform addressing mode to utilize the benefits of GNB representation. Therefore, the proposed architecture leads to a considerable reduction of computational delay. The proposed architecture has the feature of modularity and a simple control structure; it is well suited to VLSI implementations (see Algorithm 1).

In this research, we present a hardware design of the elliptic curve cryptography scheme, using Montgomery scalar multiplication based on the “add and double” algorithm, targeting as a primary goal of an increase in the speed of the hardware and an optimization in the ensuing inverse component.

3. Materials and Methods

3.1. Hardware Design. The strategy of hardware executing the ECC algorithms repos on the ability of making the scalar multiplication in the $GF(2^m)$ in a very few clock [1]. While increasing m , implementations become very time and resource consuming. Most of the known architectures concern the acceleration of the multiplication process by modifying the elliptic equations by changing the Z coordinate term [3], or by multiplication scalability [4], or by using many serial and parallel Arithmetic units [5], or using High parallel Karatsuba Multiplier [6], those based on the Massy-Omura multipliers, or the work based on a hybrid multipliers approach, also some parallel approach approaches, or the new word level structure, or through the systolic architecture, or by using the half and add method, or by parallelizing both the add and double Montgomery algorithms [7].

The second problem concerns the inversion based on the Fermat little theorem, or the almost inverse algorithm based on Kali ski’s research [8]. In order to concentrate on one of the problems, some modifications have been done on the ECC equations in order to postpone inversion to the last stage, while dealing only with the multiplication process.

3.2. Elliptic Curve Mathematical Background. ECC is based on the discrete logarithm problem applied to elliptic curves over a finite field. In particular, for an elliptic curve E that relies on the fact, it is computationally easy to find that

$$Q = kx \square P, \quad (1)$$

```

Input: Elliptic curve domain parameters ( $p, E, P, n$ )
Output: Public key  $Q$  and private key  $k$ 
(1) Select  $k \in [1, n - 1]$ 
(2) Compute  $Q = kP$ 
(3) Return( $Q, k$ )

```

ALGORITHM 2: Elliptic curve key pair generation.

where P and Q = Points of the elliptic curve E and their coordinates belong to the underlying $\text{GF}(2^m)$, k = A scalar that belongs to the set of numbers $\{1 \dots G - 1\}$, G is the order of the curve E .

Elliptic Curve Key Generation. Let E be an elliptic curve defined over a finite field $\text{GF}(2^m)$. Let P be a point in $E(\text{GF}(2^m))$, and suppose that P has prime order n . Then, the cyclic subgroup of $E(\text{GF}(2^m))$ generated by P is $P' = \{\infty, P, 2P, 3P, \dots, (n - 1)P\}$. The prime p , the equation of the elliptic curve E , and the point P and its order n are the public domain parameters. A private key is an integer k that is selected uniformly at random from the interval $[1, n - 1]$, and the corresponding public key is $Q = kP$ (see Algorithm 2).

An encryption is described in what follows.

Encryption. (1) User A-Alice first selects a random generator point (x, y) lying on the elliptic curve.

(2) Message (M) to be encrypted is coded on to an elliptic curve point $Pm = (xm, ym)$.

(3) Alice selects a random private key “ nA ” and then computes the public key as

$$PA = nA(x, y). \quad (2)$$

(4) To encrypt her message, Alice uses her private key and Bob's (user B) public key.

(5) The encrypted message denoted by Cm is created as follows:

$$Cm = \{PA, (Pm + nA \cdot PB)\} \quad (3)$$

PB is the public key of Bob—user B.

The sender transmits the points $\{PA, (Pm + nA \cdot PB)\}$ to the recipient who uses her private key k to compute $kPA = k(nA \cdot (x, y)) = nA(k \cdot (x, y)) = nA \cdot R$, where R is the public key of the recipient.

The algorithm for an encryption is described in the following.

As it can be seen from the previous algorithm, point multiplication plays a major role during the encryption process. The same hold during decryption too. The encrypted message is then communicated to the receiver. The receiver, Bob, then decrypts the message using the decryption mechanism [9].

A decryption at the receiver end is as follows.

Decryption. When Bob receives the encrypted message, he first multiplies the public key of Alice, which happens to be the first point in the encrypted message with his private key NB .

- (1) When Bob receives the encrypted message, he first multiplies the public key of Alice, which happens to be the first point in the encrypted message with his private key NB .
- (2) The result of this is then subtracted from the second point, the cipher text.
- (3) This gives him the original message Pm .

Importance of Elliptic Curve Cryptography. There are several criteria that need to be considered when selecting a family of public key schemes for a specific application. The principal ones are as follows:

- (1) functionality,
- (2) security,
- (3) performance.

Measuring the Efficiency of Algorithms. The efficiency of an algorithm is measured by the scarce resources it consumes. Typically, the measure used is time, but sometimes other measures such as space and number of processors are also considered. It is reasonable to expect that an algorithm consumes greater resources for larger inputs, and the efficiency of an algorithm is therefore described as a function of the input size. Here, the size is defined to be the number of bits needed to represent the input using a reasonable encoding.

In the affine coordinate representation, a finite field point on $\text{GF}(2^m)$ is specified by two coordinates x and y both belonging to $\text{GF}(2^m)$ satisfying (4). The point at infinity has no affine coordinates.

In most ECC hardware designs, the choice of using three coordinates responds on avoiding the periodic division of (5), which consumes a lot of resources in terms of execution cycles, as well as memory and power consumption.

The advantage here is, Bob's private key is only known to him and not to anyone else and therefore only Bob can extract the original message by subtracting the product of his private key and Alice's public key with the second point [7].

Nowadays, there is no known algorithm able to compute k given P and Q in a subexponential time. The equation of a nonsingular elliptic curve with the underlying field $\text{GF}(2^m)$ is presented in (4). It is formed by choosing the elements “ a ” and “ b ” within $\text{GF}(2^m)$ with

$$y^2 + xy = x^3 + ax^2 + b. \quad (4)$$

A point is converted from a couple of coordinates to a triple system of coordinates using one of the transforms of

$$x3 = \left(\frac{y1 + y2}{x1 + x2} \right)^2 + \left(\frac{y1 + y2}{x1 + x2} \right) + x1 + x2 + a, \quad P \neq Q,$$

$$x3 = x1^2 + \left(\frac{b}{x1^2} \right), \quad P = Q,$$

$$\begin{aligned}
y3 &= \left(\frac{y_1 + y_2}{x_1 + x_2} \right) (x_1 + x_3) + x_3 + y_1, \quad P \neq Q, \\
y3 &= x_1^2 + \left(x_1 + \frac{y_1}{x_1} \right) x_3, \quad P = Q.
\end{aligned} \tag{5}$$

Thus, a point $P(x, y)$ is mapped into $P(x, y, z)$, that is; a third projective coordinate is introduced in order to “flatten” the equations and avoid the division. Projective coordinates allow us to eliminate the need for performing inversion. The startup transformation required for the design is simply done by initializing X, Y , and Z as follows: [10]

$$\{X = \square x, Y = \square y, Z = \square 1 \square \square \square \square \square \square\}. \tag{6}$$

Introducing the new tricoordinates into (4) becomes

$$Y^2 + XYZ = X^3Z + aX^2Z^2 + bZ^4. \tag{7}$$

The VHDL implementation will be based now on (7). After completion of the successive operations of addition and multiplication, back to two affine coordinates as follows:

$$\left\{ x = \frac{X}{z}, \quad y = \frac{Y}{z^2} \right\}. \tag{8}$$

In order to make the different computations, the Montgomery point doubling and Montgomery point addition algorithms are used, mainly through the ingenious observation of Montgomery, which states that the Y coordinate does not participate into the computations and can be delayed to the first stage and it working with two projective coordinates [11] (see Algorithm 3).

In the Decryption, $Madd()$ function is the point addition operation on the elliptic curve, $Mdouble()$ is the point doubling computation, and $Mxy()$ is the conversion of projective coordinates to affine coordinates. The reader is referred to (Lopez and Dahab 1999) [12] for detailed explanation. Function $Madd()$, $Mdouble()$, and $Mxy()$ in the Decryption are defined as follows:

```

Madd(x1, y1, x2, y2, x)
{X ← x1z2x2z1 + x(x1z2 + x2z1)2;
Z ← (x1z2 + x2z1)2;
return(X, Z);
}.

```

Requiring 1 field squaring operations, 4 field multiplications, One has and two simple field additions.

```

Mdouble(x1, z1, b)
{X ← x14 + bz14;
Z ← x12z14;
return (X, Z);
}.

```

Requiring 4 field squaring operations, 2 field multiplications, and one simple field addition.

Input: $k = (k_{n-1}, k_{n-2}, \dots, k_1, k_0)_2$ with $k_{n-1} = 1$, $P(x, y) \in E(\text{GF}(2^{m}))$.

Output: $Q = KP$.

- (1) Set $x_1 = x, z_1 = 1, x_2 = x^4 + b, z_2 = x^2$;
- (2) for $i = n - 2$ to 0 do
- (3) if $k_i = 1$ then
- (4) $(x_1, z_1) \leftarrow Madd(x_1, z_1, x_2, z_2, x)$;
 $(x_2, z_2) \leftarrow Mdouble(x_2, z_2, b)$;
- (5) else
- (6) $(x_2, z_2) \leftarrow Madd(x_2, z_2, x_1, z_1, x)$;
 $(x_1, z_1) \leftarrow Mdouble(x_1, z_1, b)$;
- (7) end if
- (8) end for
- (9) $Q \leftarrow Mxy(x_1, z_1, x_2, z_2, x, y)$;
- (10) return Q ;

ALGORITHM 3: Montgomery scalar multiplication algorithm.

$Mxy(x_1, z_1, x_2, z_2, x, y)$

```

{xk ← x1/z1;
yk ← [x2 + y + (x + x1/z1)(x + x2/z2)](x + xk)/x = y;
return (xk, yk)
}.
```

In these functions, (x, y) is the coordinate of the original point P , which is fixed during the calculation of kP ; (x_k, y_k) is the coordinate of $kP \cdot k$ is represented on an m bits register. The three basic functions in turn rely on finite field operations such as addition, multiplication, and inversion.

The inversion in $\text{GF}(2^{193})$, required at the final stage, could be realized in one of the two known methods, either via the extended Euclidean algorithm, or by Fermat's theorem which states that knowing after proof that $A^{2^m} - 1 = 1$ leads to consider that $A^{-1} = A^{2^{m-2}}$ is also factual.

Thus, in order to compute the inverse of one element in $\text{GF}(2^{193})$, one needs to take the power of this element ($2^{193} - 2$) times. By using the Itoh-Tsujii algorithm based on the add and multiply Method leads to realize the inverse as presented in Table 3 [13].

C. ECC Components. The ECC processor shown in Figure 1 consists of eight main components. Eight components are host interface (HI), data memory, register file, instruction memory, control-1, control-2, AU-1, and AU-2. The HI communicates with host processor. Processor transmits all parameters for kp to HI with start signal and receives “ kp ” results and end signal. In the proposed work, we have used Intel 32-bit processor as host processor. Therefore, HI is 32-bit interface. The HI transmits all parameters for computing elliptic curve point doubling and point addition to register file and receives “ kp ” results from data memory. The data memory consists of 8×163 -bit dual port block RAM and the instruction memory contains 13 microcode sequences of 11-bits word, respectively. Thus, these two block memories are to compute coordinate conversion. For high-performance implementation of point doubling and addition, we add

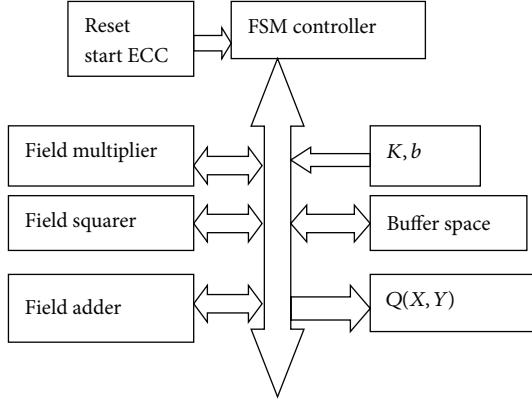


FIGURE 1: ECC components block diagram.

7×163 -bit register file, which receives data from HI and transmits temporary computation results (X_1, X_2, Z_1, Z_2) to data memory. The AU-1 is used for point doubling and addition and controlled by control-1. The AU-2 performs the coordinate conversion in algorithm. The control-2 receives operation code from instruction memory and generates control signals for AU-2, HI, and Data memory. “L” gives the number of required clock cycles to perform elliptic curve point multiplication over $\text{GF}(2^{163})$, where we assumed the digit size $\omega = 55$; that is, $L = 3$.

The 193 bits ECC components have been developed using the VHDL language. The different components forming the design are as follows: A 193 bits adder which is a simple 193 “2 bits” Xors. A 193 bits modulo which is an xor-array evaluated through a Matlab script as an input-output matrix, through polynomial reduction using the National Institute of Standards and Technology (NIST) proposed polynomial $P(x) = x^{193} + x^{15} + 1$. A 193 bits squarer that has also been generated from a Matlab script. A 193 bits modified version of the Karatsuba-Ofman multiplier circuit that is based on splitting the operands into 3 identical operands (High (H), Middle (M), and Low (L) bits). A Galois inverter circuit requiring 21 power squaring 9 field multiplications.

The 256 bits ECC components have developed a system describes the architecture of the most crucial component in ECDSA, namely that the one that implements elliptic curve operations and modular operations over a finite field $\text{GF}(P)$. For this purpose, a flexible, yet compact, elliptic curve processor is developed for applications where speed is of minor importance. The processor is optimized for FPGA families that were introduced to the market from the year 2003 on. But that are still used in many new products. These FPGAs also contain some Hard-IPs (HIPS). The presented processor uses Block RAM and multiplier HIPS, which are available on the majority of FPGA devices. The proposed system is the design of hardware that executes the ECC algorithm that reposes on the ability of making the modular operation over the $\text{GF}(2^{256})$. The research was based on using the efficient Montgomery ladder algorithm, ECDSA algorithm for EC point multiplication. In this system achieved compact architecture.

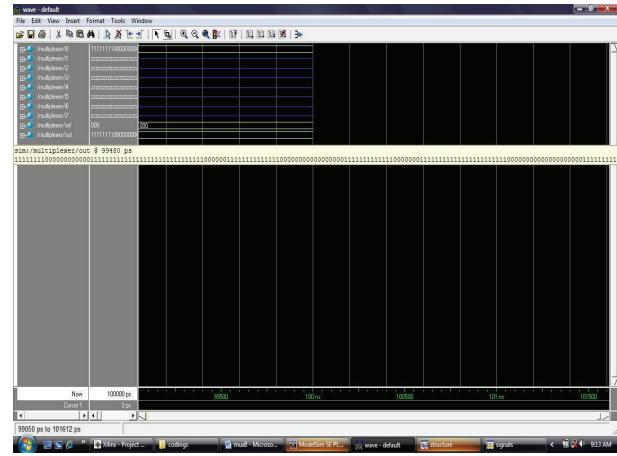


FIGURE 2: Simulation waveform of multiplexer.

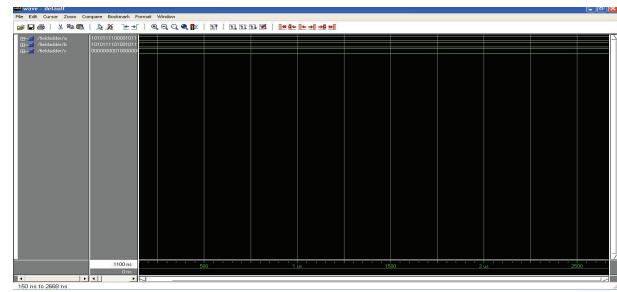


FIGURE 3: Simulation waveform of field adder.

In Figure 2 the efficient implementation of finite field arithmetic in elliptic curve system. The implementation uses dedicated multiplexer on the FPGA.

In Figure 3 the elliptic curve system achieved high throughput rates. The implementation uses dedicated field adder on the FPGA.

In Figure 4 design of an efficient Field multiplexer over $\text{GF}(2^{163})$ using Gaussian Normal Basis.

In Figure 5 the elliptic curve system achieved high performance. The implementation uses dedicated field squarer on the FPGA.

4. Results and Performance Comparisons

We present the respective estimated number of cycles, required for each part of the algorithm at each stage of FSM controller (Table I).

Working with 193 bits and 2^{193} order numbers or more is not a direct way and checking the results is very bulky. In this matter, different Matlab scripts with similar input/output behavior to the VHDL programming have been written, in order to compare the execution steps, as well the final results; timing is not taken into consideration in this specific stage.

Synthesis Result of Multiplexer

Speed Grade: -6

Minimum period: no path found

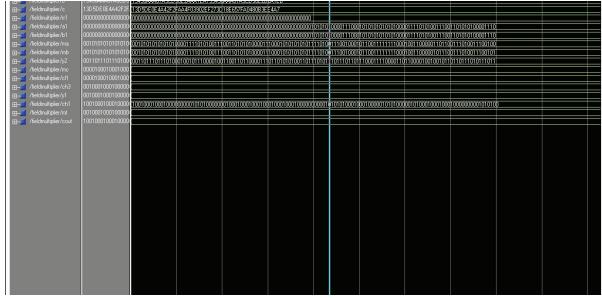


FIGURE 5: Simulation waveform of field squarer.

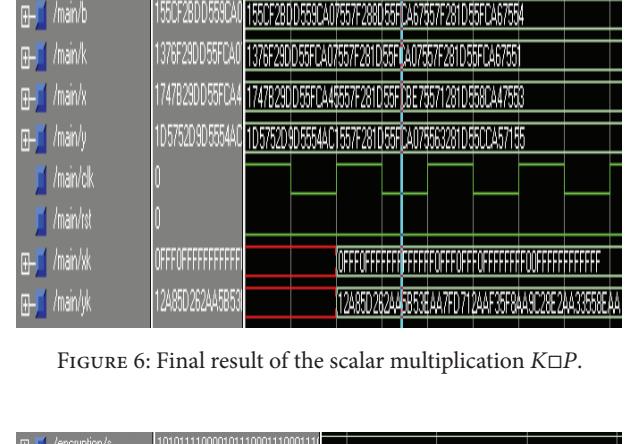
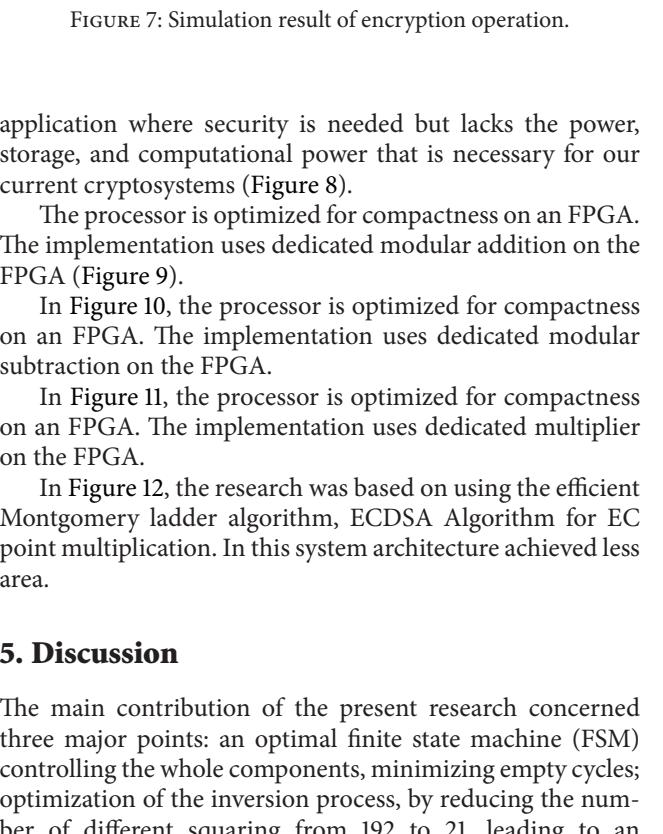
FIGURE 6: Final result of the scalar multiplication $K \square P$.

FIGURE 7: Simulation result of encryption operation.

Figure 6 and Table 5 show the output results of the ECC scalar multiplication for a “193 bits” arbitrary value of k . The frequency of encryption operation is 1930 MHZ and speed of operation also increases (Figure 7). It can be used in any application where security is needed but lacks the power, storage, and computational power that is necessary for our current cryptosystems.

The frequency of decryption operation is 1930 MHZ and the speed of operation also increases. It can be used in any

application where security is needed but lacks the power, storage, and computational power that is necessary for our current cryptosystems (Figure 8).

The processor is optimized for compactness on an FPGA. The implementation uses dedicated modular addition on the FPGA (Figure 9).

In Figure 10, the processor is optimized for compactness on an FPGA. The implementation uses dedicated modular subtraction on the FPGA.

In Figure 11, the processor is optimized for compactness on an FPGA. The implementation uses dedicated multiplier on the FPGA.

In Figure 12, the research was based on using the efficient Montgomery ladder algorithm, ECDSA Algorithm for EC point multiplication. In this system architecture achieved less area.

5. Discussion

The main contribution of the present research concerned three major points: an optimal finite state machine (FSM) controlling the whole components, minimizing empty cycles; optimization of the inversion process, by reducing the number of different squaring from 192 to 21, leading to an inversion; separation of the data path routing from the control part, in order to modify only the multiplier, the squarer, the adder and the modulo components.

The results we have obtained are very encouraging and will impact our decision on the embedding of larger encryption schemes, mainly the extension to the NIST proposed curves (233, 283, 409, and 571), taking into account the use

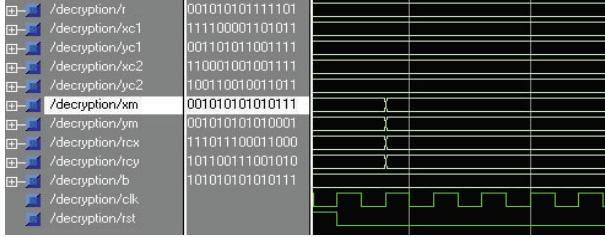


FIGURE 8: Simulation result of decryption operation.

TABLE 2: Performance comparisons of ECC ($GF(2^{193})$) with previous design.

Frequency design [MHz] (Max)	Performance [μs]
Chelton and Benaissa [10] 153.900	19.5500
Smyth et al. [13] 166.000	3720.0000
Sozzani et al. [12] 416.700	30.0000
Satoh and Takano [14] 510.200	190.0000
Sakiyama et al. [3] 555.600	12.0000
Mohamed Abdelkader 561.136	6.1799
This work (Fastest) 1930	1.625

TABLE 3: Estimation of the FSM stages and their respective execution number of cycles.

FSM steps	#stages	#execution cycles
Startup	1	1
Affine to Projective	1	1
Initial point Doubling	2	2
Counter increase	1	1
Counter compare	1	1
Montgomery point Addition	7	192
Montgomery point Doubling	7	192
Projective to affine	62	1

*The symbol # stands for “Number of”

TABLE 4: Performance comparisons.

$GF(2^m)$	Device/size	F (MHZ)/time
163-bit	XC4VLX200/24,363 Slices	143/10 μs
193-bit	XC4VLX200/6376 Slices	1930/1.625 μs
256-bit	XC4VLX200/2085 Slices	2140/1.035 μs

of two or more multipliers (tuned parallel design), the use of internal memories such as Block RAMs (optimized timing memory accesses), the speedup of the FSM, and using different ECC hardware algorithms; these optimization schemes are constrained to minimize the parallel inputs of the design and reduce routing circuitry that severely decrease efficiency, lower speed, and increase power consumption.

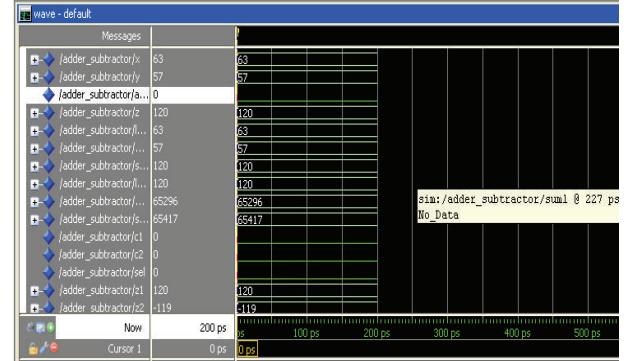


FIGURE 9: Simulation result of modular addition.

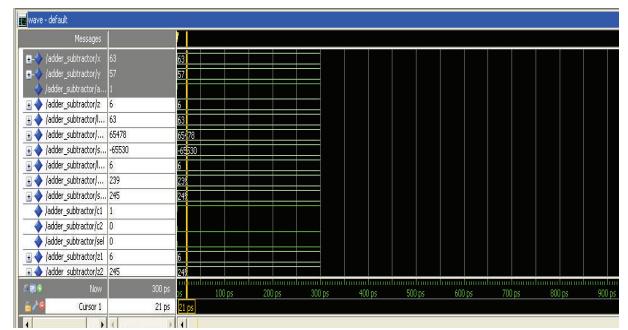


FIGURE 10: Simulation result of modular subtraction.

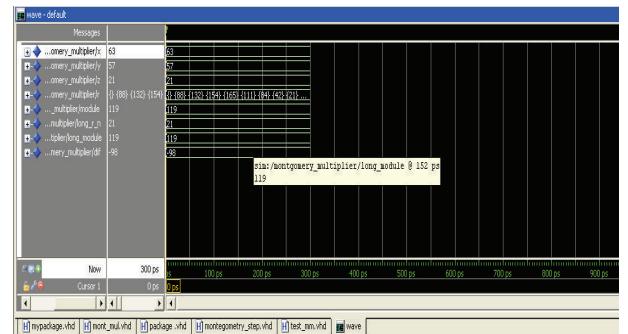


FIGURE 11: Simulation result of modular multiplication.

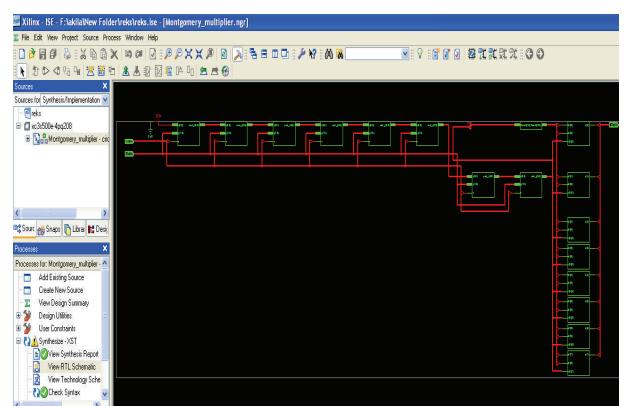


FIGURE 12: Synthesis waveform of Montgomery multiplexer.

TABLE 5: The X and Y coordinates of the result $Q = KP$.

$k = "1376F29DD55FCA07557F281055FCA07557F281D55FCA67551"$
$x_k = "0FFF0FFFFFFFFFFFF0FFF0FFF0FFFFFF00FFFFFFFFF"$
$y_k = "12A85D262AA5B53EAA7FD712AAF35F8AA9C28E2AA33558EAA"$

6. Results and Conclusion

We have presented the design of a fast version of EC crypto-hardware based on a Finite State Machine. A new ECC processor for $GF(2^{163})$ is proposed in this paper. The ECC processor consists of eight main components: host interface (HI), data memory, register file, instruction memory, control-1, control-2, AU-1, and AU-2. Secondly, $GF(2^{193})$ design introduces a better optimization at the level of multiplier and the squaring components, which utilizes the modular inverse circuit. The main characteristics of this design are concerned with the elimination of delays between the different internal components, the minimization of the global clocking resources, and a strategic separation of the data path from the control part.

We have proposed $GF(2^{193})$ result, indicating that using different optimization at the design of hardware level improves efficiency (Table 2), acceleration of the ECC scalar multiplication (615384 per seconds), and the frequency that is, the frequency of scalar multiplication, encryption, and decryption operations are 1930 MHZ and speed of operation such as key generation, encryption, and decryption are also increased. It can be used in any application where security is needed but lacks the power, storage, and computational power that is necessary for our current cryptosystem.

Thirdly, $GF(2^{256})$, an elliptic curve processor, which is used to perform the finite field operations in an elliptic curve digital signature unit. The processor is optimized for compactness on an FPGA. The implementation uses dedicated multipliers and block RAMs on the same generation on FPGAs. In this we have achieved the most compact solution. This makes our processor suitable for applications where a small overhead for security is desirable. Security issues will play an important role in the majority of communication and computer networks of the future. As the Internet becomes more and more accessible to the public, security measures have to be strengthened. Elliptic curve cryptosystems allow for shorter operand lengths than other public-key schemes based on the discrete logarithm in finite fields. We have implemented our design using Xilinx XC4VLX200 FPGA device which uses 16,209 slices and has a maximum frequency of 143 MHz. This design is roughly 4.8 times faster than two times increased hardware complexity compared with the existing methods.

References

- [1] M. A. Bencherif, H. Bessalah, and A. Guessoum, “Reconfigurable elliptic curve crypto-hardware over the galois field $GF(2^{163})$,” *American Journal of Applied Sciences*, vol. 6, no. 8, pp. 1596–1603, 2009.
- [2] T. Wollinger, J. Guajardo, and C. Paar, “Security on FPGA: state of the art implementations and attacks,” *ACM Transactions in Embedded Computing Systems*, vol. 3, pp. 53–59, 2004.
- [3] K. Sakiyama, L. Batina, B. Preneel, and I. Verbauwhede, “High-performance public-key cryptoprocessor for wireless mobile applications,” *Mobile Networks and Applications*, vol. 12, no. 4, pp. 245–258, 2007.
- [4] C. Grabbe, M. Bednara, J. Teich, J. von zur Gathen, and J. Shokrollahi, “FPGA designs of parallel high performance $GF(2^{233})$ multipliers,” in *Proceedings of the 2003 IEEE International Symposium on Circuits and Systems*, pp. 268–271, May 2003.
- [5] S. Sutikno and A. Surya, “Architecture of $F_2(2N)$ multiplier for elliptic curves cryptosystem,” in *Proceedings of the IEEE 2000 Internaitonal Symposium on Circuits and Systems*, pp. 279–282, May 2000.
- [6] G. Quan, J. P. Davis, S. Devarkal, and D. A. Buell, “High-level synthesis for large bit-width multipliers on FPGAs: a case study,” in *Proceedings of the 3rd IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and Systems Synthesis*, pp. 213–218, September 2005.
- [7] F. Rodriguez-Henriquez, N. A. Saqib, A. Diaz-perez, and C. K. Koc, *Cryptographic Algorithms on Reconfigurable Hardware*, Springer, Berlin, Germany, 2006.
- [8] Y. Dan, X. Zou, Z. Liu, Y. Han, and L. Yi, “High-performance hardware architecture of elliptic curve cryptography processor over $GF(2^{163})$,” *Journal of Zhejiang University*, vol. 10, no. 2, pp. 301–310, 2009.
- [9] W. N. Chelton and M. Benaissa, “A scalable $GF(2^m)$ arithmetic unit for application in an ECC processor,” in *Proceeding of the IEEE Workshop on Signal Processing Systems*, pp. 355–360, October 2004.
- [10] W. N. Chelton and M. Benaissa, “Fast elliptic curve cryptography on FPGA,” *IEEE Transactions on Very Large Scale Integration Systems*, vol. 16, no. 2, pp. 198–205, 2008.
- [11] Y. F. Chung, K. H. Huang, F. Lai, and T. S. Chen, “ID-based digital signature scheme on the elliptic curve cryptosystem,” *Computer Standards and Interfaces*, vol. 29, no. 6, pp. 601–604, 2007.
- [12] F. Sozzani, G. Bertoni, S. Turcato, and L. Breveglieri, “A parallelized design for an elliptic curve cryptosystem coprocessor,” in *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC '05)*, pp. 626–630, October 2005.
- [13] N. Smyth, M. McLoone, and J. V. McCanny, “An adaptable and scalable asymmetric Cryptographic processor,” in *Proceeding of the IEEE 17th International Conference on Application-specific Systems, Architectures and Processors (ASAP '06)*, pp. 341–346, September 2006.
- [14] A. Satoh and K. Takano, “A scalable dualfield elliptic curve cryptographic processor,” *IEEE Transactions on Computers*, vol. 52, no. 4, pp. 449–460, 2003.

Research Article

On the Completeness of Pruning Techniques for Planning with Conditional Effects

Dunbo Cai,^{1,2} Sheng Xu,^{1,2} Tongzhou Zhao,^{1,2} and Yanduo Zhang^{1,2}

¹ School of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan 430205, China

² Hubei Province Key Laboratory of Intelligent Robot, Wuhan Institute of Technology, Wuhan 430205, China

Correspondence should be addressed to Yanduo Zhang; zhangyanduo@hotmail.com

Received 18 May 2013; Accepted 17 July 2013

Academic Editor: William Guo

Copyright © 2013 Dunbo Cai 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.

Pruning techniques and heuristics are two keys to the heuristic search-based planning. The *helpful actions* pruning (HAP) strategy and *relaxed-plan-based heuristics* are two representatives among those methods and are still popular in the state-of-the-art planners. Here, we present new analyses on the properties of HAP. Specifically, we show new reasons for which HAP can cause incompleteness of a search procedure. We prove that, in general, HAP is incomplete for planning with conditional effects if factored expansions of actions are used. To preserve completeness, we propose a pruning strategy that is based on *relevance analysis* and *confrontation*. We will show that both *relevance analysis* and *confrontation* are necessary. We call it the *confrontation and goal relevant actions* pruning (CGRAP) strategy. However, CGRAP is computationally hard to be exactly computed. Therefore, we suggest practical approximations from the literature.

1. Introduction

The research of AI planning has advanced to a new level. Pioneers from the AI planning community have developed various practical methods that can solve much larger problems than those toy problems in early days. Two such well known methods are SAT-based planning [1], and heuristic search-based planning [2]. SAT-based planning translates planning problems into propositional satisfiability problems or more general constraint satisfaction problems (CSPs) [3]. An obvious advantage of the method is that it can exploit the power of fast algorithms from the CSP literature [4–6]. On the other hand, the “planning as search” community has been pursuing more informative heuristics to make the search fast [7, 8]. While recent studies show that planning specific heuristics can make SAT-based planning methods more competitive [9, 10], the planning as search method has shown its potential in many kinds of planning problems including classical planning [11, 12], conformant planning [13], contingent planning [14], and probabilistic planning [15]. Recent International Planning Competitions (IPCs) <http://ipc.icaps-conference.org/> have witnessed the success

of heuristic search-based planners, since the winners: Fast-Forward (FF) [11], LPG [16], SGPlan [17], and Fast Downward [12] and its successors—LAMA [18] all employ heuristic search. Two enabling techniques underlying heuristic search-based planning are *heuristic functions* and *pruning techniques*. A heuristic function measures the distances to goal of states, while pruning techniques eliminate branches that are safe to ignore. Here, we focus on the pruning techniques.

HAP is a pruning strategy developed in FF with the idea of making the search process goal directed [19]. Though, initially, it was a byproduct of the *relaxed-plan heuristic*, its notion has been popular and important in the design of top performance planners, such as Fast Downward [12]. However, the HAP strategy does not guarantee completeness; that is, it may cut branches that can reach the goal. Some of these cases were explained by Hoffmann and Nebel [11]. Nevertheless, here, we uncover a new case in which HAP can cause incompleteness. We study the conditions under which the new case will occur, the way to remedy the HAP strategy, and the cost of doing that. Based on our work, one can gain more insights into why Fast Downward, which employs the *helpful transitions* strategy, is powerful.

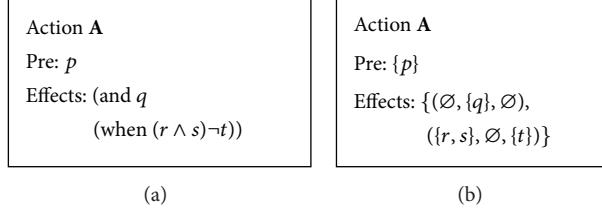


FIGURE 1: An action with conditional effects (a) and its compiling result by IPP (b).

The rest of the paper is organized as follows. In the next section, we introduce some background. Then, we show the incompleteness of HAP and extend it to a more general one called goal relevant actions pruning (GRAP), which is complete only for STRIPS planning. In Section 4, we propose our *confrontation and goal relevant actions* pruning (CGRAP), which is complete for both STRIPS planning and actions with conditional effects. In Section 5, we discuss some pruning techniques in the literature that can be seen as approximations of CGRAP. Finally, we conclude the paper and discuss some future work.

2. Background

We will first introduce notations from the state space search-based planning, then methods for handling conditional effects, and finally the heuristic function and the HAP strategy in FF.

A planning task is a quadruple $T = (\mathbf{P}, \mathbf{A}, \mathbf{I}, \mathbf{G})$, where \mathbf{P} is the set of atoms, \mathbf{A} is the set of actions, $\mathbf{I} \subseteq \mathbf{P}$ is a set of atoms called the initial state, and $\mathbf{G} \subseteq \mathbf{P}$ is the goal condition that each goal state must fulfill. States are denoted by sets of atoms. We adopted the “closed world” assumption; that is, an atom that is not in a state is false in the state. So, if $\mathbf{P} = \{p, q, r\}$ and a state $s = \{p, q\}$, then s is logically equal to $p \wedge q \wedge \neg r$. An action a is a pair $\langle \text{pre}(a), E(a) \rangle$, where $\text{pre}(a)$ is a set of atoms denoting the preconditions of a and $E(a)$ is the set of conditional effects of a . Each conditional effect $e \in E(a)$ has the form $\langle \text{con}(e), \text{add}(e), \text{del}(e) \rangle$, where $\text{con}(e)$, $\text{add}(e)$, and $\text{del}(e)$ are conditions, add effects, and delete effects of e , respectively. For an action a and a state s , if $\text{pre}(a) \subseteq s$, then we say a is applicable in s . We use $\text{App}(s) = \{a \mid \text{pre}(a) \subseteq s\}$ to denote all the actions that are applicable in s . The execution of a on s , denoted by $a(s)$, results in a state s' , where $s' = s - \bigcup_{e: \text{con}(e) \subseteq s} \text{del}(e) + \bigcup_{e: \text{con}(e) \subseteq s} \text{add}(e)$ if a is applicable in s , and $s' = s$ otherwise. A state s is called a goal state if $G \subseteq s$. A plan for a planning task is an action sequence $\pi = \langle a_0, a_1, \dots, a_{n-1} \rangle$ that transforms the initial state I into a goal state. We use $|\pi|$ to denote its length, which is the number of actions in it. Here, we assume that a plan does not have redundant actions; that is, when some action is removed from π , it will no longer be a plan.

Actions with conditional effects were introduced in the planning problem description language “Action description language” (ADL) [20]. And there are mainly three ways for handling conditional effects. Conditional effects of actions are expressed with the keyword “when”. Figure 1 (left) shows

an action with two conditional effects: the first effect q will happen with no condition and the second effect $\neg t$ will happen if $r \wedge s$ holds in the state where the action is executed. The three ways for handling conditional effects are, full expansion [21], IPP’s method [22] and factored expansion [23]. Here, we focus on IPP’s method, which is used in FF. The method will translate the action in Figure 1 (left) into the form shown in Figure 1 (right). From now on, we will call planning with actions with conditional effects ADL planning.

FF employs a forward state space search framework. Three key techniques of FF are the relaxed-plan-based heuristic (RP), HAP, and the enforced hill-climbing (EHC) algorithm. Here, we focus on RP and HAP. A relaxed plan is extracted from a relaxed version of a planning task where the delete effects of actions are ignored. Specifically, the relaxed version of a planning task $T = (\mathbf{P}, \mathbf{A}, \mathbf{I}, \mathbf{G})$ is $T' = (\mathbf{P}, \mathbf{A}', \mathbf{I}, \mathbf{G})$, where $\mathbf{A}' = \{\langle \text{pre}(a), E'(a) \rangle \mid a \in \mathbf{A}\}$ and $E'(a) = \{\langle \text{con}(e), \text{add}(e), \emptyset \rangle \mid \langle \text{con}(e), \text{add}(e), \text{del}(e) \rangle \in E(a)\}$. An action sequence $\pi = \langle a_0, a_1, \dots, a_{n-1} \rangle$ is called a *relaxed plan* for T if it is a *plan* for T' . For a state s , its heuristic value is the length of a relaxed plan for the planning task $T_s = (\mathbf{P}, \mathbf{A}, s, \mathbf{G})$. Note that the relaxed plan for T_s is not unique and FF finds one using the Graphplan [24] algorithm. During the process of computing a relaxed plan, FF keeps track of the subgoals generated in the second propositional level of a planning graph, which is saved in a set $G_1(s)$. Helpful actions are the set of actions $H(s) = \{a \mid \text{pre}(a) \subseteq s \wedge \exists e \in E(a) : (\text{con}(e) \subseteq s) \wedge (\text{add}(e) \cap G_1(s) \neq \emptyset)\}$. For a state s , the EHC algorithm only considers actions in $H(s)$ and ignores others. This search strategy is called *helpful actions* pruning. We say that a strategy is complete if using it does not make a complete algorithm eliminate search branches that are directions to goal states. As shown in [11], HAP is incomplete for STRIPS planning.

3. Complete Pruning Strategy for STRIPS

In this section, we will extend HAP to a complete strategy for STRIPS that we call *goal relevant actions* pruning (GRAP). We will then prove the completeness of GRAP for STRIPS and show that GRAP is incomplete for ADL planning. As a result, we will extend GRAP to a complete strategy for ADL planning in the next section.

3.1. Goal Relevant Actions. Helpful actions for a state s are actions in $\text{App}(s)$ that is relevant for adding the subgoals in $G_1(s)$. To obtain completeness, our goal relevant actions

for s are actions in $\text{App}(s)$ that is relevant for adding every (sub)goal generated by the GraphPlan algorithm.

Definition 1 (Dependency among Facts). For two facts $l, g \in \mathbf{P}$ and a set of actions \mathbf{A} , l is *dependent on* g with respect to \mathbf{A} (denoted as $l \triangleleft_{\mathbf{A}} g$) if

- (1) $l = g$,
- (2) $\exists a \in \mathbf{A} : \exists e \in E(a) : (l \in \text{add}(e) \wedge g \in (\text{pre}(a) \cup \text{con}(e)))$,
- (3) $\exists h \in P : (l \triangleleft_{\mathbf{A}} h \wedge h \triangleleft_{\mathbf{A}} g)$.

Definition 2 (Dependency between Facts and Actions). For an atom $l \in \mathbf{P}$ and an action $a \in \mathbf{A}$, l is *dependent on* a with respect to \mathbf{A} (denoted as $l \triangleleft_{\mathbf{A}} a$) if

- (1) $\exists a \in \mathbf{A} : \exists e \in E(a) : l \in \text{add}(e)$,
- (2) $\exists g \in P : l \triangleleft_{\mathbf{A}} g \wedge g \triangleleft_{\mathbf{A}} a$.

We note that Definitions 1 and 2 capture the relevant facts and actions to a goal. Specifically, if we are going to reach a goal g , then the actions on which g is dependent are relevant, and further, actions that adds facts on which g is dependent are also relevant. Note that in the previous definitions we use “dependent,” instead of “relevant,” to indicate a directional relation.

Now, we are ready to introduce the notion of goal relevant actions. The actions are those a search algorithm could explore for reaching some goal state. Actions that are not relevant are to be ignored.

Definition 3 (Goal Relevant Actions, GRA). Given a planning task $\mathbf{T} = (\mathbf{P}, \mathbf{A}, \mathbf{I}, \mathbf{G})$, actions that are relevant to $l \in \mathbf{P}$ are $\text{DEP}_{\triangleleft_{\mathbf{A}}}(l) = \{a \mid a \in \mathbf{A} \text{ and } l \triangleleft_{\mathbf{A}} a\}$, actions that are relevant to $\mathbf{G} \subseteq \mathbf{P}$ are $\text{DEP}_{\triangleleft_{\mathbf{A}}}(\mathbf{G}) = \bigcup_{l \in \mathbf{G}} \text{DEP}_{\triangleleft_{\mathbf{A}}}(l)$. Given a state s , the “goal relevant actions” for s is $\text{REL}_{\mathbf{G}, \triangleleft_{\mathbf{A}}}(s) = \text{DEP}_{\triangleleft_{\mathbf{A}}}(\mathbf{G}) \cap \text{App}(s)$.

We propose the following pruning strategy based on GRA. For any search algorithm and any state s , we only consider actions in $\text{REL}_{\mathbf{G}, \triangleleft_{\mathbf{A}}}(s)$ and ignore those in $\text{App}(s) - \text{REL}_{\mathbf{G}, \triangleleft_{\mathbf{A}}}(s)$. We call the strategy GRA pruning (GRAP). We will prove that GRAP is a generalization of HAP and is complete for STRIPS planning.

Proposition 4. For a planning task $\mathbf{T} = (\mathbf{P}, \mathbf{A}, \mathbf{I}, \mathbf{G})$ and any state s of T , $H(s) \subseteq \text{REL}_{\mathbf{G}, \triangleleft_{\mathbf{A}}}(s)$.

As the directional relation $\triangleleft_{\mathbf{A}}$ is transitive, the correctness of Proposition 4 is straightforward in that $G_1(s) \subseteq \text{DEP}_{\triangleleft_{\mathbf{A}}}(G)$.

Next, we will prove that GRAP is complete for STRIPS planning.

Proposition 5. GRAP is a complete pruning strategy for STRIPS planning.

Proof. Let $\mathbf{T} = (\mathbf{P}, \mathbf{A}, \mathbf{s}, \mathbf{G})$ be a STRIPS planning task, and let $\pi = \langle a_0, a_1, \dots, a_{n-1} \rangle$ be one of the plans for s . Note that π is not redundant. As we restrict to STRIPS planning, each action $a \in \mathbf{A}$ has only one conditional effect, which

is denoted by $e(a)$. We will prove that $a_0 \in \text{REL}_{\mathbf{G}, \triangleleft_{\mathbf{A}}}(s)$. We will compute $\text{DEP}_{\triangleleft_{\mathbf{A}}}(G)$ with $k = n-1, \dots, 0$. Initially, $\text{DEP}_{\triangleleft_{\mathbf{A}}}(G) = G$. For the action a_{n-1} , if its effect $e(a_{n-1})$ does not add a fact in G then π with a_{n-1} removed will still be a plan. This contradicts the assumption that π is not redundant. Following Definition 1, $\text{DEP}_{\triangleleft_{\mathbf{A}}}(G) = \text{DEP}_{\triangleleft_{\mathbf{A}}}(G) \cup \text{pre}(a_{n-1})$. Similarly, for $a_k (0 < k < n-1)$, a_k must add a fact in $\text{DEP}_{\triangleleft_{\mathbf{A}}}(G) \cup \text{pre}(a_{n-1}) \cup \dots \cup \text{pre}(a_{k+1})$; otherwise, π with a_k removed is still a plan, which contradicts the assumption that π is not redundant. Therefore, $\text{DEP}_{\triangleleft_{\mathbf{A}}}(G) = \text{DEP}_{\triangleleft_{\mathbf{A}}}(G) \cup \text{pre}(a_{n-1}) \cup \dots \cup \text{pre}(a_1)$. For a_0 , as π is not redundant, it must hold that $e(a_0) \cap (G \cup \text{pre}(a_{n-1}) \cup \dots \cup \text{pre}(a_1)) \neq \emptyset$. Following Definition 2, a_0 is in $\text{DEP}_{\triangleleft_{\mathbf{A}}}(G)$. And according to Definition 3, $a_0 \in \text{REL}_{\mathbf{G}, \triangleleft_{\mathbf{A}}}(s)$ holds. As π is an arbitrary plan, we finish the proof. \square

Note that the above proof cannot be adapted to prove the completeness of HAP, as helpful actions are defined with respect to a specific relaxed plan. In other words, the arbitrariness of plans is not guaranteed.

3.2. GRAP is Incomplete for ADL Planning. Hoffmann and Nebel [11] pointed out that HAP is incomplete as the GraphPlan algorithm is greedy in computing shorter relaxed plans. Therefore, the source of this incompleteness could be eliminated if we use other algorithms to compute relaxed plans, other than GraphPlan. The method proposed by Hoffmann and Nebel [11] works in the following way: for a state s and a relaxed planning task $\mathbf{T}' = (\mathbf{P}, \mathbf{A}', \mathbf{s}, \mathbf{G})$, they expand the planning graph to the level $|\mathbf{A}'|$ and collect subgoals backward from the level $|\mathbf{A}'|$ to level 1. Specifically, let G'_i be the subgoals at level i , then

$$G'_{|\mathbf{A}'|} = \mathbf{G},$$

$$G'_i = G'_{i+1} \cup \bigcup_{a \in \mathbf{A}_i \wedge (\exists e \in E(a) : \text{add}(e) \cap G'_{i+1} \neq \emptyset)} \text{pre}(e) \cup \text{con}(e). \quad (1)$$

Following the method, G'_1 is the union of subgoals of every relaxed plan for T' at level 1. We call actions in $\{a \mid a \in \text{App}(s), \exists e \in E(a) : \text{add}(e) \cap G'_1 \neq \emptyset\}$ “full helpful actions” (FHAs). Intuitively, an FHA is equivalent to our definition $\text{REL}_{\mathbf{G}, \triangleleft_{\mathbf{A}}}(s)$. However, the former is more procedural, and ours is more formal. We will use our definition to develop a new pruning strategy for ADL planning. Before that, we will show, by example, that both the FHA pruning (FHAP) strategy and GRAP are generally *incomplete* for ADL planning.

Example 6. Given a planning task \mathbf{T}'' , where $\mathbf{P}'' = \{p, q, r, s, m\}$, $\mathbf{I}'' = \{p, q, r\}$, $\mathbf{G}'' = \{r, m\}$, and $\mathbf{A}'' = \{a_1, a_2\}$ where a_1 is $(\{p\}, (\emptyset, \{m\}, \emptyset), (\{q\}, \{m\}, \{r\}))$ and a_2 is $(q, (\emptyset, \emptyset, \{q\}))$. The meaning of a_1 is as follows: its preconditions are $\{q\}$ and it has two conditional effects—the first is $(\emptyset, \{m\}, \emptyset)$ (denoted as e_0) and the second is $(\{q\}, \{m\}, \{r\})$ (denoted as e_1). The action a_2 has one condition q , and has a conditional effect $(\emptyset, \emptyset, \{q\})$ that falsifies q .

Next, let us consider the plan for Example 6. The difference between \mathbf{G}'' and \mathbf{I}'' is that the atom m is not *true* in \mathbf{I}'' .

To make m true, we would use action a_1 . However, executing a_1 on I'' will result in a state $s'' = \{p, q, m\}$ where atom r does not hold. After that, there is no action that can transform s'' into a goal state. One could notice that this dead end is due to the fact that e_0 and e_1 both happened and e_1 destroyed r . If we could prevent e_1 from happening, then we would succeed in finding a plan. This is the idea proposed by Weld [25], which is called “confrontation.” It is easy to see that with “confrontation” as a choice, we can find a plan $\langle a_1, a_0 \rangle$.

In Example 6, a_1 is relevant for reaching a goal state. However, pruning strategies HAP, GRAP, and FHAP all ignore it. With a generalization, we have the following results.

Proposition 7. Let $T = (\mathbf{P}, \mathbf{A}, \mathbf{s}, \mathbf{G})$ be an ADL planning task and the set of plans for T be $PLANS(T)$. If every plan $\pi \in PLANS(T)$ contains an action a of the form $(pre(a), \{(con(e_0), \emptyset, del(e_0)), \dots, (con(e_m), \emptyset, del(e_m))\})$, then HAP, FHAP, and GRAP are incomplete for T .

The correctness of Proposition 7 is in that if an action does not have any add effects, then it will not be considered as “helpful” or “relevant” anyway. As a result, this kind of actions will be mistakenly ignored.

From Example 6, we can see that actions that make “confrontations” are also “helpful” and “relevant.” Following this direction, we extend GRAP to a new pruning strategy that is complete for both STRIPS and ADL planning.

4. Complete Pruning Strategy for ADL Planning

We first introduce the notion of “confrontation and goal relevant actions” and then prove that its corresponding pruning strategy CGRAP is complete for ADL planning.

Definition 8 (Confrontational Dependency among Facts). For two atoms $l, g \in \mathbf{P}$, a set of actions \mathbf{A} , l is *confrontationally dependent on g* with respect to \mathbf{A} (denoted as $l \trianglelefteq_{\mathbf{A}} g$) if

- (1) $l = g$,
- (2) $\exists a \in \mathbf{A} : \exists e \in E(a) : l \in ((add(e) \cup del(e)) \wedge g \in (pre(a) \cup \bigcup_{e' \in E(a)} con(e')))$,
- (3) $\exists h \in \mathbf{P} : l \trianglelefteq_{\mathbf{A}} h \wedge h \trianglelefteq_{\mathbf{A}} g$.

Definition 9 (Confrontational Dependency between Facts and Actions). For two atoms $l, g \in \mathbf{P}$, an action $a \in \mathbf{A}$, l is *confrontationally dependent on a* with respect to \mathbf{A} (denoted as $l \trianglelefteq_{\mathbf{A}} a$) if

- (1) $\exists a \in \mathbf{A} : \exists e \in E(a) : l \in (add(e) \cup del(e))$,
- (2) $\exists g \in \mathbf{P} : l \trianglelefteq_{\mathbf{A}} g \wedge g \trianglelefteq_{\mathbf{A}} a$.

According to the previous two definitions, one could notice that actions that add or delete an atom l are considered as relevant to l .

Definition 10 (Confrontation and Goal Relevant Actions). Given a planning task $T = (\mathbf{P}, \mathbf{A}, \mathbf{I}, \mathbf{G})$, actions that are confrontationally relevant to $l \in \mathbf{P}$ are $DEP_{\trianglelefteq_{\mathbf{A}}}(l) = \{a \mid$

$a \in \mathbf{A} \text{ and } l \trianglelefteq_{\mathbf{A}} a\}$, and actions that are confrontationally relevant to $\mathbf{G} \subseteq \mathbf{P}$ are $DEP_{\trianglelefteq_{\mathbf{A}}}(\mathbf{G}) = \bigcup_{l \in \mathbf{G}} \{DEP_{\trianglelefteq_{\mathbf{A}}}(l)\}$. Given a state s , the “confrontation and goal relevant actions” for s is $REL_{\mathbf{G}, \trianglelefteq_{\mathbf{A}}}(s) = DEP_{\trianglelefteq_{\mathbf{A}}}(\mathbf{G}) \cap App(s)$.

The pruning strategy that considers actions in $REL_{\mathbf{G}, \trianglelefteq_{\mathbf{A}}}(s)$ only, that is, ignores $App(s) - REL_{\mathbf{G}, \trianglelefteq_{\mathbf{A}}}(s)$ is called *confrontation and goal relevant actions pruning* (CGRAP). In the following proposition, we will prove that CGRAP is complete for ADL planning.

Proposition 11. CGRAP strategy is complete for ADL planning.

Proof (Proof by Contradiction). Given a planning task $T = (\mathbf{P}, \mathbf{A}, \mathbf{I}, \mathbf{G})$ and any plan $\pi = \langle a_0, a_1, \dots, a_{n-1} \rangle$ for it, we use $s_{j+1} = a_j(s_j)$ to denote the result of executing a_j on s_j . Without loss of generality, suppose that a_i ($i = 0, \dots, n-1$) is eliminated by CGRAP; that is, $a_i \notin REL_{\mathbf{G}, \trianglelefteq_{\mathbf{A}}}(s_i)$. When $i = n-1$, as π is not a redundant plan, a_{n-1} must add one atom of \mathbf{G} ; that is, $\exists e \in E(a_{n-1}) : add(e) \cap \mathbf{G} \neq \emptyset$. Therefore, $a_i \in DEP_{\trianglelefteq_{\mathbf{A}}}(G)$. For $i \leq n-2$, a_i either (1) adds an atom $g \in \mathbf{G}$, (2) adds an atom q which is the (pre)condition of a_{i+1}, \dots, a_{n-1} , or (3) deletes one condition r of $e' \in E(a_m)$ ($m = i+1, \dots, n-1$) in order to prevent e' from happening. In case (1), $\exists g \in \mathbf{G} : g \trianglelefteq_{\mathbf{A}} a_i$, in case (2), $\exists g' \in \mathbf{G} : g' \trianglelefteq_{\mathbf{A}} q \text{ and } q \trianglelefteq_{\mathbf{A}} a_i$, and in case (3), $\exists g'' \in \mathbf{G} : g'' \trianglelefteq_{\mathbf{A}} r \text{ and } r \trianglelefteq_{\mathbf{A}} a_i$. In either cases, and according to our Definitions 8 and 9, $a_i \in DEP_{\trianglelefteq_{\mathbf{A}}}(G)$ holds. Note that $pre(a_i) \subseteq s_i$. Therefore, we conclude that $a_i \in REL_{\mathbf{G}, \trianglelefteq_{\mathbf{A}}}(s_i)$. \square

The completeness of CGRAP for ADL planning costs. One reason is that the pruning power of CGRAP is weak. In other words, CGRAP may cut a rather limited amount of branches of a search space. In addition, computing CGRAP is PSPACE-hard, as deciding irrelevant actions for a planning task is PSPACE-hard [26]. Therefore, it is practical to collect *confrontation and goal relevant actions* in an approximate way. In the next section, we will discuss some methods from the literature that fall into this scope.

5. Discussion

We will first review the *helpful transition* notion developed in Fast Downward [12] and then the *delayed partly reasoning procedure* [27].

Fast Downward is a representative planner that uses the SAS⁺ planning formalism [28], which supports multivalued variables. It translates a propositional planning problem into an SAS⁺ planning problem by utilizing an invariants analysis procedure [12]. After that, Fast Downward builds a *causal graph* that involves all the variables and a *domain transition graph* for each variable. Dependencies among variables are reasoned through the causal graph, and dependencies among values of a variable are reasoned through the corresponding domain transition graph. For details of the two kinds of graphs, please refer to [12]. The goal distance of a state s is the sum of goal distances of variables. For each variable, its goal distance is computed by solving a shortest path problem

formulated on the corresponding domain transition graph. In the problem, the source node is the value the variable currently takes, and the target node is the value that goal conditions require. When such a path is obtained, the transition associated with the first edge is labeled as “helpful transition.” Note that transitions are conditional effects of actions. So, we can collect “helpful actions” based on “helpful transitions.” Here we note that “helpful transitions” consider both goal relevant actions and actions that are for confrontations. The ability of collecting actions that are helpful for confrontations originates from the multivalued variable representation. In the representation, the change from one value to another one models both the *add and delete effects* of an action on a variable. As a result, actions have only one kind of effects, which are considered by Fast Downward to collect “helpful transitions.” In contrast, propositional planners, such as FF, consider only the *add effects* of actions for collecting “helpful actions.” Therefore, the “helpful transitions” strategy can be considered as an approximation of CGRAP.

The “delayed partly reasoning procedure” is proposed by Cai et al. [7]. This procedure is implemented on top of the propositional planning formalism. In the first action level of a planning graph, the procedure tracks harmful inducements with respect to an order of conditional effects and collects actions that confront the inducements. An inducement is that one conditional effect e induces another conditional effect e' . It is harmful if e' deletes some previously added atoms of other conditional effects. As the procedure only operates on the first actions level and works with a predefined order, it is an approximation of CGRAP. Therefore, the computational cost of the procedure is not high.

6. Conclusions and Future Work

In this work, we analyzed some well-known pruning techniques, which are currently utilized by state-of-the-art planners. In particular, we showed that the *helpful actions* pruning strategy is incomplete for ADL planning and extended it to a complete strategy called *confrontation and goal relevant actions* pruning. Though our proposed strategy is computationally hard, we discussed methods from the AI planning literature that can be seen as approximations of it. In addition, we believe that this work will help us gain more insights into why the planner Fast Downward is powerful.

This work was done on pruning techniques in search-based planning. Future directions may consider pruning techniques in SAT-based ADL planning and conformant planning. As IPP’s method for handling conditional effects does not lead to a high increase in problem size, it is suitable for SAT-based planning. Therefore, developing adaptations or approximations of our proposed strategy CGRAP in that settings could be interesting.

Acknowledgments

This work is supported by Natural Science Foundation of China (Grant no. 61103136), Educational Commission of Hubei Province of China (Grant no. D20111507), Hubei Province Key Laboratory of Intelligent Robot Open

Foundation (Grant no. HBIR200909), and Youths Science Foundation of Wuhan Institute of Technology (Grant no. 12106022).

References

- [1] H. Kautz and B. Selman, “Pushing the envelope: planning, propositional logic, and stochastic search,” in *Proceedings of the 13th National Conference on Artificial Intelligence*, pp. 1194–1201, August 1996.
- [2] B. Bonet and H. Geffner, “Planning as heuristic search,” *Artificial Intelligence*, vol. 129, no. 1-2, pp. 5–33, 2001, Heuristic search in artificial intelligence.
- [3] M. B. Do and S. Kambhampati, “Planning as constraint satisfaction: solving the planning graph by compiling it into CSP,” *Artificial Intelligence*, vol. 132, no. 2, pp. 151–182, 2001.
- [4] N. Eén and A. Biere, “Effective preprocessing in SAT through variable and clause elimination,” in *Theory and Applications of Satisfiability Testing*, vol. 3569, pp. 61–75, Springer, Berlin, Germany, 2005.
- [5] X. Li and M. Yin, “An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure,” *Advances in Engineering Software*, vol. 55, pp. 10–31, 2013.
- [6] X. Li and M. Yin, “A discrete artificial bee colony algorithm with composite mutation strategies for permutation flow shop scheduling problem,” *Scientia Iranica*, vol. 19, no. 6, pp. 1921–1935, 2012.
- [7] D. Cai, J. Sun, and M. Yin, “Making FF faster in ADL domains,” in *Proceedings of the 3rd International Conference on Natural Computation (ICNC ’07)*, vol. 2, pp. 160–164, August 2007.
- [8] M. Katz, J. Hoffmann, and C. Domshlak, “Who said we need to relax all variables?” in *Proceedings of the 23rd International Conference on Automated Planning and Scheduling (ICAPS ’13)*, AAAI Press, 2013.
- [9] P. H. Tu, T. C. Son, M. Gelfond, and A. R. Morales, “Approximation of action theories and its application to conformant planning,” *Artificial Intelligence*, vol. 175, no. 1, pp. 79–119, 2011.
- [10] J. Rintanen, “Planning as satisfiability: heuristics,” *Artificial Intelligence*, vol. 193, pp. 45–86, 2012.
- [11] J. Hoffmann and B. Nebel, “The FF planning system: fast plan generation through heuristic search,” *Journal of Artificial Intelligence Research*, vol. 14, pp. 253–302, 2001.
- [12] M. Helmert, “The fast downward planning system,” *Journal of Artificial Intelligence Research*, vol. 26, pp. 191–246, 2006.
- [13] J. Hoffmann and R. I. Brafman, “Conformant planning via heuristic forward search: a new approach,” *Artificial Intelligence*, vol. 170, no. 6–7, pp. 507–541, 2006.
- [14] J. Hoffmann and R. Brafman, “Contingent planning via heuristic forward search with implicit belief states,” in *Proceedings of the 15th International Conference on Automated Planning and Scheduling (ICAPS’05)*, vol. 2005, 2005.
- [15] B. Bonet and E. Hansen, “Heuristic Search for Planning under Uncertainty,” in *Heuristics, Probability and Causality: A Tribute To Judea Pearl*, pp. 3–22, College Publications, 2010.
- [16] A. Gerevini, I. Serina, A. Saetti, and S. Spinoni, “Local search techniques for temporal planning in lpg,” in *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS ’03)*, pp. 62–72, 2003.

- [17] Y. Chen, B. W. Wah, and C.-W. Hsu, “Temporal planning using subgoal partitioning and resolution in SGPlan,” *Journal of Artificial Intelligence Research*, vol. 26, pp. 323–369, 2006.
- [18] S. Richter, M. Helmert, and M. Westphal, “Landmarks revisited,” in *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pp. 975–982, July 2008.
- [19] D. McDermott, “Using regression-match graphs to control search in planning,” *Artificial Intelligence*, vol. 109, no. 1, pp. 111–159, 1999.
- [20] E. P. D. Pednault, “ADL: exploring the middle ground between STRIPS and the situation calculus,” in *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning*, pp. 324–332, 1989.
- [21] B. C. Gazen and C. A. Knoblock, “Combining the expressivity of ucpop with the efficiency of graphplan,” in *Proceedings of the 4th European Conference on Planning (ECP '97)*, pp. 221–233, 1997.
- [22] J. Koehler, B. Nebel, J. Hoffmann, and Y. Dimopoulos, “Extending planning graphs to an adl subset,” in *Proceedings of the 4th European Conference on Planning: Recent Advances in AI Planning (ECP '97)*, pp. 273–285, 1997.
- [23] C. R. Anderson, D. E. Smith, and D. S. Weld, “Conditional effects in graphplan,” in *Proceedings of the AIPS*, pp. 44–53, 1998.
- [24] A. L. Blum and M. L. Furst, “Fast planning through planning graph analysis,” *Artificial Intelligence*, vol. 90, no. 1-2, pp. 281–300, 1997.
- [25] D. S. Weld, “Introduction to least commitment planning,” *AI Magazine*, vol. 15, no. 4, pp. 27–61, 1994.
- [26] B. Nebel, Y. Dimopoulos, and J. Koehler, “Ignoring irrelevant facts and operators in plan generation,” in *Proceedings of the European Conference on Planning (ECP '97)*, pp. 338–350, 1997.
- [27] D. Cai, J. Hoffmann, and M. Helmert, “Enhancing the context-enhanced additive heuristic with precedence constraints,” in *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS '09)*, pp. 50–57, grc, September 2009.
- [28] C. Bäckström and B. Nebel, “Complexity results for SAS+ planning,” *Computational Intelligence*, vol. 11, no. 4, pp. 625–655, 1995.