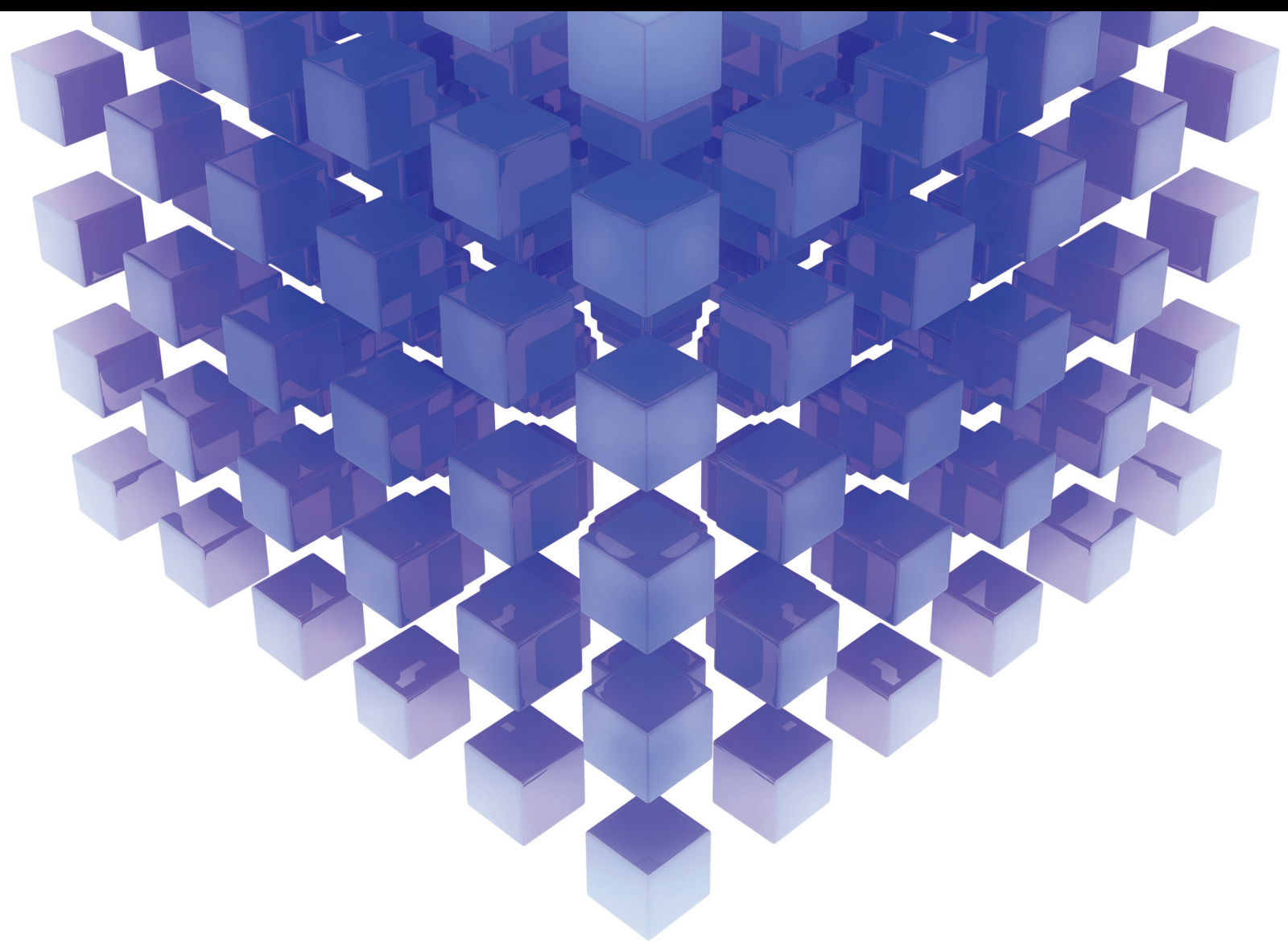
Mathematical Problems in Engineering

Evolutionary Algorithms and Metaheuristics: Applications in Engineering Design and Optimization

Guest Editors: David Greiner, Jacques Periaux, Domenico Quagliarella, Jorge Magalhaes-Mendes, and Blas Galván



Evolutionary Algorithms and Metaheuristics: Applications in Engineering Design and Optimization

Mathematical Problems in Engineering

Evolutionary Algorithms and Metaheuristics: Applications in Engineering Design and Optimization

Guest Editors: David Greiner , Jacques Periaux, Domenico Quagliarella, Jorge Magalhaes-Mendes, and Blas Galván

Editorial Board

Contents

Evolutionary Algorithms and Metaheuristics: Applications in Engineering Design and Optimization
David Greiner, Jacques Periaux, Domenico Quagliarella, Jorge Magalhaes-Mendes, and Blas Galván
Volume 2018, Article ID 2793762, 4 pages

MOQPSO-D/S for Air and Missile Defense WTA Problem under Uncertainty
Hao Xu, Qinghua Xing, and Zhenhao Tian
Volume 2017, Article ID 9897153, 13 pages

Optimal Routing and Scheduling of Charge for Electric Vehicles: A Case Study
J. Barco, A. Guerra, L. Muñoz, and N. Quijano
Volume 2017, Article ID 8509783, 16 pages

Mode-Based versus Activity-Based Search for a Nonredundant Resolution of the Multimode Resource-Constrained Project Scheduling Problem
Daniel Morillo, Federico Barber, and Miguel A. Salido
Volume 2017, Article ID 4627856, 15 pages

Bee-Inspired Algorithms Applied to Vehicle Routing Problems: A Survey and a Proposal
Thiago A. S. Masutti and Leandro N. de Castro
Volume 2017, Article ID 3046830, 20 pages

Memetic Computing Applied to the Design of Composite Materials and Structures
Jose Ignacio Pelaez, Jose Antonio Gomez-Ruiz, Jorge Veintimilla, Gustavo Vaccaro, and Patricia Witt
Volume 2017, Article ID 4723863, 16 pages

A GODFIP Control Algorithm for an IRC Grain Dryer
Aini Dai, Xiaoguang Zhou, and Xiangdong Liu
Volume 2017, Article ID 1406292, 14 pages

A Parallel Biased Random-Key Genetic Algorithm with Multiple Populations Applied to Irregular Strip Packing Problems
Bonfim Amaro Júnior, Plácido Rogério Pinheiro, and Pedro Veras Coelho
Volume 2017, Article ID 1670709, 11 pages

Research on Optimized Torque-Distribution Control Method for Front/Rear Axle Electric Wheel Loader
Zhiyu Yang, Jixin Wang, Guangzong Gao, and Xiangyun Shi
Volume 2017, Article ID 7076583, 12 pages

A Simulated Annealing Approach for the Train Design Optimization Problem
Federico Alonso-Pecina and David Romero
Volume 2017, Article ID 4703106, 11 pages

Reconfiguration of Distribution Networks with Distributed Generation Using a Dual Hybrid Particle Swarm Optimization Algorithm
Caoyuan Ma, Chunxiao Li, Xuezi Zhang, Guoxin Li, and Yonggang Han
Volume 2017, Article ID 1517435, 10 pages

A Developed Optimization Method of Tight Reservoir Porosity
Fenqiang Wu and Jingling Xu
Volume 2017, Article ID 2048475, 12 pages

Application of Heuristic and Metaheuristic Algorithms in Solving Constrained Weber Problem with Feasible Region Bounded by Arcs
Igor Stojanović, Ivona Brajević, Predrag S. Stanimirović, Lev A. Kazakovtsev, and Zoran Zdravev
Volume 2017, Article ID 8306732, 13 pages

A Hybrid Method for Truss Mass Minimization considering Uncertainties
Herbert M. Gomes and Leandro L. Corso
Volume 2017, Article ID 2324316, 14 pages

Solving a Two-Stage Stochastic Capacitated Location-Allocation Problem with an Improved PSO in Emergency Logistics
Ye Deng, Wanhong Zhu, Jian Tang, and Jianfei Qin
Volume 2017, Article ID 6710929, 15 pages

An Improved Particle Swarm Optimization Algorithm Using Eagle Strategy for Power Loss Minimization
Hamza Yapıcı and Nurettin Çetinkaya
Volume 2017, Article ID 1063045, 11 pages

Optimizing Production Scheduling of Steel Plate Hot Rolling for Economic Load Dispatch under Time-of-Use Electricity Pricing
Mao Tan, Hua-li Yang, Bin Duan, Yong-xin Su, and Feng He
Volume 2017, Article ID 1048081, 13 pages

Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 2793762, 4 pages
https://doi.org/10.1155/2018/2793762

Editorial

Evolutionary Algorithms and Metaheuristics: Applications in Engineering Design and Optimization

David Greiner,[1] **Jacques Periaux**,[2] **Domenico Quagliarella**,[3] **Jorge Magalhaes-Mendes**,[4] **and Blas Galván**[1]

[1] *Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (SIANI), Universidad de Las Palmas de Gran Canaria, 35017 Las Palmas de Gran Canaria, Spain*
[2] *University of Jyväskylä, Jyväskylä, Finland*
[3] *Italian Aerospace Research Center (CIRA), Capua, Italy*
[4] *Politecnico do Porto, Porto, Portugal*

Correspondence should be addressed to David Greiner; david.greiner@ulpgc.es

Received 14 December 2017; Accepted 17 December 2017; Published 17 January 2018

1. Introduction

Evolutionary algorithms and, more generally, nature-inspired metaheuristics are gaining increasing favor as computational intelligence methods, very useful for global optimization problems. The success of these population-based frameworks is mainly due to their flexibility and ease of adaptation to the most different and complex optimization problems, without requiring any special feature or condition to the objective functions and related constraints, like continuity, derivability, or convexity. Discrete and combinatorial optimization problems, as well as mixed ones, are not a limit for this class of optimizers. Moreover, the requirement of uncertainty quantification in the search process, like in reliability-based optimization and robust design, is not a limit for this approach. Finally, population-based optimization algorithms can deal naturally with multiobjective problems, and this has made a big leap forward in the ability to effectively handle this class of problems possible. These advantages, together with the steady improvement of computer performance, are fostering their increased use in research and industry in a wide variety of engineering branches.

These methodologies are empowering the enhancement in engineering design and optimization practices in areas in which classical optimization techniques are still not able effective. Indeed the aforementioned requisites and limitations are usual, such as the nondifferentiability of the modeling, in real world engineering problems. For example, this is the case of automotive industry, aeronautical and aerospace industry, and civil, structural, and mechanical engineering, where the calculation of the objective function values requires the resolution of numerical models, using (nonlinear) partial differential equations, based on finite elements, boundary elements, finite volumes, and so on. As stated in [1], the origins of Evolution Strategies [2] during the middle sixties in University of Berlin (Germany) were ignited by the necessity of solving an "optimal shape of bodies in a flow" problem during wing tunnel experiments in the Institute of Flow Engineering, after unsuccessful attempts with the coordinate and simple gradient strategies. Early applications of evolutionary algorithms dealing with engineering design and optimization date from the late eighties [3, 4] and early nineties as in [5, 6]. There have been applications compiled in book volumes as in [7–10], and the field has been continuously growing, as in the case of evolutionary multiobjective applications where a state-of-the-art review can be found in [11], or [12, 13]. Recent volumes of scientific contributions in the field are covered by [14–16].

The advances in the use of evolutionary algorithms and nature-inspired metaheuristics in engineering applications bring an opportunity and also a challenge for researchers to improve and advance in design and optimization of products, systems, and services for societal benefit. The purpose of this special issue is to publish high-quality research or

review articles that address recent development from a variety of engineering fields in relation to the application of evolutionary algorithms and metaheuristics for design and optimization and that, hopefully, will stimulate other researchers to continue the efforts to improve the current state of the art on the aforementioned field.

2. Scientific Contributions of the Special Issue

In this special issue, a reviewing process has been performed where at least two reviewers per paper have been assigned, where a 15% acceptation rate has been held.

The accepted papers can be classified according to the following engineering/application categories: (a) energy and electrical engineering; (b) structural and civil engineering; (c) scheduling transport and combinatorial optimization; (d) control; (e) other applications/military.

A brief description of each contribution published in the special issue is given in the following paragraphs according to the previous classification.

2.1. Energy and Electrical Engineering. A particle swarm optimization algorithm using the eagle strategy (ESPSO), a method of combination of global search and intensive local search, is introduced for solving the reactive power losses minimization problem, by H. Yapıcı and N. Cetinkaya. Experiments cover the IEEE 30-bus and IEEE 118-bus power systems and a real power distribution subsystem. A comparison with other metaheuristics is provided.

The reconfiguration of smart grid with distributed generation is studied by C. Ma et al., using a dual hybrid particle swarm optimization (an improved binary particle swarm optimization algorithm was used in branch group search, and the proposed group binary particle swarm optimization search algorithm was used for searching within the group). From the simulations on the IEEE 33-bus distribution power system, after the reconfiguration of the distributed power grid, the loss of the distribution network is reduced, and the quality of the power supply voltage and the power quality of the grid are improved.

M. Tan et al. introduce a multiobjective optimization model of Hot Rolling Production Scheduling Problem under Time-of-Use electricity pricing, for simultaneous minimization of electricity costs in production and minimizing the total penalties caused by jumps between adjacent slabs. A nondominated sorting genetic algorithm-II (NSGA-II) based production scheduling was performed to obtain nondominated solutions, and TOPSIS decision-making method was used for final solution selection. Experiments confirm the success of the approach.

2.2. Structural and Civil Engineering. J. I. Pelaez et al. present a memetic algorithm for the design of Symmetric Laminated Composites and Structures, taking into account in the fitness function economic and safety criteria in design and implementing a set of local search operators. It is compared with other four metaheuristics. The model has been tested with the design of a plate under distributed Nx and Ny loading and compared with two literature models, being optimum designs validated with the ANSYS software package.

F. Wu and J. Xu present an optimization method to evaluate the porosity of tight reservoirs by the use of a modified multicomponent model to a mixed-matrix model and a simulated annealing algorithm. The method is validated with a set of data from tight reservoirs.

A hybrid reliability-based design optimization (RBDO) algorithm is proposed by H. M. Gomes and L. L. Corso, which combines characteristics of genetic algorithms and particle swarm optimization and sequential quadratic programming for local search. The hybrid method is analyzed based on three structural trusses RBDO benchmark examples for sizing optimization with stress, displacements, and frequency constraints.

2.3. Scheduling, Transport, and Combinatorial Optimization. A two-optimization phase based genetic algorithm (GA) is proposed by D. Morillo et al. for solving an energy-based extension of the Multimode Resource-Constrained Project Scheduling Problem, where the search is focused on Mode Lists instead of doing it on Activity Lists. Five GA variants were compared, where the proposed algorithm outperforms the others in the set of problems of the project scheduling problem library PSP-LIB.

A two-stage stochastic capacitated location-allocation problem in emergency logistics is considered by Y. Deng et al., where the number and capacities of supply centers are uncertain and had to be determined. To solve this problem, a two-stage expected value model and a generalized cost function are proposed. An improved particle swarm optimizer with Gaussian cloud operator, restart strategy, and adaptive parameter strategy is used, as well as using the interior point method instead of the simplex method in the second stage. The proposed methods improve precision and convergence rates when compared with the classic one-stage expected value model.

T. A. S. Masutti and L. N. de Castro present a thorough review of bee-inspired methods designed to solve the vehicle routing problems. A taxonomy of methods was detailed and the review followed considering problems solved and modifications introduced in the bee-inspired algorithms. Additionally, the TSPoptBees algorithm, a modification of the original optBees purposely focused to solve the traveling salesman problem (TSP), is compared with other optimization methods inspired by the behavior of bees to solve a set of 28 instances of the TSPLIB with competitive results.

Differential Evolution is compared with genetic algorithms to solve the Electric Vehicle Routing Problem, by J. Barco et al. The problem is based on a scheme to coordinate the battery electric vehicles' (BEV) routing and recharge scheduling, considering operation and battery degradation costs. The model is based on the longitudinal dynamics equation of motion estimating the energy consumption of each BEV, where a case study, airport shuttle service scenario, is solved.

The irregular strip packing problem, present in many production processes in factories, with a rectangular stage, a fixed width, and an unlimited length, is solved in the work

proposed by B. A. Júnior et al., combining a collision-free region placement procedure with a parallel Biased Random-Key Genetic Algorithm with multiple subpopulations, where the objective is to minimize the required area to allocate the demand. The approach is tested in a set of EURO Special Interest Group on Cutting and Packing (ESICUP) problems and compared with other six optimization algorithms.

F. Alonso-Pecina and D. Romero propose a two-step method to solve the Train Design Optimization Problem, where the first step aims to produce an initial feasible solution and the second uses simulated annealing to improve the initial solution, followed by procedures that attempt to decrease the number of required trains without incrementing the overall cost. Experiments cover well-known instances improving other optimization methods.

I. Stojanović et al. solve the constrained nonconvex optimization Weber problem with feasible region bounded by arcs, with four swarm-intelligence techniques: the artificial bee colony (ABC) for constrained optimization, the crossover-based ABC algorithm, the firefly algorithm for constrained optimization, and the enhanced firefly algorithm; also a heuristic algorithm based on the modified Weiszfeld procedure is used. The crossover-based ABC outperforms the other metaheuristics (and also the heuristic algorithm) with respect to the quality of the results, robustness, and computational efficiency, in the experiments published in this work.

2.4. Control. The optimized torque-distribution control method is a critical technology for front/rear axle electric wheel loader (FREWL) to improve the operation performance and energy efficiency. A weighted sum approach for minimization of mean and variance of tire workload and maximization of total motor efficiency on a longitudinal dynamics model of FREWL is proposed by Z. Yang et al. The following optimization algorithms are used to solve the problem: quasi-newton Lagrangian multiplier method, sequential quadratic programming, adaptive genetic algorithms, and particle swarm optimization with random weighting and natural selection. Results confirm advantages of the controlled FREWL over noncontrolled FREWL.

A genetic optimization dual fuzzy immune Proportional-Integral-Derivative (GODFIP) controller is proposed by A. Dai et al., considering energy savings, stability, accuracy, and rapidity. Its structure consists of two fuzzy controllers, a PID controller, an immune algorithm, and a genetic optimization algorithm. It is designed and simulated to control an infrared radiation and convection grain dryer represented by an identified autoregressive with exogenous input (NARX) model, improving the control performance of a fuzzy immune PID controller.

2.5. Other Applications/Military. The multiobjective weapon target assignment (WTA) problem under uncertainty, whose goals are to obtain maximum interception efficiency and minimum interception consumption, is optimized by H. Xu et al., with a multiobjective quantum-behaved particle swarm optimization with double/single well (MOQPSO-D/S), and compared with other PSO variants.

Acknowledgments

The invited editors of the special issue gratefully thank all the reviewers that have contributed to the process of developing this special issue. We hope that the works selected for publication will inspire the engineering design and optimization community to apply new and state-of-the-art metaheuristics/evolutionary algorithms to their challenging applicative and industrial problems.

David Greiner
Jacques Periaux
Domenico Quagliarella
Jorge Magalhaes-Mendes
Blas Galván

References

[1] H. Schwefel, *Numerische optimierung von computer-modellen mittels der evolutionsstrategie*, Birkhäuser, Basel and Stuttgart, 1977.

[2] I. Rechenberg, *Evolutionstrategie—optimierung technischer systeme nach prinzipien der biologischen evolution*, Fromman-Holzboog, 1973.

[3] D. E. Goldberg and M. P. Samtani, "Engineering optimization via genetic algorithm," in *Proceedings of the 9th Conference on Electronic Computation ASCE*, pp. 471–482, New York, NY, USA, 1986.

[4] D. E. Goldberg, *Genetic Algorithms for Search, Optimisation, and Machine Learning*, vol. 27, Addison-Wesley, Reading, 1989.

[5] P. Hajela, "Genetic search - An approach to the nonconvex optimization problem," *AIAA Journal*, vol. 28, no. 7, pp. 1205–1210, 1990.

[6] K. Deb, "Optimal design of a welded beam via genetic algorithms," *AIAA Journal*, vol. 29, no. 11, pp. 2013–2015, 1991.

[7] G. Winter, J. Periaux, M. Galan, and P. Cuesta, *Genetic Algorithms in Engineering and Computer Science*, John Wiley & Sons, 1996.

[8] D. Quagliarella, J. Periaux, C. Poloni, and G. Winter, *Genetic Algorithms and Evolution Strategy in Engineering and Computer Science: Recent Advances and Industrial Applications*, John Wiley & Son Ltd., 1998.

[9] K. Giannakoglou, D. T. Tsahalis, J. Periaux, and T. Fogarty, *Evolutionary Methods for Design, Optimization and Control*, CIMNE, 2002.

[10] G. Bugeda, J. A. Desideri, J. Periaux, M. Schoenauer, and G. Winter, "Evolutionary methods for design, optimization and control. Applications to industrial and societal problems," CIMNE, 2003.

[11] C. Coello Coello, G. Lamont, and D. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, vol. 5, Springer, 2007.

[12] P. Neittaanmäki, J. Périaux, and T. Tuovinen, "Evolutionary and deterministic methods for design, optimization and control. Applications to industrial and societal problems," CIMNE, 2008.

[13] T. Burczynski and J. Périaux, *Evolutionary and Deterministic Methods for Design, Optimization And Control*, CIMNE, 2011.

[14] A. E. Eiben and J. Smith, "From evolutionary computation to the evolution of things," *Nature*, vol. 521, no. 7553, pp. 476–482, 2015.

[15] D. Greiner, B. Galvan, J. Periaux, N. Gauger, K. Giannakoglou, and G. Winter, "Advances in evolutionary and deterministic methods for design, optimization and control in engineering and sciences," in *Computational Methods in Applied Sciences*, vol. 36, Springer, New York, NY, USA, 2015.

[16] J. Magalhaes-Mendes and D. Greiner, "Evolutionary algorithms and metaheuristics in civil engineering and construction management," in *Computational Methods in Applied Sciences*, vol. 39, Springer, New York, NY, USA, 2015.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 9897153, 13 pages
https://doi.org/10.1155/2017/9897153

Research Article

MOQPSO-D/S for Air and Missile Defense WTA Problem under Uncertainty

Hao Xu, Qinghua Xing, and Zhenhao Tian

School of Air and Missile Defense, Air Force Engineering University, Xi'an 710051, China

Correspondence should be addressed to Hao Xu; kgdxuhao@163.com

Received 16 December 2016; Revised 5 November 2017; Accepted 12 November 2017; Published 14 December 2017

Academic Editor: Erik Cuevas

Aiming at the shortcomings of single objective optimization for solving weapon target assignment (WTA) and the existing multiobjective optimization based WTA method having problems being applied in air and missile defense combat under uncertainty, a fuzzy multiobjective programming based WTA method was proposed to enhance the adaptability of WTA decision to the changes of battlefield situation. Firstly, a multiobjective quantum-behaved particle swarm optimization with double/single-well (MOQPSO-D/S) algorithm was proposed by adopting the double/single-well based position update method, the hybrid random mutation method, and the two-stage based guider particles selection method. Secondly, a fuzzy multiobjective programming WTA model was constructed with consideration of air and missile defense combat's characteristics. And, the uncertain WTA model was equivalently clarified based on the necessity degree principle of uncertainty theory. Thirdly, with particles encoding and illegal particles adjusting, the MOQPSO-D/S algorithm was adopted to solve the fuzzy multiobjective programming based WTA model. Finally, example simulation was conducted, and the result shows that the WTA model constructed is rational and MOQPSO-D/S algorithm is efficient.

1. Introduction

In information warfare, air attack is a highly integrated operation form. Defenders need to carry out the task of anti-aerodynamic targets and antiballistic missiles simultaneously [1]. As one of the key phases in air and missile defense combat, weapon target assignment (WTA) plays a significant role on operational effectiveness. Therefore, study on air and missile defense WTA problem is of great significance.

Given the importance, WTA has been studied with application of Lagrange relaxation algorithm [2], ant colony algorithm [3], genetic algorithm [4], clone selection algorithm [5], and particle swarm optimization algorithm [6]. Most of these researches construct WTA model based on single objective optimization with the objective function of maximizing the kill probability to enemy. They need to allocate all the firepower without consideration of the sequent operation need and cannot fit with the operational practice. In order to avoid allocating all firepower, a WTA model was constructed with the objective function of maximizing effectiveness-cost ratio in literature [6]. However, the adaptability of WTA decision, based on this method, to the changes of battlefield situation is still weak. Because this method is still a single objective optimization method, when conducting this method, the decision-making preferences need to be provided beforehand and only one WTA alternative can be given. Aiming at overcoming these weak points, the multiobjective optimization method was introduced into studying WTA problem [7–9]. With consideration of every objective, the Pareto solution set, obtained by multiobjective optimization method, can provide a series of WTA alternatives for the commander. And the final WTA decision will be made by choosing one from the alternatives. These studies [7–9] could provide useful reference for research on air and missile defense WTA problem. However, they cannot be applied to air and missile defense combat, which has the characteristic of multiweapon composite disposition, directly. Moreover, with the growing complexity of battlefield, the certain WTA method [3–9] can hardly be applied to air and missile defense combat under uncertainty.

In order to address the issues above, the WTA model for air and missile defense with uncertain threat information is constructed based on fuzzy multiobjective programming, with consideration of the air and missile defense combat's characteristics. Due to the superior performance of quantum-behaved particle swarm optimization algorithm, a multiobjective quantum-behaved particle swarm optimization with double/single-well algorithm is proposed for solving the WTA model.

2. Basic Theory

2.1. Concepts of Multiobjective Optimization

Definition 1 (multiobjective optimization problem). Taking the minimizing problem as an example, a multiobjective optimization problem can be defined as follows:

$$\begin{aligned} \min \quad & f(\mathbf{x}) = \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_o(\mathbf{x})\right) \\ \text{s.t.} \quad & \mathbf{x} \in \Omega. \end{aligned} \tag{1}$$

In (1), $f_l(\mathbf{x})\,(l = 1, 2, \ldots, o)$ is objective function, $\mathbf{x}$ is the decision vector, and Ω is the decision space defined by a series of constraints.

Definition 2 (Pareto dominance). Let $\mathbf{x}_a$ and $\mathbf{x}_b$ be two feasible solutions in Ω; if

$$\begin{aligned} &\text{all}\left(f_l(\mathbf{x}_a) \le f_l(\mathbf{x}_b)\right) \,\&\&\, \text{any}\left(f_l(\mathbf{x}_a) < f_l(\mathbf{x}_b)\right) \\ &\qquad\qquad (l = 1, 2, \ldots, o), \end{aligned} \tag{2}$$

then $\mathbf{x}_a$ Pareto dominates $\mathbf{x}_b$, denoted by $\mathbf{x}_a \prec \mathbf{x}_b$.

Definition 3 (Pareto optimal solution). Let $\mathbf{x}_a \in \Omega$; if $\neg\exists \mathbf{x}_b \in \Omega$, meeting $\mathbf{x}_b \prec \mathbf{x}_a$, then $\mathbf{x}_a$ would be defined as a Pareto optimal solution. And the set $\Omega_p = \{\mathbf{x}_a \in \Omega \mid \neg\exists \mathbf{x}_b \in \Omega, \mathbf{x}_b \prec \mathbf{x}_a\}$ would be defined as Pareto optimal solution set.

Definition 4 (Pareto front). The region which consisted of the objective values, corresponding to all solutions in Ω_p, is defined as Pareto front.

2.2. QPSO Algorithm. Quantum-behaved particle swarm optimization (QPSO) algorithm [10, 11], compared with the particle swarm optimization algorithm, has many advantages, such as faster convergence rate, fewer control parameters, and better global convergence. It has attracted much attention of scholars [12, 13].

QPSO algorithm is a quantum mechanics based optimization algorithm. It is assumed that the motion state of the particles in the optimal space relative to the attractor $\mathbf{P}$ can be described by the wave function $\psi(\mathbf{Y})$ in δ potential well. And $\psi(\mathbf{Y})$ is described as follows:

$$\psi(\mathbf{Y}) = \frac{1}{\sqrt{\mathbf{L}}} e^{-|\mathbf{Y}|/\mathbf{L}}. \tag{3}$$

In (3), $\mathbf{L}$ is the characteristic length of δ potential well.

The corresponding probability density function is $Q(\mathbf{Y}) = (1/\mathbf{L})e^{-2|\mathbf{Y}|/\mathbf{L}}$, and the probability distribution function is $F(\mathbf{Y}) = e^{-2|\mathbf{Y}|/\mathbf{L}}$.

Suppose $u \sim U(0, 1)$, and let $u = F(\mathbf{Y})$; the position of the particle relative to the attractor $\mathbf{P}$ can be obtained as $\mathbf{Y} = \pm(\mathbf{L}/2)\ln(1/u)$.

And the absolute position of the particle in the optimal space is $\mathbf{x} = \mathbf{P} + \mathbf{Y}$. Therefore, the position update formula of particle $\mathbf{x}_k$ can be obtained as follows:

$$\mathbf{x}_k(t+1) = \mathbf{P}_k(t) \pm \frac{\mathbf{L}_k(t)}{2} \ln\left(\frac{1}{u_k(t)}\right). \tag{4}$$

In (4), t is the current generation, $u_k(t) \sim U(0, 1)$. If $u_k(t) > 0.5$, "±" would be chosen as "+", or it would be chosen as "−". $\mathbf{P}_k(t)$ and $\mathbf{L}_k(t)$ are as follows:

$$\mathbf{P}_k(t) = \varphi_k(t)\,\mathbf{pbest}_k(t) + \left(1 - \varphi_k(t)\right)\mathbf{gbest}(t), \tag{5}$$

$$\mathbf{L}_k(t) = 2\alpha\left|\mathbf{P}_k(t) - \mathbf{x}_k(t)\right|. \tag{6}$$

In (5), $\varphi_k(t) \sim U(0, 1)$, $\mathbf{pbest}_k(t)$ is the individual best position of particle $\mathbf{x}_k$, and $\mathbf{gbest}(t)$ is the global best position. In (6), α is the expansion-constriction factor.

2.3. QPSO with Double-Well Algorithm. Based on the double-δ potential well quantum model, a QPSO with double-well algorithm is proposed by Xu et al. [14] to resolve the problem of population diversity decline due to the fast convergence of QPSO algorithm. In this algorithm, the motion of particles is simulated as in a double-well space. The double-well based position update formula of particle $\mathbf{x}_k$ can be defined as follows:

$$\begin{aligned} \mathbf{x}_k(t+1) = {} & \frac{1}{2}\left[\left(1 + \beta_k(t)\right)\mathbf{P}_k^1(t) + \left(1 - \beta_k(t)\right)\mathbf{P}_k^2(t)\right] \\ & \pm \frac{\mathbf{L}_k(t)}{2} \ln\left(\frac{1}{u_k(t)}\right). \end{aligned} \tag{7}$$

In (7), $\beta_k(t)$ is the distance coefficient between two δ potential wells, and $\beta_k(t)$ is a random number uniformly distributing over $[0, 1]$. If $u_k(t) > 0.5$, "±" would be chosen as "+", or it would be chosen as "−". The two attractors $\mathbf{P}_k^1(t)$ and $\mathbf{P}_k^2(t)$ are

defined as (8), and the characteristic length $\mathbf{L}_k(t)$ is defined as (9).

$$\begin{aligned}\mathbf{P}_k^1(t) &= \beta_k^1(t)\,\mathbf{pbest}_k(t) + \left(1-\beta_k^1(t)\right)\mathbf{gbest}_k^1(t),\\ \mathbf{P}_k^2(t) &= \beta_k^2(t)\,\mathbf{pbest}_k(t) + \left(1-\beta_k^2(t)\right)\mathbf{gbest}_k^2(t)\end{aligned} \tag{8}$$

$$\begin{aligned}&\mathbf{L}_k(t)\\ &= \alpha\left|\left(1+\beta_k(t)\right)\mathbf{P}_k^1(t) + \left(1-\beta_k(t)\right)\mathbf{P}_k^2(t) - 2\mathbf{x}_k(t)\right|.\end{aligned} \tag{9}$$

In (8), $\beta_k^1(t) \sim U(0,1)$, $\beta_k^2(t) \sim U(0,1)$, $\mathbf{pbest}_k(t)$ is the individual best position, and $\mathbf{gbest}_k^1(t)$ and $\mathbf{gbest}_k^2(t)$ are two guider particles.

3. MOQPSO-D/S Algorithm

Given the superior performance, the QPSO algorithm is introduced to solve the multiobjective problems. And the multiobjective quantum-behaved particle swarm optimization algorithm (MOQPSO) [14–18] is put forward. However, the requirement for solving a multiobjective optimization problem is different from solving a single objective optimization problem. It requires that the diversity of solutions should be good, the distribution of solutions should be uniform, and the distance between solution and the true Pareto front should be as close as possible. The fast convergence performance of QPSO would lead to the premature convergence of MOQPSO. And the global convergence to an optimal solution of QPSO would go against the diversity and distribution of solutions obtained by MOQPSO. In order to improve the performance of MOQPSO algorithm to solve the multiobjective programming problem, a multiobjective quantum-behaved particle swarm optimization with double/single-well (MOQPSO-D/S) algorithm was proposed by combining the QPSO algorithm and the QPSO with double-well algorithm. In this algorithm, a double/single-well based position update method is adopted to balance the relationship of solutions' diversity, convergence precision, and convergence rate. A hybrid random mutation method is also adopted to avoid premature convergence and improve the convergence precision. A two-stage based guider particles selection method is applied to this algorithm to improve the solution distribution uniformity and convergence rate.

3.1. Double/Single-Well Based Position Update Method. The so-called double/single-well based position update method includes two aspects. On one hand, in order to improve solutions' diversity and avoid premature convergence, the particle position is updated based on QPSO with double-well algorithm in the early optimization stage. On the other hand, in order to improve convergence precision and convergence rate, the particle position is updated based on QPSO algorithm in the final optimization stage. The double/single-well based position update formula is as follows:

$$\mathbf{x}_k(t+1) = \begin{cases} \dfrac{1}{2}\left[\left(1+\beta_k(t)\right)\mathbf{P}_k^1(t) + \left(1-\beta_k(t)\right)\mathbf{P}_k^2(t)\right] \pm \dfrac{\mathbf{L}_k(t)}{2}\ln\left(\dfrac{1}{u_k(t)}\right), & t \le \dfrac{t_{\max}}{2} \\ \mathbf{P}_k^1(t) \pm \dfrac{\mathbf{L}_k(t)}{2}\ln\left(\dfrac{1}{u_k(t)}\right), & t > \dfrac{t_{\max}}{2}. \end{cases} \tag{10}$$

In (10), $t_{\max}$ is the maximum generation. If $u_k(t) > 0.5$, "±" would be chosen as "+", or it would be chosen as "−". The characteristic length $\mathbf{L}_k(t)$ is defined as in the following equation.

$$\mathbf{L}_k(t) = \begin{cases} \alpha\left|\left(1+\beta_k(t)\right)\mathbf{P}_k^1(t) + \left(1-\beta_k(t)\right)\mathbf{P}_k^2(t) - 2\mathbf{x}_k(t)\right|, & t \le \dfrac{t_{\max}}{2} \\ 2\alpha\left|\mathbf{P}_k^1(t) - \mathbf{x}_k(t)\right|, & t > \dfrac{t_{\max}}{2}. \end{cases} \tag{11}$$

3.2. Hybrid Random Mutation Method. In order to avoid premature convergence and improve convergence precision, a hybrid random mutation method is proposed. The so-called hybrid random mutation is to conduct uniform random mutation in the early optimization stage of the algorithm and to conduct Gaussian random mutation in the final optimization stage. Particles can traverse on the optimal space with equal probability completely by conducting uniform random mutation, which can improve the diversity of solution and avoid premature convergence. Particles can move around the original position by conducting Gaussian mutation so as to avoid lowering optimization efficiency caused by the too large mutation range in the final optimization stage and improve convergence precision. The detail of hybrid random mutation is as follows:

$$\begin{aligned}&\mathbf{x}_k(t)\\ &= \begin{cases} \mathbf{x}_k(t), & \text{rand} > p_m \\ \text{Rand}\left(\mathbf{x}_{\min}, \mathbf{x}_{\max}\right), & \text{rand} \le p_m \cap t \le \dfrac{t_{\max}}{2} \\ \text{Gaussian}\left(\mathbf{x}_k(t), \sigma\right), & \text{rand} \le p_m \cap t > \dfrac{t_{\max}}{2}. \end{cases}\end{aligned} \tag{12}$$

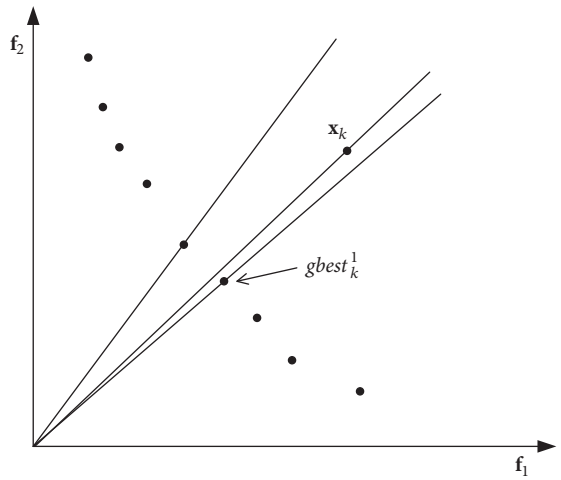


Figure 1: Sigma method based guider particle selection.

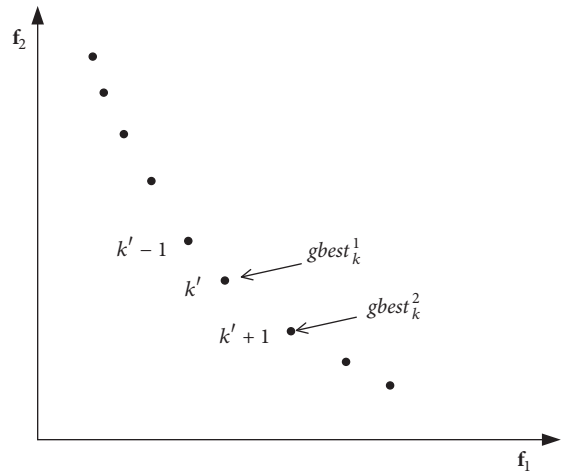


Figure 2: The $\mathbf{gbest}_k^2$ selection.

In (12), rand is a random number uniformly distributing on $[0, 1]$, $[\mathbf{x}_{\min}, \mathbf{x}_{\max}]$ is optimization range, Rand(•) is the uniform random mutation operator, and Gaussian(•) is the Gaussian random mutation operator. p_m is the mutation probability, which is defined as follows:

$$p_m = \begin{cases} 0.2, & t \le \dfrac{t_{\max}}{4} \\ -\dfrac{4}{15}\dfrac{t}{t_{\max}} + \dfrac{4}{15}, & t > \dfrac{t_{\max}}{4}. \end{cases} \tag{13}$$

3.3. Two-Stage Based Guider Particles Selection Method. In MOQPSO-D/S algorithm, the guider particles for each particle should be selected during iteration. The selection of guider particles from the external archive, with application of the roulette method [17], is not conducive to the distribution uniformity of solution, due to its strong randomicity. In order to address the issue above, a two-stage based guider particles selection method is proposed. The so-called two-stage based guider particles selection method includes the following two steps. Firstly, particles in the external archive should be sorted in descending order according to their crowding distance, and the 10% of particles with the biggest crowding distance is preliminary selected. Secondly, the particle with the closest sigma value to particle $\mathbf{x}_k$ is selected as the guider particle $\mathbf{gbest}_k^1$ for $\mathbf{x}_k$. The calculation method of the crowding distance for each particle in external archive can be referred to literature [19]. The calculation method of sigma value [20] for each particle is shown as in (14), and the detail of guider particle selection by adopting sigma value method is shown in Figure 1. When the particle position is updated based on QPSO with double-well, the second guider particle should be selected. The neighbor particle, with the larger distance to $\mathbf{gbest}_k^1$, will be selected as the second guider particle $\mathbf{gbest}_k^2$. The detail of this method for $\mathbf{gbest}_k^2$ selection is shown in Figure 2.

$$\text{sigma}(\mathbf{x}) = \frac{\left(f_1^2(\mathbf{x}) - f_2^2(\mathbf{x}), \ldots, f_o^2(\mathbf{x}) - f_o^2(\mathbf{x})\right)}{\sum_{l=1}^{o} f_l^2(\mathbf{x})}. \tag{14}$$

3.4. Algorithm's Complexity. Suppose that the number of objective functions in the multiobjective problem is o; the particle population is popsize; the size of external archive is N_{archive}. When operating one iteration, the complexity of MOQPSO-D/S for solving the multiobjective problem includes the following parts:

(1) The complexity of calculating the fitness of all particles equals $O(o \times \text{popsize})$.

(2) The complexity of calculating the individual best positions of all particles equals $O(o \times \text{popsize})$.

(3) After updating the positions of all particles, the complexity of comparing all particles in the population with the particles in the external archive equals $O(o \times \text{popsize} \times N_{\text{archive}})$.

(4) For updating the external archive, the complexity of conducting the crowding distance sorting equals $O(o \times (\text{popsize} + N_{\text{archive}})^2)$.

(5) After updating the external archive, the complexity of calculating the sigma values of all particles in the external archive equals $O(N_{\text{archive}})$.

Thus, when operating one iteration, the complexity of MOQPSO-D/S for solving the multiobjective problem equals $O(o \times (\text{popsize} + N_{\text{archive}})^2)$, which equals the complexity of MOQPSO-CD (the multiobjective quantum-behaved particle swarm optimization algorithm based on QPSO and crowding distance sorting) [17] and is acceptable.

4. Modeling WTA Problem Based on Fuzzy Multiobjective Programming

WTA in air and missile defense is to decide the problems of which fire unit should be chosen to intercept air targets, which air target will be intercepted by the chosen fire unit, and how many interceptors should be launched to intercept the chosen air target by the chosen fire unit. The goal of WTA decision-making is to obtain maximum interception efficiency and minimum interception consumption.

4.1. Model Assumptions and Parameter Settings. Suppose that in air and missile defense operation, m fire units $W = \{W_1, W_2, \ldots, W_m\}$ need intercept n incoming targets $T = \{T_1, T_2, \ldots, T_n\}$, where $T_1 \sim T_a$ belong to Type A (aerodynamic target), the remaining targets belong to Type B (ballistic target). Assuming that N_A or more interceptors are needed to intercept a target belonging to Type A and N_B or more interceptors are needed to intercept a target belonging to Type B, different types of targets need to be intercepted by different interceptors; for example, a target belonging to Type A can only be intercepted by Type A interceptors. The fire unit W_i $(i = 1, 2, \ldots, m)$ can intercept different types of targets but can only intercept a type of targets simultaneously. Due to the sensor error and other issues, the targets threat value is uncertain. The threat value of T_j $(j = 1, 2, \ldots, n)$ will be characterized with application of triangular fuzzy number $\widetilde{w}_j = (w_j^p, w_j^m, w_j^o)$, where w_j^p, w_j^m, and w_j^o denote the most pessimistic, the most likely, and the most optimistic threat value, respectively. W_i $(i = 1, 2, \ldots, m)$ launches y_{ij} interceptors to intercept T_j $(j = 1, 2, \ldots, n)$, and the single-shot kill probability is D_{ij}.

The correlative parameters are defined as follows:

m_i^A denotes the number of Type A targets that W_i can intercept simultaneously,

$\widehat{m}_i^A$ denotes the number of Type A interceptors that W_i can launch simultaneously,

N_i^A denotes the number of Type A interceptors that W_i stores,

C_i^A denotes the value of a Type A interceptor that W_i stores,

m_i^B denotes the number of Type B targets that W_i can intercept simultaneously,

$\widehat{m}_i^B$ denotes the number of Type B interceptors that W_i can launch simultaneously,

N_i^B denotes the number of Type B interceptors that W_i stores,

C_i^B denotes the value of a Type B interceptor that W_i stores.

4.2. Modeling Based on Fuzzy Multiobjective Programming. Taking the maximizing interception efficiency and the minimizing interception consumption as goals, a multiobjective optimization based WTA model is established as follows.

$$\max \quad f_1 = \sum_{j=1}^{n} \widetilde{w}_j \left[1 - \prod_{i=1}^{m} \left(1 - D_{ij}\right)^{y_{ij}} \right], \tag{15}$$

$$\min \quad f_2 = \sum_{i=1}^{m} \left[\sum_{j=1}^{a} C_i^A y_{ij} + \sum_{j=a+1}^{n} C_i^B y_{ij} \right]. \tag{16}$$

The model needs to meet the resource constraints and the firepower constraint. For the fire unit W_i $(i = 1, 2, \ldots, m)$, the resource constraints are needed to be met as follows.

(1) Interception Constraint. The fire unit W_i can only intercept a type of target simultaneously; thus the model should meet

$$\delta\left(\sum_{j=1}^{a} y_{ij}\right) + \delta\left(\sum_{j=a+1}^{n} y_{ij}\right) \le 1. \tag{17}$$

In (17), $\delta(\xi) = \begin{cases} 1, & \xi > 0 \\ 0, & \xi \le 0 \end{cases}$.

(2) Interception Capability Constraint. The fire unit W_i can intercept less than m_i^A Type A targets or m_i^B Type B targets simultaneously; thus the model should meet

$$\begin{aligned} \sum_{j=1}^{a} \delta\left(y_{ij}\right) &\le m_i^A \\ \sum_{j=a+1}^{n} \delta\left(y_{ij}\right) &\le m_i^B. \end{aligned} \tag{18}$$

(3) Number Constraint of Interceptors. The fire unit W_i can launch less than $\widehat{m}_i^A$ and N_i^A to intercept Type A targets or launch less than $\widehat{m}_i^B$ and N_i^B to intercept Type B targets; thus the model should meet

$$\begin{aligned} \sum_{j=1}^{a} y_{ij} &\le \min\left(\widehat{m}_i^A, N_i^A\right) \\ \sum_{j=a+1}^{n} y_{ij} &\le \min\left(\widehat{m}_i^B, N_i^B\right). \end{aligned} \tag{19}$$

For intercepting the incoming target T_j $(j = 1, 2, \ldots, n)$, the firepower constraint is needed to be met as follows.

N_A or more Type A interceptors are needed to intercept a Type A target, and N_B or more Type B interceptors are needed to intercept a Type B target; thus the model should meet

$$\begin{aligned} \sum_{i=1}^{m} y_{ij} &\ge N_A, \quad j = 1, 2, \ldots, a \\ \sum_{i=1}^{m} y_{ij} &\ge N_B, \quad j = a+1, a+2, \ldots, n. \end{aligned} \tag{20}$$

4.3. Clarifying Model Equivalently. This WTA model cannot be resolved by adopting multiobjective optimization algorithm, for its objective function containing fuzzy parameter shown as (15). Therefore, it is necessary to clarify the fuzzy multiobjective programming based WTA model to be a certain model equivalently. The necessity degree principle of uncertainty theory [21] is adopted to turn the objective function with fuzzy parameter into a certain objective function. The detailed process is as follows.

Firstly, the objective function shown as (15) is turned into a fuzzy chance constrained programming based model

$$\begin{aligned} &\min \quad c \\ &\text{s.t.} \quad \text{Nec}\left\{\sum_{j=1}^{n}\widetilde{w}_j\prod_{i=1}^{m}\left(1-D_{ij}\right)^{y_{ij}}\le c\right\}\ge\theta. \end{aligned} \tag{21}$$

In (21), θ denotes the necessity degree that the whole rest incoming targets' threat value is equal to or less than c.

Secondly, the fuzzy chance constrained programming based model shown as in (21) is turned into certain form with application of the necessity degree principle of uncertainty theory. Because $\text{Nec}\{\sum_{j=1}^{n}\widetilde{w}_j\prod_{i=1}^{m}(1-D_{ij})^{y_{ij}}\le c\}\ge\theta$ can be turned into $\sum_{j=1}^{n}[w_j^m+\theta(w_j^o-w_j^m)]\prod_{i=1}^{m}(1-D_{ij})^{y_{ij}}\le c$ equivalently, (17) can be turned into

$$\begin{aligned} &\min \quad c \\ &\text{s.t.} \quad \sum_{j=1}^{n}\left[w_j^m+\theta\left(w_j^o-w_j^m\right)\right]\prod_{i=1}^{m}\left(1-D_{ij}\right)^{y_{ij}}\le c. \end{aligned} \tag{22}$$

Finally, the certain objective function is obtained as follows:

$$\min \quad f_1'=\sum_{j=1}^{n}\left[w_j^m+\theta\left(w_j^o-w_j^m\right)\right]\prod_{i=1}^{m}\left(1-D_{ij}\right)^{y_{ij}}. \tag{23}$$

In this way, the fuzzy multiobjective programming based WTA model is turned into a certain multiobjective optimization model. The model clarified could be solved with application of multiobjective optimization algorithm. The WTA model constructed in this paper will be solved by adopting MOQPSO-D/S algorithm in the next section.

5. Solving WTA Model Based on MOQPSO-D/S

5.1. Particle Coding. The MOQPSO-D/S algorithm cannot solve the discrete problem directly. Therefore, it is necessary to code the position of particles when the WTA model is resolved by adopting MOQPSO-D/S algorithm. The position of particles will be coded with application of matrix encoding. $\mathbf{y}$ denotes the position of a particle, and it can be coded as follows:

$$\mathbf{y}=\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{m1} & y_{m2} & \cdots & y_{mn} \end{bmatrix}. \tag{24}$$

In (24), y_{ij} denotes the number of interceptors that W_i launches to intercept T_j.

5.2. Illegal Particle Code Adjusting. When initializing and updating the particle position, the particle position may not satisfy the constraints of the WTA model; namely, the infeasible solution comes out. Therefore, it is necessary to check the particle coding legitimacy and to adjust the illegal particle code. The process of particle code checking and illegal code adjusting is shown in Figure 3.

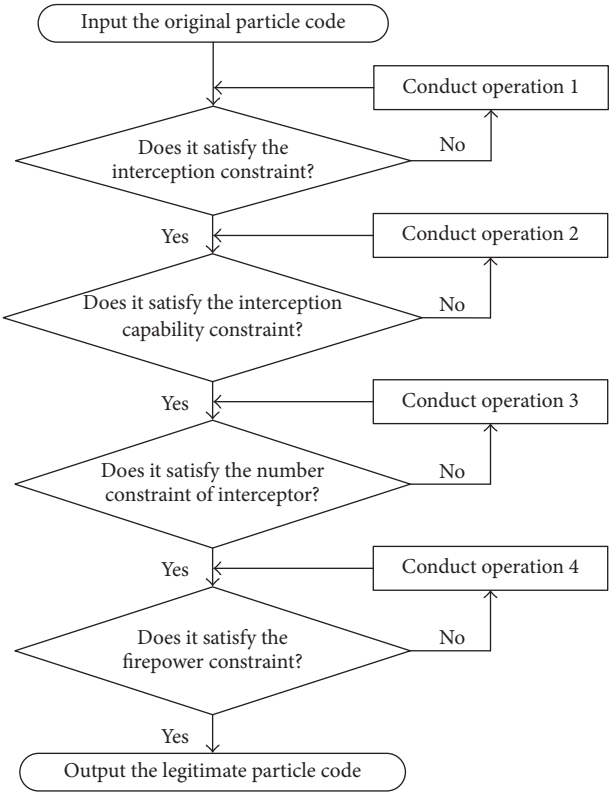


Figure 3: The process of particle code checking and illegal code adjusting.

For example, when a particle code does not meet the interception capability constraint for fire unit W_i ($i = 1, 2, \ldots, m$), Operation 2 should be conducted as follows:

Firstly, judge whether $\sum_{j=1}^{a} y_{ij} = 0$ or $\sum_{j=a+1}^{n} y_{ij} = 0$.

Case 1. One has

$$\sum_{j=1}^{a} y_{ij} = 0 \tag{25}$$

C1S1 (Case 1 Step 1). Let count_A $= \sum_{j=1}^{a}\delta(y_{ij})$; if $\delta(y_{ij}) = 1$, then store the subscript j into the set Ω_{A}.

C1S2. Let $N_{\text{exc\_A}} = \text{count\_A-}m_i^{\text{A}}$.

C1S3. Select randomly $N_{\text{exc\_A}}$ subscript from the set Ω_{A}, and let the corresponding $y_{ij} = 0$.

Case 2. One has

$$\sum_{j=a+1}^{n} y_{ij} = 0 \tag{26}$$

C2S2. Let count_B $= \sum_{j=a+1}^{n} \delta(y_{ij})$; if $\delta(y_{ij}) = 1$, then store the subscript j into the set Ω_B.

C1S2. Let $N_{\text{exc\_B}} = \text{count\_B-}m_i^B$.

C1S3. Select randomly $N_{\text{exc\_B}}$ subscript from the set Ω_B, and let the corresponding $y_{ij} = 0$.

Finally, output the particle code that meets the interception capability constraint for fire unit W_i $(i = 1, 2, \ldots, m)$.

Due to limitations on space, Operation 1, Operation 3, and Operation 4 will not be introduced here.

5.3. Solving Steps. After the particles are encoded and the illegal particles are adjusted, the MOQPSO-D/S algorithm can be adopted to solve the fuzzy multiobjective programming based WTA model clarified equivalently. The detailed steps are as follows.

Step 1. Set the algorithm parameters, and initialize the particle swarm randomly.

Step 2. Check out whether every particle code meets the constraints. If a particle code does not meet the constraints, then adjust it. Otherwise, turn to Step 3.

Step 3. Calculate the fitness of every particle $f(\mathbf{y}) = (f_1'(\mathbf{y}), f_2(\mathbf{y}))$ according to (16) and (23); let $\mathbf{pbest} = \mathbf{y}$, $f(\mathbf{pbest}) = f(\mathbf{y})$, where **pbest** denotes the individual best position of $\mathbf{y}$.

Step 4. Initialize the external storage file ARCHIVE. Calculate the dominated relationship between particles according to their fitness, and store the nondominated particles into ARCHIVE, then calculate the crowding distance and sigma value of particles in ARCHIVE.

Step 5. Initialize iteration, and let $t = 1$.

Step 6. Select guider particles $\mathbf{gbest}^1$ and $\mathbf{gbest}^2$ with application of two-stage method. If $t > t_{\max}/2$, the guider particle $\mathbf{gbest}^2$ would not be selected. Update the position of every particle according to (10).

Step 7. Judge whether the mutation condition is satisfied or not. If it is satisfied, conduct mutation for the particle by adopting the hybrid random mutation method. Otherwise, turn to Step 8.

Step 8. Check out whether every particle code meets the constraints. If a particle code does not meet the constraints, then adjust it. Otherwise, turn to Step 9.

Step 9. Calculate the fitness of every particle $f(\mathbf{y}(t)) = (f_1'(\mathbf{y}(t)), f_2(\mathbf{y}(t)))$, and calculate the dominated relationship between the current position and the individual best position of every particle. If the current position dominates the individual best position of the particle $\mathbf{y}_k$, then let $\mathbf{pbest}_k = \mathbf{y}_k(t)$, $f(\mathbf{pbest}_k) = f(\mathbf{y}_k(t))$. If the current position and the individual best position do not dominate each other, then conduct the operation $\mathbf{pbest}_k = \mathbf{y}_k(t)$, $f(\mathbf{pbest}_k) = f(\mathbf{y}_k(t))$ with the probability of 0.5. Otherwise, turn to Step 10.

Step 10. Calculate the dominated relationship between every particle and particles in ARCHIVE. If the particle $\mathbf{y}_k$ dominates the particle $\mathbf{y}_g$ in ARCHIVE, then store the particle $\mathbf{y}_k$ into ARCHIVE, delete the particle $\mathbf{y}_g$, and turn to Step 11. If the particle $\mathbf{y}_k$ and the particle $\mathbf{y}_g$ do not dominate each other, then store the particle $\mathbf{y}_k$ into ARCHIVE, and turn to Step 11. Otherwise, turn to Step 12.

Step 11. Calculate the crowding distance of particles in ARCHIVE updated, and calculate the sigma value of the particles newly stored. Judge whether the number of particles in ARCHIVE exceeds its capacity. If it exceeds the capacity, then delete the particle with the smallest crowning distance. Otherwise, turn to Step 12.

Step 12. Let $t = t + 1$. If $t > t_{\max}$, then end the iteration, and output all the particles in ARCHIVE as the WTA alternatives for air and missile defense combat. Otherwise, turn to Step 6.

6. Simulation Experiment and Analysis

In order to verify the performance of MOQPSO-D/S algorithm for solving multiobjective optimization problems and WTA problem in air and missile defense operation, two simulation experiments are designed. In experiment 1, the convergence and solution distribution uniformity of the MOQPSO-D/S algorithm for solving multiobjective optimization problem are validated. In experiment 2, after the simulation scenarios are set, the rationality of the fuzzy multiobjective programming based WTA model and the feasibility of the MOQPSO-D/S algorithm for solving the model constructed in this paper are validated by solving the WTA model with application of MOQPSO-D/S algorithm coded. The experiments above are conducted on the computer with INTEL CORE i5-4590, 3.3 GHz CUP, and 4 G RAM after programming with MATLAB 2013a. The capability of ARCHIVE for all multiobjective optimization algorithms is 100.

6.1. Verify the Performance of MOQPSO-D/S for Solving Multiobjective Optimization Problems. The test functions ZDT1~ZDT4 [22] are chosen to verify the performance of MOQPSO-D/S algorithm for solving multiobjective optimization problems. The multiobjective quantum-behaved particle swarm optimization algorithm with double-potential well and share-learning (MOQPSO-DPS) [14], the MOQPSO-CD [17], and the multiobjective particle swarm optimization algorithm based on crowding distance sorting (MOPSO-CD) [23] are chosen as the comparative algorithms. Set the particle population to be 100, and set the iteration number to be 200. The variables number of the test functions ZDT1~ZDT3 is 30, and the variables number of the test function ZDT4 is 10. The algorithms run 30 times independently. Select the solution of one time from the 30 times of every algorithm for solving test functions ZDT1~ZDT4 randomly, and the results are shown in Figure 4.

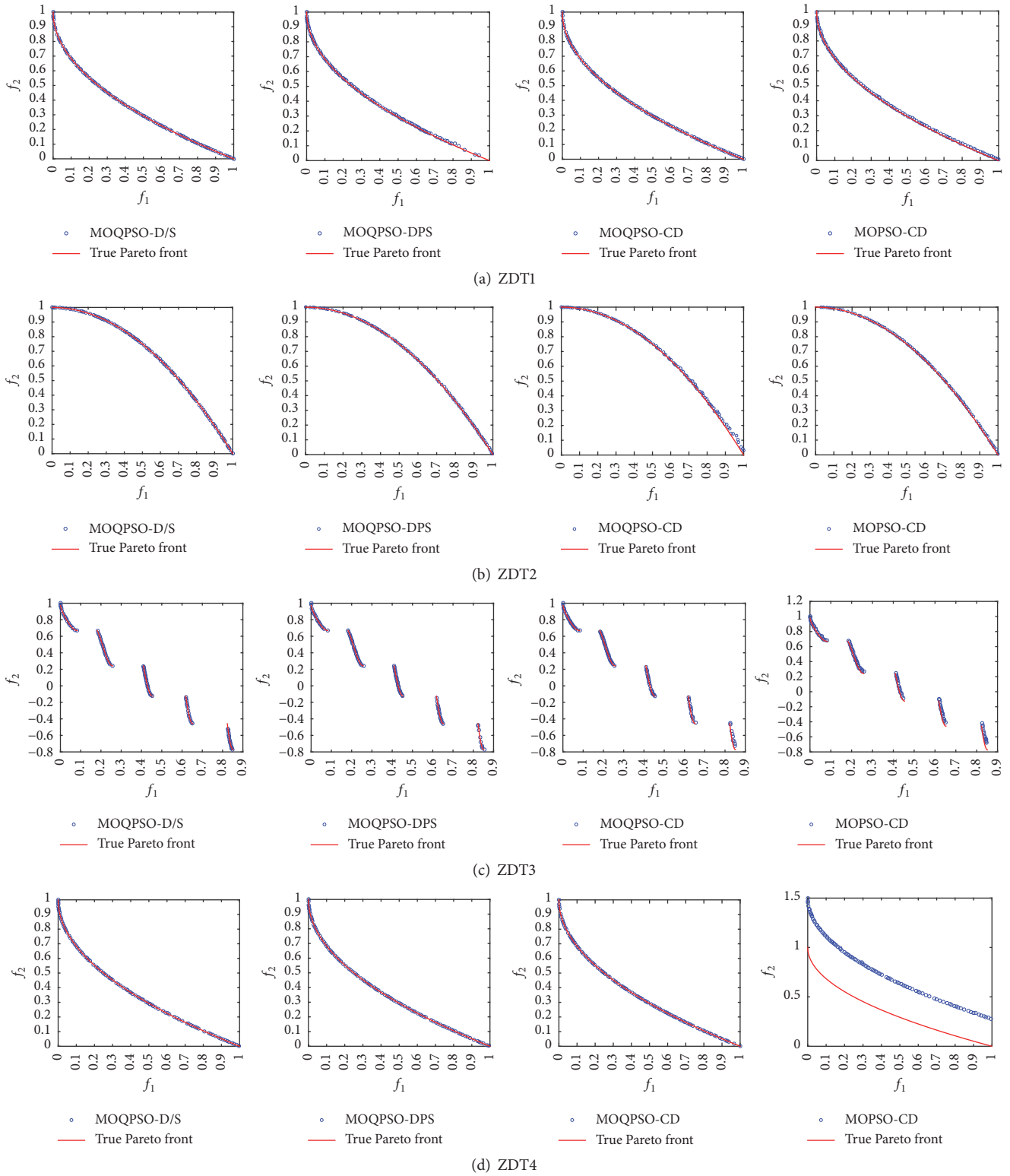


Figure 4: Pareto front obtained by solving ZDT1~ZDT4 with application of four algorithms.

Table 1: ER of four algorithms for solving the test functions.

Algorithms	Test functions			
	ZDT1	ZDT2	ZDT3	ZDT4
MOQPSO-D/S	**0.0097**	**0.0087**	**0.0083**	**0.0290**
MOQPSO-DPS	0.0150	0.0127	0.0187	0.0370
MOQPSO-CD	0.0113	0.0253	0.0343	0.0337
MOPSO-CD	0.0283	0.0363	0.1143	0.2783

Table 2: SP of four algorithms for solving the test functions.

Algorithms	Test functions			
	ZDT1	ZDT2	ZDT3	ZDT4
MOQPSO-D/S	**0.0043**	0.0075	**0.0089**	**0.0074**
MOQPSO-DPS	0.0117	**0.0073**	0.0114	0.0131
MOQPSO-CD	0.0071	0.0101	0.0121	0.0087
MOPSO-CD	0.0079	0.0084	0.0143	0.0241

Table 3: Mean running time of four algorithms for solving the test functions.

Algorithms	Test functions			
	ZDT1	ZDT2	ZDT3	ZDT4
MOQPSO-D/S	**5.1134**	**6.7634**	**4.0197**	**4.5163**
MOQPSO-DPS	7.2471	7.3521	4.4110	6.0217
MOQPSO-CD	8.3283	8.1126	4.7853	6.4313
MOPSO-CD	10.3590	11.8781	6.7596	8.8357

Error ration (ER) [14] and spacing metric (SP) [24] are chosen to weigh the convergence of four algorithms for solving the test functions and the distribution uniformity of the Pareto optimal solution, respectively. And, the mean running time is chosen to weigh the operating rate. The calculations of ER and SP are shown in (27) and (28). ER index represents the ratio of the Pareto solutions obtained by four algorithms in the true Pareto optimal solution set. The smaller the ER value is, the better the convergence is. The smaller the SP value is, the more uniformly the Pareto optimal solutions distribute.

$$\mathrm{ER} = \frac{\sum_{i=1}^{n_P} e_i}{n_P}. \tag{27}$$

In (27), e_i denotes the probability that a Pareto optimal solution is not in the true Pareto optimal solution set, if it is in, $e_i = 0$; or, $e_i = 1$. n_P denotes the number of elements in a Pareto optimal solution set.

$$\mathrm{SP} = \sqrt{\frac{1}{n_P - 1}\sum_{k=1}^{n_P}\left(\overline{d} - d_k\right)^2}. \tag{28}$$

In (28), d_k denotes the Hamming distance between the dot corresponding to solution $\mathbf{x}_k$ and its neighbor dot in the Pareto front. The calculation of d_k is shown as in (29). $\overline{d}$ denotes the mean distance, shown in (30).

$$d_k = \min_{r \neq k}\left(\sum_{\xi=1}^{o}\left|f_\xi(\mathbf{x}_k) - f_\xi(\mathbf{x}_r)\right|\right), \tag{29}$$

$$\overline{d} = \frac{1}{n_P}\sum_{k=1}^{n_P} d_k. \tag{30}$$

The comparative results are shown in Tables 1–3, where the ER values, SP values, and the mean running time values are the average values of four algorithms for solving the test functions 30 times.

From Table 1, we can see that the ER indexes of MOQPSO-D/S for solving the test functions are smaller than the remaining algorithms'. Combining with Figure 4, it can be concluded that the convergence of MOQPSO-D/S is better than the remaining algorithms'. And, it is proved that the double/single-well based position update method and the hybrid random mutation method can improve convergence precision.

From Table 2, we can see that the SP indexes of MOQPSO-D/S for solving the test functions are smaller than the remaining algorithms', except for solving ZDT2. Combining with Figure 4, it can be concluded that the distribution uniformity of the Pareto optimal solution of MOQPSO-D/S is better than the remaining algorithms'. And, it is proved that the two-stage based guider particles selection method can improve solution distribution uniformity.

Table 4: Parameters value.

	W_1	W_2	W_3	W_4	W_5	W_6
m_i^a	1	3	2	5	4	0
$\widehat{m}_i^a$	3	6	4	10	8	0
m_i^b	0	0	0	1	1	1
$\widehat{m}_i^b$	0	0	0	2	2	2
N_i^a	6	6	8	9	8	0
N_i^b	0	0	0	2	2	2
C_i^a	0.1	0.2	0.15	0.25	0.3	-
C_i^b	-	-	-	0.45	0.5	0.6

Table 5: Single-shot kill probability.

	W_1	W_2	W_3	W_4	W_5	W_6
T_1	0.45	0.32	0.54	0.47	0.75	0
T_2	0.5	0.27	0.34	0.73	0.78	0
T_3	0.28	0.62	0.58	0.56	0.57	0
T_4	0	0	0	0.63	0.55	0.7
T_5	0	0	0	0.45	0.5	0.85

From Table 3, we can see that the mean running time of MOQPSO-D/S for solving the test functions is smaller than the remaining algorithms'. It can be concluded that the operating rate of MOQPSO-D/S is better than the remaining algorithms'. And, it is proved that the two-stage based guider particles selection method can improve operating rate.

6.2. Verify the Performance of MOQPSO-D/S for Solving Uncertain WTA Problem. In order to validate the rationality of the fuzzy multiobjective programming based WTA model and to verify the performance of the MOQPSO-D/S algorithm for solving the model simultaneously, the following case for air and missile defense combat is given.

Suppose that, in air and missile defense operation, 6 fire units W_i $(i = 1, 2, \ldots, 6)$ need intercept 5 incoming targets T_j $(j = 1, 2, \ldots, 5)$. The number of targets that W_i $(i = 1, 2, \ldots, 6)$ can intercept simultaneously, the number of interceptors that W_i can launch simultaneously, the number of interceptors that W_i stores, and the dimensionless value of an interceptor that W_i stores are listed in Table 4. The incoming targets $T_1 \sim T_3$ belong to Type A, and the remaining targets belong to Type B. The target threat values are weighed with application of triangular fuzzy number as follows: $\widetilde{w}_1 = (0.41, 0.43, 0.45)$, $\widetilde{w}_2 = (0.48, 0.51, 0.54)$, $\widetilde{w}_3 = (0.33, 0.35, 0.37)$, $\widetilde{w}_4 = (0.87, 0.89, 0.91)$, $\widetilde{w}_5 = (0.90, 0.92, 0.94)$. The least interceptors to intercept a Type A target and a Type B target are $N_A = 1$ and $N_B = 2$, respectively. The necessity degree θ is set to be 0.8. The single-shot kill probability of the fire unit W_i $(i = 1, 2, \ldots, 6)$ that launches an interceptor to intercept the target T_j $(j = 1, 2, \ldots, 5)$ is listed in Table 5. The particle encoding method and the illegal particle code adjusting method are adopted in the subsequent algorithms for solving the example.

6.2.1. Analyze the Result of MOQPSO-D/S for Solving Uncertain WTA Problem. With different population size and different iterations, the WTA case is solved with application of MOQPSO-D/S algorithm. The algorithm for solving the WTA case under every condition runs 30 times independently. The mean running time is listed in Table 6. It can be seen from Table 6 that the real-time performance of MOQPSO-D/S algorithm can meet the need of WTA in air and missile defense combat.

Table 6: Mean running time/s.

Population size	Iterations		
	50	100	200
50	0.7504	1.4830	2.9733
100	1.5179	2.9313	5.9883
200	3.0576	6.0737	11.8876

With the population size and iterations both being 50, the solution of the WTA case solved by MOQPSO-D/S algorithm is shown in Figure 5. It can be seen from Figure 5 that the Pareto optimal solution, obtained by MOQPSO-D/S algorithm for solving the fuzzy multiobjective programming based WTA model in air and missile defense combat, distributes uniformly, where every dot refers to a WTA alternative. Therefore, it can provide a series of alternatives for commanders to make decision. The commanders could make the final decision by choosing a suitable WTA alternative from the solution with consideration of the battlefield situation and risk preference. Comparing with the single objective optimization based WTA method, which can only give a WTA alternative, this method could preferably adapt to the change of the battlefield situation and combine the tactical thought of the commander to make decision.

6.2.2. Compare with the Single Objective Optimization Based WTA Method. In order to validate the feasibility of the solution obtained by MOQPSO-D/S algorithm for solving WTA model, in comparison, the example is solved by adopting the method proposed in literature [6], taking maximizing effectiveness-cost ratio as the objective function and adopting

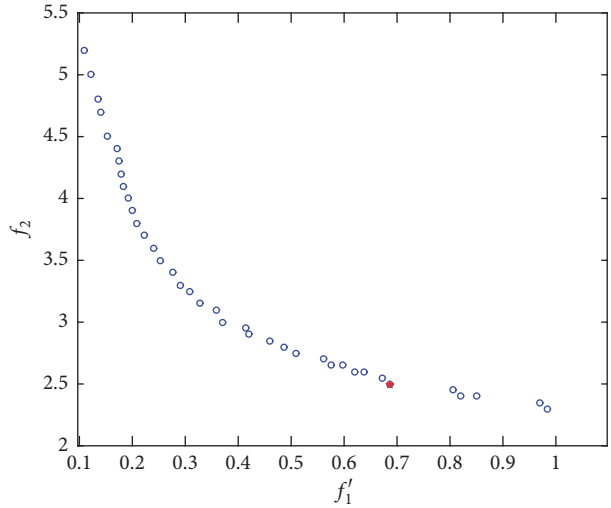


Figure 5: Solution of MOQPSO-D/S for solving the case.

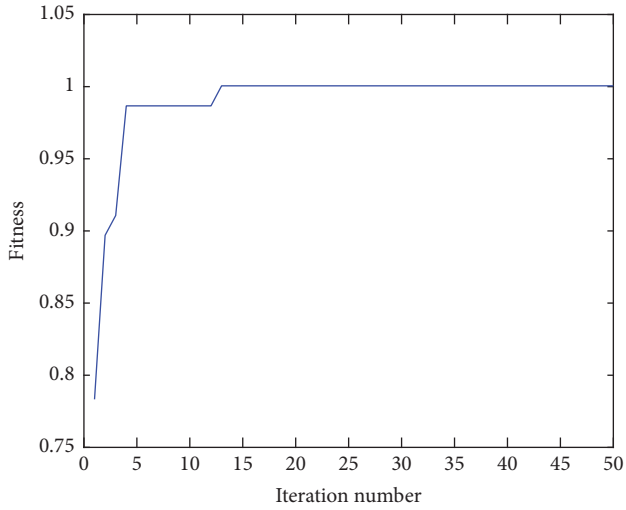


Figure 6: Best fitness over generations based on IDPSO algorithm.

IDPSO algorithm to solve the WTA model. With the population size and iterations being both 50, the single objective optimization method for solving the WTA case runs 30 times. The mean running time is 0.7193 s. The best fitness over generations based on IDPSO is shown as in Figure 6. The optimal WTA result is

$$\begin{array}{c} \\ W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \end{array} \begin{array}{c} \begin{array}{ccccc} T_1 & T_2 & T_3 & T_4 & T_5 \end{array} \\ \begin{bmatrix} 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}. \quad (31)$$

The optimal result is just the WTA alternative corresponding to the dot marked by a red five-angle star shown in Figure 5. And the mean running time is a little shorter than the MOQPSO-D/S algorithm with the same population size and iterations. It is proved that MOQPSO-D/S algorithm, when solving the WTA model, would disperse computation resource but can still obtain the satisfactory solution containing the optimal result obtained by single optimization method. Meanwhile, the fuzzy multiobjective programming WTA model for air and missile defense is proved to be reasonable and adopting MOQPSO-D/S algorithm to solve the model is feasible.

Table 7: Dominated ratio among four algorithms' Pareto optimal sets.

X_2	X_1			
	A_1	A_2	A_3	A_4
A_1	0	0	0.0185	0.0370
A_2	0.0588	0	0.1176	0.1765
A_3	0.0556	0	0	0
A_4	0.0755	0.0755	0.0755	0

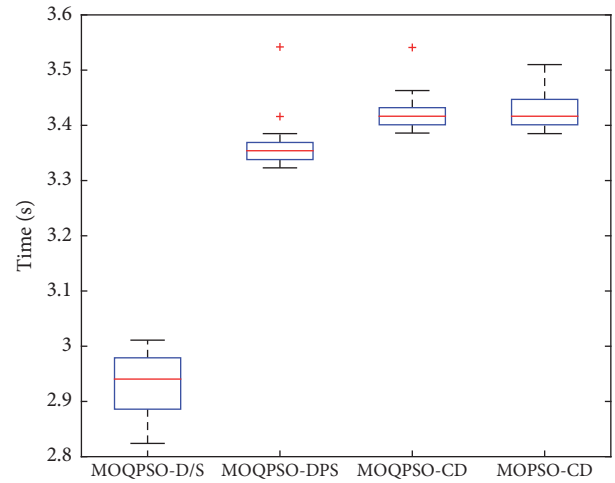


Figure 7: Statistical results for running time of four algorithms.

6.2.3. Compare with Other Multiobjective Optimization Algorithm for Solving WTA Method. In order to further verify the performance of MOQPSO-D/S algorithm for solving WTA problem, in comparison, the example is solved by adopting MOQPSO-DPS, MOQPSO-CD, and MOPSO-CD algorithm with the population size and iterations being both 100. Every algorithm for solving the WTA case runs 30 times independently. The statistical results for running time of every algorithm are shown in Figure 7. It can be seen from Figure 7 that the real-time performance of MOQPSO-D/S algorithm is obviously superior to the comparable algorithms'.

The convergence of every algorithm for solving the WTA case is weighed by dominated ratio $C(X_1, X_2)$ [25]. The calculation of $C(X_1, X_2)$ is shown as in (32). X_1 and X_2 denote the Pareto optimal solution sets obtained by two algorithms, respectively. If $C(X_1, X_2)$ is bigger than $C(X_2, X_1)$, the convergence of the algorithm corresponding to X_1 is superior to the algorithm corresponding to X_2. The convergence comparison among four algorithms is listed in Table 7. A_1, A_2, A_3, and A_4 listed in Table 7 denote the

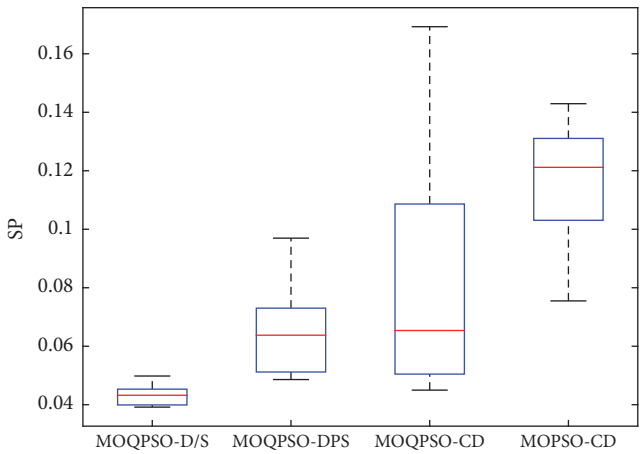


Figure 8: Statistical results for SP value of four algorithms.

Pareto optimal solution sets obtained by MOQPSO-D/S, MOQPSO-DPS, MOQPSO-CD, and MOPSO-CD algorithm, respectively. It can be concluded from Table 7 that the convergence of MOQPSO-D/S algorithm for solving the WTA case is the best of four algorithms.

$$C(X_1, X_2) = \frac{\text{numel}\left(\{\mathbf{x}_2 \in X_2 \mid \exists \mathbf{x}_1 \in X_1 : \mathbf{x}_1 \prec \mathbf{x}_2\}\right)}{\text{numel}(X_2)}. \quad (32)$$

In (32), the function numel(•) is used to count the number of elements in a set.

The distribution uniformity of the Pareto optimal solution set obtained by every algorithm for solving the WTA case is weighed by SP. The statistical results for SP value of the Pareto optimal solution set obtained by every algorithm are shown in Figure 8. It can be concluded from Figure 8 that the Pareto optimal solutions obtained by MOQPSO-D/S algorithm for solving the WTA case distribute the most uniformly, and its distribution uniformity is the most stable.

7. Conclusions

The goal of this article is to find a method to aid decision-making of WTA in air and missile defense combat under uncertainty. A fuzzy multiobjective programming based WTA model has been constructed. And multiobjective optimal algorithm, called MOQPSO-D/S, has been put forward to solve the fuzzy multiobjective programming based WTA model. The simulation results shows that the performance of MOQPSO-D/S algorithm is superior, the WTA model constructed in this paper is rational and the MOQPSO-D/S algorithm can solve the WTA model efficiently.

In future research, we may study WTA problem under uncertainty based on other uncertain methods, for example, the Taguchi method and the Bayesian method. Yet, we may study multiobjective optimization algorithms, for example, the game theory based methodologies, to solve the multiobjective programming based WTA model.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant no. 71771216 and Grant no. 71701209).

References

[1] J. M. Teague, K. R. Dorner, and W. B. Hartman, *Back to the future: integrated air and missile defense in the pacific*, Air University, Air Force Research Institute USA, Maxwell, USA, 2015.

[2] M. Ni, Z. Yu, F. Ma, and X. Wu, "A lagrange relaxation method for solving weapon-target assignment problem," *Mathematical Problems in Engineering*, vol. 2011, Article ID 873292, 10 pages, 2011.

[3] T. Q. Chang, J. W. Chen, N. Hao et al., "Terminating control of ant colony algorithm for armored unit," *System Engineering and Electronics*, vol. 37, no. 2, pp. 336–342, 2015.

[4] R. Wang H and C. Wang, "Variable value control technology of genetic algorithm for wta of ground target attacking," *Acta Armamentraii*, vol. 37, no. 10, pp. 1889–1895, 2016.

[5] H. Liang and F. Kang, "Adaptive chaos parallel clonal selection algorithm for objective optimization in WTA application," *Optik - International Journal for Light and Electron Optics*, vol. 127, no. 6, pp. 3459–3465, 2016.

[6] C. L. Fan, Q. H. Xing, M. F. Zheng et al., "Weapon-target allocation optimization algorithm based on IDPSO," *System Engineering and Electronics*, vol. 37, no. 2, pp. 336–342, 2015.

[7] X. Liu, Z. Liu, W. S. Hou et al., "Improved MOPSO algorithm for multi-objective programming model of weapon-target assignment," *Journal of Systems Engineering and Electronics*, vol. 35, no. 2, pp. 326–330, 2013.

[8] Y. Zhang, R. N. Yang, and J. L. Zuo, "Improved decomposition-based evolutionary algorithm for multi-objective optimization model of dynamic weapon-target assignment," *Acta Armamentraii*, vol. 36, no. 8, pp. 1533–1540, 2015.

[9] Y. Zhang, R. N. Yang, J. L. Zuo et al., "Improved MOEA/D for dynamic weapon-target assignment problem," *Journal of Harbin Institute of Technology*, vol. 22, no. 6, pp. 121–128, 2015.

[10] J. Sun, B. Feng, and W. B. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 325–331, IEEE, Portland, OR, USA, 2004.

[11] J. Sun, W. B. Xu, and B. Feng, "A global search strategy of quantum-behaved particle swarm optimization," in *Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems*, pp. 111–116, IEEE, Singapore, Singapore, 2004.

[12] Y. J. Won, K. H. Lee, and J. H. Lee, "Performance improvement of space borne SAR using antenna pattern synthesis based on quantum-behaved particle swarm optimization," *International Journal of Antennas and Propagation*, pp. 1–12, 2017.

[13] M. Xu, L. Zhang, B. Du et al., "A mutation operator accelerated quantum-behaved particle swarm optimization algorithm for hyperspectral endmember extraction," *Remote Sensing*, vol. 9, no. 3, p. 197, 2017.

[14] S.-H. Xu, X.-D. Mu, D. Chai, and P. Zhao, "Multi-objective quantum-behaved particle swarm optimization algorithm with

double-potential well and share-learning," *Optik - International Journal for Light and Electron Optics*, vol. 127, no. 12, pp. 4921–4927, 2016.

[15] Y. Li, Y. Wang, and L. Jiao, "MOQPSO: A new quantum-behaved particle swarm optimization for nearest neighborhood classification," in *Proceedings of the 2016 IEEE Region 10 Conference, TENCON 2016*, pp. 3165–3168, Singapore, November 2016.

[16] S. N. Omkar, R. Khandelwal, T. V. S. Ananth, G. Narayana Naik, and S. Gopalakrishnan, "Quantum behaved Particle Swarm Optimization (QPSO) for multi-objective design optimization of composite structures," *Expert Systems with Applications*, vol. 36, no. 8, pp. 11312–11322, 2009.

[17] Z. Shi and Q. W. Chen, "Multi-objective quantum-behaved particle swarm optimization algorithm based on QPSO and crowding distance sorting," *Control and Decision*, vol. 26, no. 4, pp. 540–547, 2011.

[18] A. B. Heyam, *A quantum behaved particle swarm approach to multi-objective optimization*, University of Birmingham, Birmingham, England, 2015.

[19] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[20] S. Mostaghim and J. Teich, "Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)," in *Proceedings of the IEEE Swarm Intelligence Symposium*, pp. 26–33, IEEE, Indianapolis, IN, USA, April 2003.

[21] B. D. Liu, "Uncertainty Theory," (Fifth Edition), Uncertainty Theory Laboratory, 2016, http://orsc.edu.cn/liu/ut.pdf.

[22] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.

[23] C. R. Raquel and P. C. Naval Jr., "An effective use of crowding distance in multiobjective particle swarm optimization," in *Proceedings of the 7th Annual conference on Genetic and Evolutionary Computation*, pp. 257–264, ACM, New York, NY, USA, June 2005.

[24] J. R. Schott, *Fault tolerant design using single and multicriteria genetic algorithm optimization*, Massachusetts Institute of Technology, Cambridge, MA, USA, 1995.

[25] E. Zitzler, *Evolutionary Algorithms for Multiobjective Optimization Methods and Applications*, Swiss Federal Institute of Technology Zurich, Zurich, Switzerland, 1999.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 8509783, 16 pages
https://doi.org/10.1155/2017/8509783

Research Article

Optimal Routing and Scheduling of Charge for Electric Vehicles: A Case Study

J. Barco,[1,2] A. Guerra,[3] L. Muñoz,[3] and N. Quijano[2]

[1]*Engineering Department, Electronics Engineering Program, Institución Universitaria CESMAG, Carrera 20A No. 14–54, San Juan de Pasto, Colombia*
[2]*Electrical and Electronics Engineering Department, Universidad de Los Andes, Carrera 1 Este No. 19 A 40, Bogotá, Colombia*
[3]*Mechanical Engineering Department, Universidad de Los Andes, Carrera 1 Este No. 19 A 40, Bogotá, Colombia*

Correspondence should be addressed to L. Muñoz; lui-muno@uniandes.edu.co

Received 8 April 2017; Revised 4 September 2017; Accepted 3 October 2017; Published 14 November 2017

Academic Editor: Domenico Quagliarella

There are increasing interests in improving public transportation systems. One of the proposed strategies for this improvement is the use of Battery Electric Vehicles (BEVs). This approach leads to a new challenge as the BEVs' routing is exposed to the traditional routing problems of conventional vehicles, as well as the particular requirements of the electrical technologies of BEVs. Examples of BEVs' routing problems include the autonomy, battery degradation, and charge process. This work presents a differential evolution algorithm for solving an electric vehicle routing problem (EVRP). The formulation of the EVRP to be solved is based on a scheme to coordinate the BEVs' routing and recharge scheduling, considering operation and battery degradation costs. A model based on the longitudinal dynamics equation of motion estimates the energy consumption of each BEV. A case study, consisting of an airport shuttle service scenario, is used to illustrate the proposed methodology. For this transport service, the BEV energy consumption is estimated based on experimentally measured driving patterns.

1. Introduction

The integration of Battery Electric Vehicles (BEVs) to the public transportation system has been encouraged by the favorable efficiency in the use of energy and the reduction of CO_2 emissions [1]. From the energy consumption perspective, BEVs are driven by high-efficiency motors with the possibility of implementing a regenerative braking system. Additionally, charging a BEV is less expensive than refueling a conventional vehicle because the electric energy is cheaper than its equivalent in fossil fuel (e.g., gasoline and diesel). From the emissions point of view, when a BEV is used in combination with renewable sources for the electricity generation, the outcome is a reduction in emissions associated with fossil fuel combustion; therefore, BEVs are one of the best alternatives to be integrated into cities as part of a public transportation system.

The use of BEVs in public transportation systems faces several challenges, mainly related to the combination of conventional-fuel-service characteristics with those of electric vehicles. Examples of these challenges are the routing of electric vehicles used in public transportation, the recharge scheduling, and the battery state of health (SOH) [2]. These three challenges are treated in this work using a methodology developed for the optimal routing and scheduling of charge for BEVs.

The first challenge is the routing of BEVs. In addition to the usual routing issues, the BEVs should be routed taking particular attention to minimizing energy consumption. For the routing of BEVs, two steps are considered. The first step consists in finding the minimum consumption paths to travel between two points. In this step, it is necessary to consider the technical characteristics of BEVs. The second step consists in determining optimal routes to satisfy the transportation demand in different places and at different schedules, while minimizing energy consumption. Similar to the first step, the calculation of optimal routes is performed considering the BEVs' characteristics. For this case, the vehicle's traveling

range is considered, mainly defined by the battery technology available [3]. This consideration could require intermediate recharge stages to extend the traveling range of the vehicles.

The second challenge is the scheduling for recharging. The recharge scheduling is a challenge as the energy rate variation during peak and valley hours must be considered. This process should be coordinated with the routing layout to guarantee a reliable operation, while the recharge costs are minimized. Additionally, for the recharge scheduling, it is necessary to consider the amount of energy required to perform the subsequent trip and the time required to recharge.

The third challenge is the SOH of the battery. The aim is to increase the battery lifespan to reduce the long-term operational cost. It must be considered that the useful life of the battery depends on the charge and discharge cycles [4]. This aspect is relevant because the battery is the most expensive component of the BEVs [5].

The three challenges for the implementation of BEVs for public transportation systems described in this section have been previously discussed in the literature. Nevertheless, most of those studies discuss one topic at a time. In [6, 7], the energy consumption during trips is considered. The authors propose a reduced order model for the energy consumption, which considers the BEVs' characteristics (e.g., weight, rolling resistance, and drag coefficient) and the road characteristics (e.g., grade, length, and traffic). From this model, an energy graph can be constructed to find the minimum energy consumption path between two locations using classic algorithms (e.g., Dijkstra [8] and Johnson [9]).

On the other hand, in [7, 10], the formulated EVRP consists of finding a set of minimum consumption routes to satisfy the demand of the users. Furthermore, some additional restrictions are included to consider the battery capacity of the BEVs. In [11], the vehicle routing problem is studied for the case of alternative fueled vehicles, considering their fueling time. In [12], a realistic energy consumption model is used for a vehicle routing problem in which the fleet includes both BEVs and diesel vehicles. In [13], the EVRP considers a heterogeneous fleet composition. In [14], an electric traveling salesman problem is studied. In [15], exact algorithms are presented for different charging conditions that generate variants of the EVRP. In [16, 17], the EVRP model is formulated considering the BEVs' recharge time and a set of recharge stations. Nonetheless, the battery degradation costs are not considered.

Other works such as [18–21] discuss the recharge control to minimize the energy recharge cost. Some papers [19, 21] study the recharge control for large fleets of BEVs. The control strategy proposed in those works minimizes the costs of energy generation and the BEVs fleet recharge. This strategy is not centralized, allowing the vehicle's autonomy preservation; nevertheless, these articles do not consider the energy consumed by each BEV nor the optimal recharge profiles obtained could increase the battery degradation.

In [22–24], another method to obtain recharge profiles for BEVs is discussed. In these studies, an optimization problem is formulated to minimize power losses and maximize the load factor. The formulation considers the network topology characteristics. Thereby, the obtained charge profiles contain information about the schedule and the vehicle that needs to be recharged, depending on the network connection point, whereby, the vehicles connected in the locations that induce higher losses are recharged in off-peak hours. However, these studies do not consider the effect of the recharge profile on the battery life nor the recharge cost in the objective function.

In [25, 26], the battery SOH is studied through a degradation model for the lithium-ion batteries. The model considers the main features that influence battery degradation such as the mean state of charge (SOC), depth of discharge (DOD), and battery temperature. In [25], some models to minimize the battery degradation and the cost of energy recharge are proposed. The recharge profile obtained is considerably different to those obtained in [19, 21], where it is calculated considering only the recharge cost.

In [27, 28], the EVRP is combined with infrastructure problems. In [27], the combination of the EVRP and the location-routing problem is presented, including the determination of battery swap stations and also the possibility of intermediate stops. In [28], different scenarios for the use of artificial intelligence for the management of BEVs are presented; those scenarios include issues for the grid-to-vehicle interaction as load balancing, energy pricing, placing of recharging places, and rerouting of BEVs to charge points.

This work proposes a methodology that relates the three challenges presented before to determine: minimum consumption paths, a set of optimal routes, the route assignment for each vehicle, and the recharge scheduling for electric vehicles for public transportation. A centralized controller based on a program that minimizes the costs of electric vehicles' operation is proposed. The objective function to minimize considers both the recharge cost and the cost associated with the battery degradation caused by route assignation and recharge cycles.

As a case study, an airport shuttle service attended by a fleet of BEVs is used to analyze the EVRP, while considering the topics exposed previously. This service consists of transporting passengers from a hotel to a nearby airport. Comparative results between the recharge scheduling obtained with a reduced order model and real energy consumption data of an electric vehicle are presented. Finally, findings on the BEVs' charge patterns, routing, and operational costs are presented. This information could be useful for public transportation companies interested in upgrading their fleet to BEVs.

This paper is organized as follows. Section 2 presents the energy consumption and optimal routing model; the energy consumption of the BEVs is estimated using a model based on the longitudinal dynamics equation of motion. Section 3 introduces the recharge scheduling problem, comprised of a route assignation and the charge model. Section 4 describes the battery degradation model. Section 5 presents an evolutionary algorithm to solve the proposed EVRP. Section 6 describes the case study and presents simulation results. Section 7 presents the conclusion.

2. Optimal Routing for BEVs

The considered scenario takes into account the presence of an operation center (OC), which coordinates both the transport

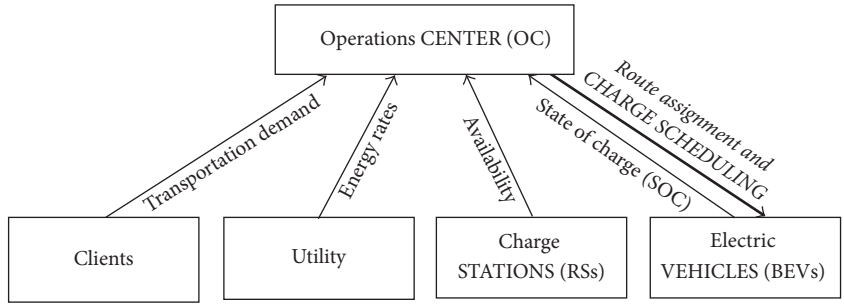


Figure 1: Operation center scheme.

service and the charge scheduling. The transport service consists of a BEV fleet that offers an airport shuttle service.

The charge scheduling seeks to minimize the cost; for this reason, the OC assigns routes to the BEVs that satisfy the transportation demand and establishes the charge schedules in the available charge stations (RSs) according to the energy rate. To coordinate the operation of the BEV fleet, the OC can communicate with BEVs, clients (i.e., potential passengers), RSs, and the public utility as presented in Figure 1. In this way, the OC receives information regarding requests from customers one day in advance, availability of RSs, state of charge of BEVs, and energy rates from the public utility. The route assignment and the charge scheduling are calculated and transmitted to the electric vehicles.

In the next subsections, the models of the systems, in which the OC is based, are described. These models are presented in the following order: estimation of energy consumption on the road, determination of optimal routes, assignment of routes, scheduling of a specific charge problem, and the solution method based on differential evolution.

2.1. Energy Consumption on the Road. For a BEV, the energy consumption on the road is sensitive to environment, road, and vehicle characteristics. Factors as the road grade and the travel speed can significantly influence the energy consumption. Additionally, the traffic conditions and other environmental factors can cause acceleration/deceleration rates that have an impact on the energy consumption of the vehicle. Consequently, a model that considers a vehicle moving at a constant speed is not sufficient.

The energy consumption model presented in this section is based on the longitudinal dynamics equations of motion presented in [29]. This model allows determining the instantaneous power consumption of the vehicle during its operation, considering the resistive loads imposed on the vehicle (i.e., aerodynamic drag force, rolling resistance force due to contact between tires and road, and longitudinal component of the gravitational force due to driving on hilly roads). For the implementation, the rolling resistance force was modeled by considering a constant rolling coefficient; hence, the rolling resistance force changes linearly with the changes of the normal force between the vehicle and the road [29].

The energy consumption of the vehicle is determined by integrating the instantaneous power consumption of the vehicle, which is a function of the environment, road, and vehicle characteristics. For the total energy computation, the energy consumed by a BEV while accelerating or in motion with a constant speed is considered; the energy recovered by the system with regenerative braking is not considered.

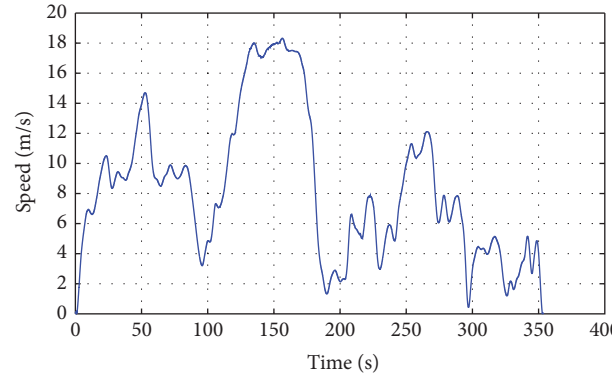


Figure 2: Example of a speed profile for a route.

The road characteristics are modeled in a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The vertices (i.e., key points or nodes) $v \in \mathcal{V}$ represent the points of special interest on the street maps. The edges $\varepsilon_{ij} \in \mathcal{E}$, where $\varepsilon_{ij} = (v_i, v_j)$, represent road sections between key points. The model assumes that, for each edge, the representative driving pattern associated with the given edge is known and it is represented by its speed profile (see Figure 2). Hence, there is a function $s : \mathcal{E} \to \kappa(\mathbb{R})$, where $s_{ij} = s(\varepsilon_{ij})$ corresponds to the representative speed profile associated with the road section that connects the vertices v_i and v_j, where $\kappa(\mathbb{R})$ is the space of continuous real functions with compact support.

The model also assumes that the key point elevation $z : \mathcal{V} \to \mathbb{R}$ is known. On the other hand, (1) presents the terms involved into the power consumption of the vehicle:

$$P_{i,j}(t) = \frac{P_{\text{aerodynamic}}(t) + P_{\text{rolling}}(t) + P_{\text{specific}}(t) + P_{\text{gravitational}}(t)}{\widehat{\eta}}. \quad (1)$$

The first term on the right side of (1) corresponds to the power dissipated by aerodynamic drag. The second term corresponds to the rolling resistance between tires and asphalt. The third term stands for the specific power required to overcome

the vehicle inertia, and the fourth term corresponds to the power consumed/gained due to the height of the vehicle originated by the road grade, represented by angle γ. The vehicle characteristics that influence the power consumption are mass (m), frontal area (A), drag and rolling resistance coefficients (C_D, f_r), and the powertrain efficiency ($\widehat{\eta}$). Given these characteristics, the power consumption at instant t of the trip between vertex i and vertex j is given by

$$P_{i,j}(t) = \frac{(1/2)\,\rho A C_D \left[s_{ij}(t)\right]^3 + mgf_r s_{ij}(t) + m\left(ds_{ij}/dt\right)(t)\, s_{ij}(t) + mg\tan(\gamma)\, s_{ij}(t)}{\widehat{\eta}}. \tag{2}$$

The energy consumption can be calculated by integrating (2) over the total time of each displacement between vertices, to obtain the energy consumption associated with the trip. Under the assumption of constant powertrain efficiency, the gravitational component can be analytically integrated. This leads to the following expression:

$$c_{i,j} = \int_0^{t_f} \left(\frac{(1/2)\,\rho A C_D \left[s_{ij}(t)\right]^3 + mgf_r s_{ij}(t) + m\left(ds_{ij}/dt\right)(t)\, s_{ij}(t)}{\widehat{\eta}} \right) dt - \frac{mg\left[z\left(v_j\right) - z\left(v_i\right)\right]}{\widehat{\eta}}. \tag{3}$$

Finally, energy consumption can be computed for an arbitrary path composed of different edges of the graph. A path P_h is defined as a sequence of l vertices $(v_1, v_2, \ldots, v_l)$ with $(v_i, v_{i+1}) \in \mathscr{E}$ for $i = 1, \ldots, l-1$. The energy necessary to travel along the path P_h in the road network is the sum of the energy consumed to complete each one of the road sections that compose the path as follows:

$$C\left(P_h\right) = \sum_{i=1}^{l-1} c_{i,j}. \tag{4}$$

2.2. Determination of Routes. Similar to the classical vehicle routing problem formulated in [30], an EVRP is proposed to model the airport shuttle service. It consists of searching a set of minimum consumption routes that satisfy the transportation demand and the operational constraints. Figure 1 presents the symbols used to explain the routing model.

The EVRP model is formulated on the energy graph $\mathscr{G}_s = (\mathscr{V}_s, \mathscr{E}_s)$, which is a simplification of the road network. The graph nodes are composed of $\mathscr{V}_s = \{v_{d_1}, v_{d_2}\} \cup C \cup R_s$, where C is the set of nodes within a transportation demand, and R_s is the set of RSs. The expression $2 \cdot |C|$ is the number of requests made by customers, where the operator $|\cdot|$ represents the set module. The depot nodes are denoted by v_{d_1} and v_{d_2}, which represent the route's starting and ending nodes. The graph edges, defined by $\mathscr{E}_s = \{(v_i, v_j) \mid v_i, v_j \in \mathscr{V}_S\}$, where $(v_i, v_j) \in \mathscr{E}_s$, are paths of minimum consumption that can be found through routing algorithms applied to the road network model. Therefore, the transportation demand implies m requests to pick up passengers and m requests to drop off passengers, each one with its associated number of passengers. The requests are identified by two nodes i and $m+i$, corresponding to the pick-up and drop-off stops, respectively. The set of pick-up nodes is denoted by $P = \{1, \ldots, m\}$, and the set of drop-off nodes is denoted by $D = \{m+1, \ldots, 2m\}$. Therefore, it is possible to define $C = P \cup D$ and if the request i consists of transporting q_i passengers from i to $m+i$, then $q_{(m+i)} = -q_i$.

The energy consumed during the trip from i to j is given by c_{ij}, and the traveling time is t_{ij}, where $i, j \in \mathscr{V}_s$. Each customer $i \in C$ has a time window $[a_i, b_i]$ in which the service must take place. Electric vehicles have a maximum passenger capacity Q and a maximum battery capacity B. The objective function minimizes the energy consumption for all routes H, meaning

$$\min \sum_{h \in H} \sum_{(i,j) \in \mathscr{E}_s} c_{ij} x_{ij}^h, \tag{5}$$

where $h \in H$ is the set of routes to satisfy all the transportation demand and x_{ij}^h are the flow variables, which are equal to 1 if the arc (i, j) is used on route h, or 0 otherwise. In addition, two constraints are defined to guarantee that all the passenger demands are satisfied, which are

$$\sum_{h \in H} \sum_{j \in \mathscr{V}_s \setminus v_{d_1}} x_{ij}^h = 1 \quad \forall i \in P, \tag{6}$$

$$\sum_{i \in \mathscr{V}_s \setminus v_{d_2}} x_{ij}^h - \sum_{i \in \mathscr{V}_s \setminus v_{d_1}} x_{j,m+1}^h = 0 \quad \forall h \in H,\ j \in \mathscr{V}_s \setminus \left\{v_{d_1}, v_{d_2}\right\}. \tag{7}$$

Equation (6) guarantees that each node with passenger demand is attended or visited by one route. Moreover, (7) forces passengers picked up at node i to be dropped off at node $m+i$. Three constraints are also defined to satisfy the flow through vertices; these are

$$\sum_{i \in \mathscr{V}_s \setminus v_{d_2}} x_{ij}^h - \sum_{i \in \mathscr{V}_s \setminus v_{d_1}} x_{ji}^h = 0 \quad \forall h \in H,\ j \in \mathscr{V}_s \setminus \left\{v_{d_1}, v_{d_2}\right\},\ i \neq j, \tag{8}$$

$$\sum_{j \in \mathscr{V}_s \setminus v_{d_2}} x^h_{v_{d_1}, j} \leq 1, \quad \forall h \in H, \tag{9}$$

$$\sum_{i \in \mathscr{V}_s \setminus v_{d_2}} x^h_{i, v_{d_2}} \leq 1, \quad \forall h \in H. \tag{10}$$

Equation (8) indicates that if a route h arrives at node j, then the same route must depart from node j, keeping the flow equilibrium. Also, (9) indicates that a route h can exit one time of the departing node v_{d_1}. On the other hand, (10) indicates that route h can enter once to the arriving node v_{d_2}. Furthermore, three sets of constraints are defined: (i) time constraints; (ii) capacity constraints; and (iii) energy constraints. The capacity constraints are

$$y^h_j \geq y^h_i + q_j - Q + Q x^h_{ij}, \quad \forall h \in H,\ i, j \in P,\ i \neq j, \tag{11}$$

$$q_j \leq y^h_j \leq Q, \quad \forall h \in H,\ j \in \mathscr{V}_s, \tag{12}$$

$$0 \leq y^h_{j+m} \leq Q - q_j, \quad \forall h \in H,\ j \in P, \tag{13}$$

where y^h_j is the capacity variable representing the amount of passengers picked up or dropped off during a trip along route h to node j. Therefore, (11) counts the number of passengers traveling along route h, with constraints (12) and (13) guaranteeing that the capacity variable does not exceed the maximum capacity Q. The time constraints are

$$w^h_j \geq w^h_i + t_{ij} - M + M x^h_{ij}, \quad \forall h \in H,\ i, j \in \mathscr{V}_s,\ i \neq j, \tag{14}$$

$$a_j \leq w^h_j \leq b_j, \quad \forall h \in H,\ j \in \mathscr{V}_s, \tag{15}$$

where w^h_i is the time variable specifying the instant at which passenger i is picked up or dropped off and M is a constant higher than any value of w^h_i. Therefore, (14) counts the elapsed time until passenger i is attended by route h. In addition, the constraint in (15) guarantees that passenger i is picked up or dropped off within the time window. Finally, the energy constraints are

$$e^h_j \leq e^h_i - c_{ij} x^h_{ij} + B - B x^h_{ij}, \quad \forall h \in H,\ i, j \in \mathscr{V}_s,\ i \neq j, \tag{16}$$

$$e_{\min} \leq e^h_j, \quad \forall h \in H,\ j \in \mathscr{V}_s, \tag{17}$$

where e^h_i is the energy variable specifying the remaining battery charge level when route h arrives to vertex i. Additionally, (16) allows computing the battery level based on the traveled vertices sequence. Moreover, (17) guarantees that at the end of route h the battery level will never drop below the minimum level $e_{\min}$.

The solution of proposed EVPR model is a set of optimal routes, H, which minimizes the energy consumed while satisfying the passenger demand, indirectly reducing the charge costs.

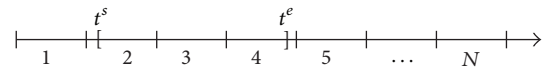


Figure 3: Timeline in the programming horizon.

3. Scheduling of Charge Problem

In this section, the scheduling of charge for BEVs is presented. First, the assignment of routes for the BEV fleet is defined. Then, the model for the charge control is established. Finally, the problem formulation is presented.

3.1. Assignment of Routes. The assignment of routes consists of the distribution of all the optimal routes S within the BEV fleet. It is considered that the assignment of a set of routes in the vehicle fleet, K, is performed in a programming horizon N. The programming horizon is illustrated in Figure 3.

The assignment of one route to a specific BEV is denoted by a^k_s, with $k \in K$, a binary variable, which is equal to 1 if route s is assigned to an EV, or 0 otherwise. A binary variable for "unavailability" is defined. This variable indicates that some periods of $i \in N$ are used to travel along route s assigned to a BEV. The unavailability variable is defined as

$$d^k_s(i) = \begin{cases} 1, & \text{if } t^s_s \leq i \leq t^e_s, \\ 0, & \text{otherwise}, \end{cases} \tag{18}$$

$$d^k_{s1}(i) + d^k_{s2}(i) \leq 1, \quad \forall i \in N,\ \forall k \in K,\ s_1, s_2 \in S, \tag{19}$$

where $[t^s_s, t^s_e]$ is the interval of duration of each route s, with starting time t^s_s and ending time t^s_e. This way, variable $d^k_s(i)$ allows locating each route s on the horizon N, while (19) guarantees that two different routes (s_i, s_2), with similar duration intervals, are not assigned at the same time to the same BEV. On the other hand, the energy consumed by the BEV k in route s for interval i of the horizon N is defined as $e^k_s(i)$.

Finally, an assignation profile is defined with vector $\aleph = \{a^k_s : \sum_{k \in K} \sum_{s \in S} a^k_s = |S|\}$, where the double sum guarantees that all the routes of S are assigned to the electric vehicles composing the fleet. Considering this, an assignation profile is valid if the constraint on (19) is satisfied for all the BEVs in the fleet.

3.2. Model for Charge Control. A charge control is proposed for a BEV fleet K, over a programming horizon N, based on the work presented in [19]. The SOC is defined for each vehicle $k \in K$ at instant $i \in N$ as $\mathrm{SOC}^k(i)$, with the following considerations:

$$\mathrm{SOC}^k(i) = \mathrm{SOC}^k(i-1) + \sum_{z \in R} \left[\eta_z r_z u^k_z(i) - \eta^d_z r^d_z v^k_z(i) - e^k_z(i)\right] - \sum_{s \in S} e^k_s(i), \quad \forall i \in N,\ \forall k \in K, \tag{20}$$

where $u_z^k(i)$ is a binary variable, equal to 1 when the vehicle k performs a charge action at instant i, r_z is the charge rate, and η_z is the efficiency of the charge process; $v_z^k(i)$ is a binary variable, which allows us to integrate the vehicle to building (V2B) energy discharge action, this variable is equal to 1 when the vehicle k performs a discharge action at instant i, r_z^d is the discharge rate, and η_z^d is the efficiency of the discharge process. Subscript $z \in R$ indicates the place where the charge or discharge is performed. Variable $e_z^k(i)$ indicates the energy consumed during the displacement towards a RS z, explained in the next section. Also, a restriction for SOC of the vehicles must be considered as

$$\mathrm{SOC}_{\min} \leq \mathrm{SOC}^k(i) \leq \mathrm{SOC}_{\max} \quad i \in N,\ \forall k \in K, \tag{21}$$

where $\mathrm{SOC}_{\min}$ and $\mathrm{SOC}_{\max}$ are the lower and upper bounds of SOC, respectively. Both bounds can be established according to the battery manufacturer recommendation, considering the battery technology. For this study, the initial and final SOC are equal to the $\mathrm{SOC}_{\max}$ value to guarantee that the vehicle begins and ends in the programming horizon with the same battery energy level, meaning

$$\begin{aligned} \mathrm{SOC}^k(0) &= \mathrm{SOC}_{\max}, \\ \mathrm{SOC}^k(N) &= \mathrm{SOC}_{\max}, \\ &\forall k \in K, \end{aligned} \tag{22}$$

where $\mathrm{SOC}^k(0)$ and $\mathrm{SOC}^k(N)$ are the initial and final states of charge of a BEV k. Finally, a charge profile for a vehicle k is defined as a set of charge actions $u_z^k(i)$. A charge profile for the BEV k is denoted $\mathbf{u}^k$ and defined as $\mathbf{u}^k = \{u_z^k(i) : \eta_z r_z \sum_{i \in N} u_z^k(i) = \sum_{i \in N}(e_s^k(i) + e_z^k(i))\}$, where the sum of all the charge actions must be equal to the energy consumed during all the trips. The profile $\mathbf{u}^k$ is valid if the constraints on (20) to (22) are satisfied.

3.3. Reroute of Charge Stations. In the model for determining routes, it is assumed that when a vehicle k is not traveling along any of them, then the vehicle is parked on the departing/arriving node (v_{d_1}, v_{d_2}), where the vehicles can be charged in a private station denoted by $R_1 \in R$. It is also considered that the vehicle can be charged in public stations identified in the operation zones; these charge actions on public stations might be necessary when the vehicles do not have enough energy to return to the arriving/departing node or when the private stations are occupied by other vehicles.

When considering that the vehicles can be charged in public stations, it must also be considered that their trajectory has to be reprogrammed or their initially assigned route must be modified, which is known as rerouting. Rerouting is defined as the redirection of the vehicle towards a station to perform a charge action $u_z^k : z \neq R_1$ without altering the vehicle's destination. The redirecting or rerouting is denoted with subscript z in the charge action u_z^k. If subscript z indicates a place different than R_1, then BEV k must travel to another station $R_2 \in R$, to perform the charge action.

It must be considered that redirecting towards a station implies that the BEV must have available time to travel back and forth to the RS. Similarly, the BEV must have some time intervals available to charge the energy that it needs; this is assured using two constraints. The first constraint is

$$\begin{aligned} d_s^k(j) + u_y^k(j) &= 0 \\ i_z - \Delta i_z \leq j < i_z,\ \forall y \neq z,\ y \in R, \end{aligned} \tag{23}$$

where $i_z \in N$ is the interval at the beginning of the charge and u_z^k, Δi_z is the number of time intervals that a BEV k needs to travel from v_{d_1} to z. If the result of (23) is zero, the BEV k can be rerouted towards a public recharge station because it has enough time to perform the action. The second constraint is

$$\begin{aligned} d_s^k(j) + u_y^k(j) &= 0 \\ i_z + \Delta i_z \leq j < i_z + \Delta i_{u_z} + \Delta i_{v_{d_1}},\ \forall y \neq z, \end{aligned} \tag{24}$$

where Δi_{u_z} is the number of time intervals in which the charge action needs to be completed, and $\Delta i_{v_{d_1}}$ is the number of time intervals that a BEV k needs to travel from z to v_{d_1}. Equations (23) and (24) guarantee that BEV k can be rerouted when there is enough time available. On the other hand, the energy consumed in the rerouted trip is denoted as e_z^k, which corresponds to the energy consumed traveling from v_{d_1} to z and back. This is considered in (20) assuring that the vehicle has the energy necessary to perform the displacement needed when redirected.

3.4. Optimization Problem of Charge for BEVs. The proposed optimization problem seeks to minimize the operation cost of the electric vehicle fleet, defined as

$$\min \sum_{z \in R} \sum_{k \in K} \sum_{i \in N} p_z(i)\, u_z^k(i) + \sum_{k \in K} c_{\deg}^k. \tag{25}$$

The first term is the charge energy cost for all the BEVs in the fleet, where $p_x(i)$ is the energy rate in the recharge station x at instant i; the last term of the objective function is the cost of battery degradation for all the BEVs in the fleet. The objective function in (25) is subjected to a valid assignment profile $\aleph$ and a valid charge profile $\mathbf{u}^k$ for all the BEVs $k \in K$. Besides, if there is redirecting towards public stations, then the constraints in (23) and (24) should be satisfied.

Likewise, a constraint related to the recharge stations' availability is defined as $\sum_{k \in K} u_z^k(i) \leq A_z(i)$, $\forall i \in N, z \in R$ to guarantee that the charge actions are programmed in available charge points, where $A_z(i)$ is the number of spaces available in recharge station z.

4. Battery Degradation Model

In this section, a simplified version of the lithium-ion battery degradation model proposed in [25] is presented.

The model used in this work estimates the cost of battery degradation $c_{\deg}$ in terms of the reduction of the lifespan of a battery, which is a function of three factors: temperature,

SOC, and DOD. These factors are related to the cost as follows:

$$c_{\text{deg}} = c_{\text{bat}} \left(L_{Q,T} + L_{Q,\text{SOC}} + L_{Q,\text{DOD}} \right), \tag{26}$$

where c_{bat} is the initial cost of the battery and $L_{Q,T}$, $L_{Q,\text{SOC}}$, and $L_{Q,\text{DOD}}$ are the percentage terms of battery degradation due to temperature, SOC, and DOD, respectively.

Each of the terms $L_{Q,T}$, $L_{Q,\text{SOC}}$, and $L_{Q,\text{DOD}}$ can be interpreted as a ratio $(\Delta L_x / L_x)$, where ΔL_x is the battery lifetime degradation due to a complete charge cycle during a day, and L_X is the total lifetime of the battery if the evaluated charge cycle is repeated until the end of the battery life, that is, when the energy capacity of the battery is lower than 80%.

The term $L_{Q,T}$ refers to the battery degradation due to temperature and charge time. This term is proportional to the charge power since a high operation temperature corresponds to a high charge power. This term is defined as

$$L_{Q,T} = \int_{t_{\text{ch}}} \frac{1}{n_{hy} l_y \left(T_{\text{amb}} + R_{\text{th}} \left| P_t \left(t \right) \right| \right)} dt + \frac{t_{\max} - t_{\text{ch}}}{n_{hy} l \left(T_{\text{amb}} \right)}, \tag{27}$$

where t_{ch} is the charge time in hours, $t_{\max}$ is the maximum time available to perform a charge in hours, and n_{hy} is the number of hours in a year. T_{amb} is the ambient temperature, established with a constant value of 25°C, and R_{th} is the thermal resistance of the battery, established with a constant value of 2°C/kW. P_t corresponds to the charge or discharge power in (kW), and $l_y(T)$ is a function that estimates the total number of years that the battery will last at a temperature T, defined as $l_y(T) = ae^{b/T}$, where a and b are the parameters of the model of a lithium battery fixed as $\mathbf{a} = 3.73 \times 10^{-4}$ and $\mathbf{b} = 636$ according to [25].

The term $L_{Q,\text{SOC}}$ describes the battery lifetime degradation due to average SOC, defined as

$$L_{Q,\text{SOC}} = \frac{m_a \text{SOC}_{\text{avg}} - d_a}{\text{CF}_{\max} y_p n_{hy}}, \tag{28}$$

where $m_a = 1.6 \times 10^{-5}$ and $d_a = 6.4 \times 10^{-6}$ are the parameters of the model for the lithium-ion batteries defined in [25], while $y_p = 15$ corresponds to the number of years which the battery is expected to last, and $\text{CF}_{\max} = 0.80$ represents the fading capacity at the end of the battery life. From (28), it can be noticed that $L_{Q,\text{SOC}}$ is proportional to SOC_{avg}; thus, a high SOC_{avg} reduces the useful life of the battery more than a moderated one.

The term $L_{Q,\text{DOD}}$ describes the degradation of the battery lifetime due to the mean DOD. This term is calculated using the energy performance concept [31]; hence, $L_{Q,\text{DOD}}$ is the ratio between the energy throughput used in a complete charge cycle and the energy throughput used during the battery lifetime, defined as

$$L_{Q,\text{DOD}} = \frac{B \sum_{i=1}^{N_c} \text{DOD}_i}{N_l \left(\text{DOD}_{\text{avg}} \right) * \text{DOD}_{\text{avg}} * B}, \tag{29}$$

where DOD_i is the ith subcycle in a complete charge cycle, N_c is the number of subcycles, DOD_{avg} is the average DOD cycle, and $N_l(\text{DOD})$ is the life time of the battery in cycles for a given DOD. This is defined as $N_l(\text{DOD}) = (\text{DOD}/145.71)^{-1/0.6844}$, which is fitted for lithium-ion battery technology.

Summarizing, the model has two attributes that allow its integration to the optimization problem. First, it allows computing the battery degradation cost of a BEV due to the actions of charge and discharge previously programmed. Second, the simplicity of the model allows performing the calculations in a fast way, reducing the use of computational resources, embedding it with the optimization problem and its solution method, explained in the next section.

5. Differential Evolution

A differential evolution (DE) algorithm is presented to solve the problem formulated in previous sections. The DE is a direct stochastic search algorithm based on the evolution of a population, which was developed by Storn et al. in 1996 [32, 33]. The DE operates through steps that are similar to those used in a standard *evolutionary algorithm* (EA).

The DE employs the differences between members of the population to explore the objective function. For this reason, it does not use a probability function to generate the offspring. Hence, the DE uses an approach that is less stochastic, making it more efficient for solving several kinds of problems [34].

In the next subsections, the format of a solution and the steps composing the DE algorithm are presented.

5.1. Format of a Solution. Before solving the problem, a possible solution must be represented in a format that the DE can optimize. For this case, the possible solution of the electric vehicle charge problem is represented by the valid assignment profile $\aleph$, and the collection of the charge profiles $\mathbf{u}^k$, associated with each vehicle. The assignment profile contains each route assignation variable a_k^h, $\forall k \in K, h \in H$. For each vehicle $k \in K$, the associated charge profile contains each charge action variable for that vehicle, $u_z^k(n)$, $\forall z \in R$, $n \in N$.

$$\begin{aligned} X^u = &\left\{ u_1^1(1), \ldots, u_1^1(N), \ldots, u_1^{|K|}(1), \ldots, u_1^{|K|}(N), \ldots, \right. \\ &\left. u_{|R|}^1(1), \ldots, u_{|R|}^1(N), \ldots, u_{|R|}^{|K|}(1), \ldots, u_{|R|}^{|K|}(N) \right\}, \\ X^a = &\left\{ a_1^1, \ldots, a_1^{|K|}, \ldots, a_{|H|}^1, \ldots, a_{|H|}^{|K|} \right\}, \end{aligned} \tag{30}$$

where X^u and X^a, maintaining the notation of [32], are denoted as vectors of charge and assignment parameters, respectively. Considering this, $X = X^u \cup X^a$ is defined as a feasible solution for the electric vehicle charge problem, denoted as the parameter vector X in the development of each step of the DE algorithm, which are described next.

5.2. Population Initialization. Similar to the EA, the DE steps are (i) population initialization; (ii) mutation; (iii) crossing over; and (iv) selection. A population is composed of a number N_p of vectors X, where each vector is known as an

individual. The following notation is adopted to represent a vector of parameters $i \in N_P$ of the population in the current generation G_e

$$X_{i,G_e} = \left[x_{1,i,G_e}, x_{2,i,G_e}, \ldots, x_{m_v,i,G_e}\right], \tag{31}$$

where $G_e = 0, 1, \ldots, G_{\max}$ denotes the future generations, $G_{\max}$ the last generation, and m_v the size or dimension of the parameter vector.

The population of vectors is produced through a random generator, where each vector must meet the constraints; otherwise, it will be rejected and will not be part of the population. This rejection step ensures the feasibility of the individuals of the population given that the constraints were defined specifically for that purpose. The size of the population N_P is considered as a control variable of the DE method. In this case, $N_P = 10m_v$ is used according to [34].

5.3. Mutation. The mutation allows generating a donor vector V_{i,G_e} by means of a differential operation. This operation is defined as

$$V_{i,G_e} = X_{r_1,G_e} \text{OR} \left(X_{r_2,G_e} \text{ XOR } X_{r_3,G_e}\right), \tag{32}$$

where X_{r_1,G_e}, X_{r_2,G_e}, and X_{r_3,G_e} are three individuals of the population generation G_e and the indices r_1, r_2, and r_3 are randomly selected from the range $[1, N_P]$. These indices must be mutually exclusive to diversify the results of the mutation operation.

The differential operation consists of adding the difference of two vectors to another vector of the population [33]. In this case, the differential operation is implemented with the exclusive disjunction (XOR). By using the logic operator XOR, the algorithm identifies the differences between two vectors. After that, the sum operation is implemented using the disjunction operator (OR), which permits combining the difference between the two vectors with another vector of the population.

5.4. Crossover. The crossover operation allows the exchange of components between the donor vector V_{i,G_e} and the target vector X_{i,G_e}. The result of the crossover is a trial vector $U_{i,G_e} = [u_{1,i,G_e}, u_{2,i,G_e}, \ldots, u_{m_v,i,G_e}]$. This operation is defined as

$$\begin{aligned} u_{j,i,G_e} &= v_{j,i,G_e} \quad \text{for } n_v + 1 \le j \le n_v + L - 1, \\ u_{j,i,G_e} &= x_{j,i,G_e} \quad \text{otherwise,} \end{aligned} \tag{33}$$

where n_v is randomly selected among $[1, m_v]$ with uniform probability and L is obtained from $[1, m_v]$ with probability equal to $C_r \in [0, 1]$. The variable $C_r \in [0, 1]$ is known as the crossover rate and constitutes a control variable of the DE method. C_r must be considerably lower than one (e.g., 0.3). If the convergence towards a solution is not achieved, then C_r can be chosen in the interval [0.5, 1] according to [34].

5.5. Selection. The selection operation allows choosing between the target vector X_{i,G_e} and the vector U_{i,G_e} (obtained in the crossover operation) to be part of the next generation $G_e + 1$. The selection operation is defined as

$$\begin{aligned} X_{i,G_e+1} &= U_{i,G_e} \quad \text{if } f\left(U_{i,G_e}\right) \le f\left(X_{i,G_e}\right), \\ X_{i,G_e+1} &= X_{i,G_e} \quad \text{otherwise,} \end{aligned} \tag{34}$$

where $f(\cdot)$ is the objective function presented in (25) to be minimized. This operation indicates that if the trial vector U_{i,G_e} has a value $f(\cdot)$ equal to or lower than the value $f(\cdot)$ of the target vector; then the trial vector will be part of the next generation; otherwise, the target vector X_{i,G_e} is retained in the population of generation $G_e + 1$. Therefore, the steps of mutation, crossover, and selection are cycled until the maximum generation $G_{\max}$ is reached. A summary of this process is presented in Algorithm 1.

5.6. Genetic Algorithm Benchmark. For the analysis of the performance of the DE algorithm, a standard Genetic Algorithm (GA) was used as a benchmark. The performance was analyzed in terms of convergence and computational cost. For the GA formulation, the chromosomes that describe each individual are binary strings, equal to those used in the DE.

The GA used has the typical stages of an evolutionary algorithm. The first stage is the generation of the initial population with the population size (P_G) as a parameter of adjustment. Population size was chosen as 10 times the quantity of chromosome's genes (N_G). The second stage is the selection by tournament: in this stage two individuals with the highest fitness are chosen from a randomly selected subpopulation. The third stage is crossover. A two-point crossover operation with a crossover probability of $P_c = 0.8$ was performed. The fourth stage is mutation, which is intended to maintain a portion of random search. Mutation provides the possibility of changing one of the genes of the chromosomes of the descendants. Mutation probability was chosen as $P_m = 0.01$. Finally, if the ending criterion (i.e., generation is equal to maximum generation, $G = G_{\max}$) is not reached, then a generational change is performed. In the generational change, the descendants obtained through the last stages replace their parents by initiating a new cycle. On the other hand, if the ending criterion is reached, the GA finishes. Figure 4 presents a block diagram of the GA used as a benchmark.

6. Case Study

The case study consists of an airport shuttle service using a BEV fleet. For this service, it is required to determine the route assignment and the charge scheduling in a coordinated manner. The location defined for this case study is El Dorado airport in Bogotá, Colombia. In this section, the case study is described in detail through four aspects: (i) the operation zone; (ii) the BEVs' characteristics and data logging; (iii) the data processing; and (iv) the demand for transportation.

6.1. Operation Zone. The operation zone used for this case study is defined by three nodes: El Dorado airport, defined as the north-western node; a mall located in the south node; and a public recharge station (RS) in the eastern node (see Figure 5). Within the area defined by the nodes, a total of 54

Inputs: Read values of the control parameters: C_r, N_P
Initialization of population ($\mathbf{P_{G_e}}$): $G_e = 0, P_G = \{X_{1,G_e}; \ldots; X_{N_P,G_e}\}$
for $G_e = 1$ to $G_{\max}$ **do**
  for $i = 1$ to N_P **do**
    Mutation: Generate a donor vector $V_{i,G}$
    $V_{i,G_e} = X_{r2,G_e}$ OR $(X_{r_2,G_e}$ XOR $X_{r_3,G_e})$
    Crossover: Generate a trial vector U_{i,G_e}
    $u_{j,i,G_e} = v_{j,i,G_e}$ for $n_v + 1 \leq j \leq n_v + L - 1$
    $u_{j,i,G_e} = x_{j,i,G_e}$ *otherwise*
    Selection: Evaluate the trial vector U_{i,G_e}
    $X_{i,G_e+1} = U_{i,G_e}$ if $f(U_{i,G_e}) \leq f(X_{i,G_e})$
    $X_{i,G_e+1} = X_{i,G_e}$ otherwise
  end for
end for

Algorithm 1: DE algorithm.

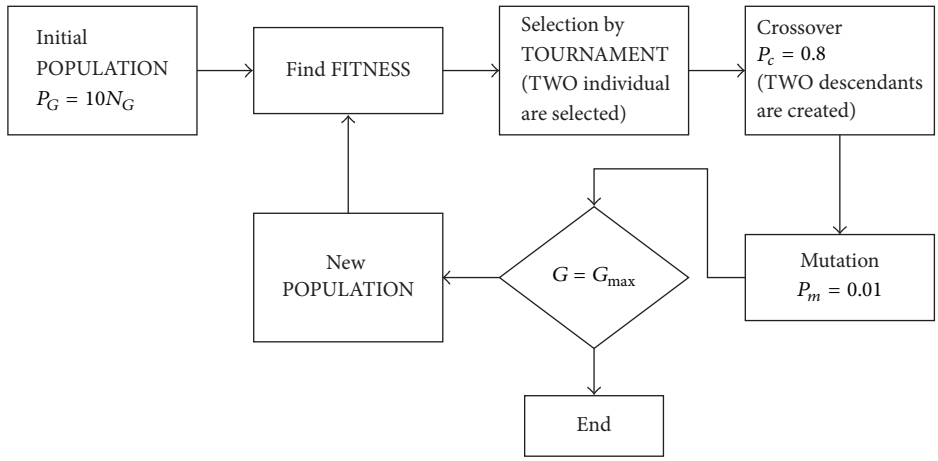


Figure 4: Block diagram of the GA used as a benchmark.

nodes are used as references to determine the routes. From these nodes, six are identified as main nodes and are used to build the graphics (see Figure 5 and Table 1). The main nodes are (1) airport whereabouts 1; (2) airport whereabouts 2; (3) hotel, operation center (OC), private RS with $r_2 = 3$ kW, $\eta_2 = 0.9$; (4) public RS with $r_3 = 6$ kW, $\eta_3 = 0.9$; (5) mall; and (6) bus terminal.

6.2. Test Vehicle Characteristics. The aim of the case study is to analyze the feasibility of the implementation of a fleet of BEVs for an airport shuttle service. In order to do this, the results of scheduling and routing are used.

The kinematical information about the displacements is needed to estimate the energy consumption of each path. This information is measured by using a different vehicle that has been selected to be as similar as possible to the BEV under study. The characteristics of the vehicle used in the case study are summarized in Table 2.

The most relevant characteristics considered to estimate the energy consumption properly are those related to resistive forces and traction acting on the vehicle. The characteristics include the dimensions (i.e., frontal area and drag coefficient) and weight to power relation of the vehicle. The test vehicle selected is a saloon type. It has been necessary to add weight to the vehicle to obtain a similar weight to power ratio.

6.3. Data Logging and Data Processing. The characteristics of the GPS unit used to measure and register the data are presented in Table 3. With the data obtained from the speed profile (see Figure 6), it has been determined that a signal conditioning stage must be implemented to handle signal noise and atypical points due to any interruptions of communication between GPS and satellites. These issues are originated by the buildings located near the streets and by bridges on the paths. Figure 6 presents an example of a lost-of-signal issue, which generates a signal spike on the raw speed profile.

The data processing considers the use of filters to deal with the data logging issues. Two filtering methods were considered to obtain a smooth speed profile: the Kalman filter described in [35] and the Savitzky-Golay filter described in [36, 37]. Figure 7 presents a comparison between the processed data obtained by using these filters. The result obtained by using each filter is compared with the original data. The goal is to remove the signal spikes originated by

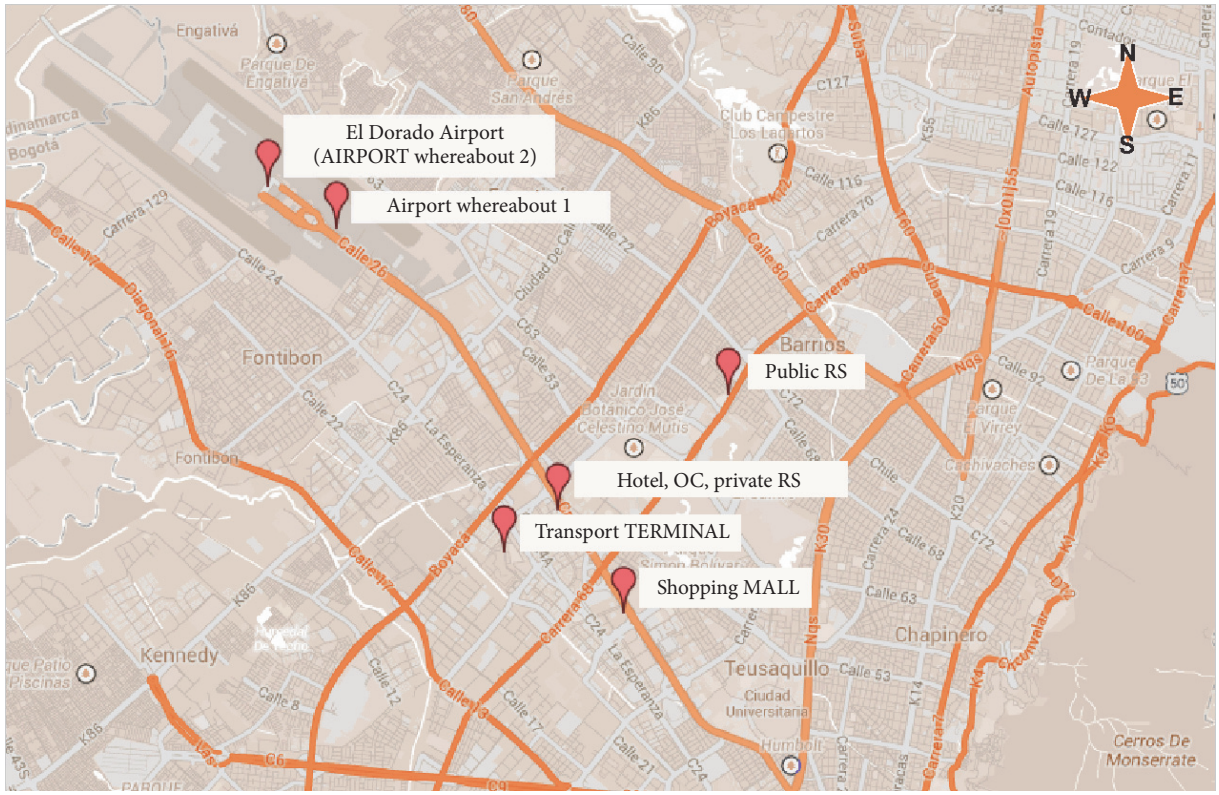


Figure 5: Main nodes of reference located in the operation zone.

Table 1: Distance matrix.

	Distance [km]					
Depart\arrive	Airport 1	Airport 2	Hotel	Public RS	Mall	Terminal
Airport 1	—	0.94	5.92	9.55	7.62	6.69
Airport 2	2.94	—	5.86	9.54	7.6	6.66
Hotel	6.2	7.11	—	5.5	1.79	3.33
Public RS	6.99	7.87	4.21	—	4.21	5.9
Mall	7.74	8.62	2.61	4.9	—	4.59
Terminal	7.48	8.37	2.1	6.03	3.25	—

Table 2: Characteristics of the test vehicle.

Parameter	Value
m [kg]	1312
A [m^2]	1.86
C_D	0.32
f_r	0.0117
$\widehat{\eta}$	0.9
Length [m]	4.32
Width [m]	1.69

Table 3: Characteristics of the GPS unit.

Parameter	Value
Sampling frequency [Hz]	100
Speed resolution [km/h]	0.01
Speed accuracy [km/h]	0.1
Distance resolution [m]	0.01
Distance accuracy [%]	0.05

the issues of the data logging process, keeping the rest of the signal unaltered. It can be observed that, for this case study, the Savitzky-Golay filter (Figure 7(a)) is more effective than the Kalman filter (Figure 7(b)).

6.4. Transport Demand. An illustrative transport demand is defined. It consists of a case where one passenger is departing from the airport whereabouts 1 and two passengers departing from the airport whereabouts 2. This scenario is repeated every hour from 8:00 to 16:00. Each passenger must pay a 5 USD ticket and has to be picked up within a time window of

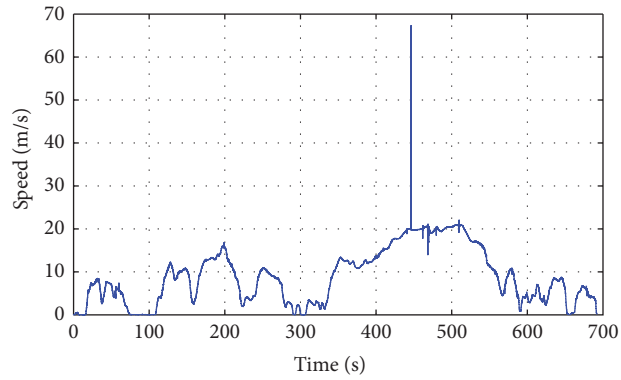


Figure 6: Raw speed profile data.

30 minutes. It is assumed that the passengers may be traveling to the mall, the terminal, or the hotel according to their needs.

6.5. Energy Consumption. The results of energy consumed by the vehicle during a trip along each path are calculated; this is achieved by the introduction of the filtered speed profile (see Figure 7) in the longitudinal dynamics model described in Section 2.1. The results are presented in the energy matrix shown in Table 4.

6.6. Simulation Results. This section presents the results of the implementation of the proposed methodology for the solution of the case study. The comparison of the proposed DE metaheuristic with a Genetic Algorithm (GA) is also presented.

The routing problem proposed is solved using the optimization tool XPRESS. The results show that nine routes must be used to satisfy the total transport demand of the case study. Each route is optimal and has an energy consumption of approximately 1.66 kWh. Traveling along each route takes less than one hour. The routes must be followed on an hourly basis during the period from 07:30 to 15:30 according to the transportation demand. The charge and route assignment actions are planned with a 24-hour programming horizon starting at 7:00. The programming horizon is discretized in half hour intervals. Four scenarios (SC) are considered with different working conditions to illustrate possible circumstances.

In SC1 the charge is allowed only in the private station (RS2) with a two-level energy rate: 0.20 USD/kWh between [7:00–22:00) and 0.15 USD/kWh between [22:00–7:00). For SC1 the battery degradation cost is not considered. In SC2 the charge is also allowed only in the private station (RS2) with the same two levels energy rate of SC1, and the battery degradation cost is considered.

In SC 3, the charge is allowed in both the private station (RS2) and the public station (RS3). In SC3, the energy rate of RS2 is the same one used in SC1 and RS3 has a two-level energy rate: 0.10 USD/kWh between [7:00–22:00) and 0.05 USD/kWh between [22:00–7:00). In SC3, the battery degradation cost is considered, and the V2B operation is allowed (i.e., energy discharge of BEVs towards the hotel). In SC4, the conditions are similar to those in SC3 except for the difference in energy rate, where both RS2 and RS3 have single-level energy rates: 0.50 USD/kWh for RS2 and 0.05 USD/kWh for RS3. In SC4, the energy rate for the private station is artificially incremented to investigate a scenario where V2B operations are intuitively profitable.

The results obtained with the scenarios previously described are presented in Figures 8 and 9. For each scenario, the rate evolution and the SOC for two vehicles of the fleet and their charge actions on each station are presented across the programming horizon.

Figure 8(a) presents the charge and route assignment obtained for SC1 using DE. It is observed that the charge actions were programmed at the end of the horizon, taking advantage of the low energy rate during the night.

Figure 8(b) presents the charge and route assignment obtained for SC2 using DE. It is found that the SOC of the BEVs describes a travel-charge pattern, meaning that the BEVs travel along one or two routes and immediately have to charge. When analyzing this behavior, it is found that the travel-charge pattern reduces the battery degradation cost by maintaining a low DOD. This is consistent with the battery model since the cycles with a large DOD are the main source of battery degradation.

SC1 and SC2 were used to compare the impact of scenario conditions on battery lifespan. When the charge and road assignments obtained for SC1 are implemented, the expected battery lifespan is approximately 6400 cycles. On the other hand, when the SC2 assignments are used, the expected battery lifespan is approximately 8400 cycles. This represents a difference of 2000 cycles, which is comparable with five years of operation of a BEV in transportation service.

Figure 9(a) presents the charge and route scheduling for SC3 obtained with DE. It was obtained that the algorithm does not schedule the V2B operation although the energy discharges are allowed. It is observed that the energy rate for SC3 is not profitable when performing the energy discharge actions because the economic benefit to afford a V2B operation is surpassed by the battery degradation cost.

Figure 9(b) presents the charge and route scheduling for SC4 obtained with DE. It was obtained that several charge actions in RS3 are programmed as well as the operation V2B through discharge actions. This behavior can be explained due to the high difference in the energy rate in the recharge stations RS2 and RS3. For this reason, the V2B operation becomes profitable. However, this difference in rates is not easily accomplished in real scenarios.

6.7. Benchmark Results. To analyze the convergence of the solution and the computational cost of the DE algorithm, a standard GA algorithm was used as a benchmark, as described in Section 5.6. The algorithms were compared under the conditions of scenarios SC1 and SC2. Given the fact that both algorithms depend on pseudorandom variables, several iterations were considered. The results are presented indicating the average values and the standard deviation obtained for each observed variable.

Table 5 presents the results obtained for the comparison under SC1 conditions. Both DE and GA converged to the same value of the daily recharge cost of a vehicle (i.e., 1.42 USD). The recharge schedules obtained were essentially

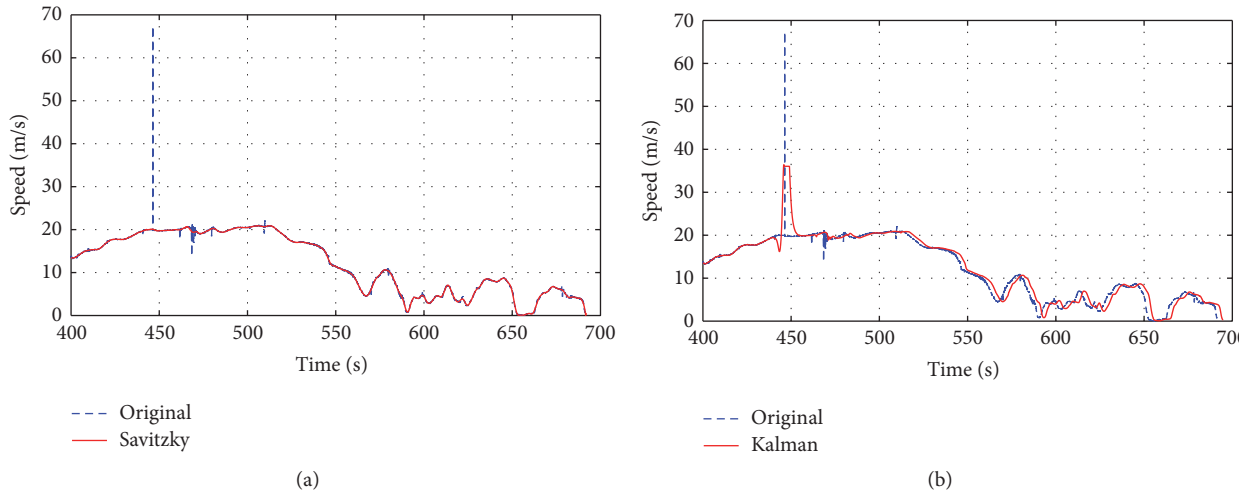


Figure 7: Comparison between filtered data. Savitzky-Golay filter (a). Kalman filter (b).

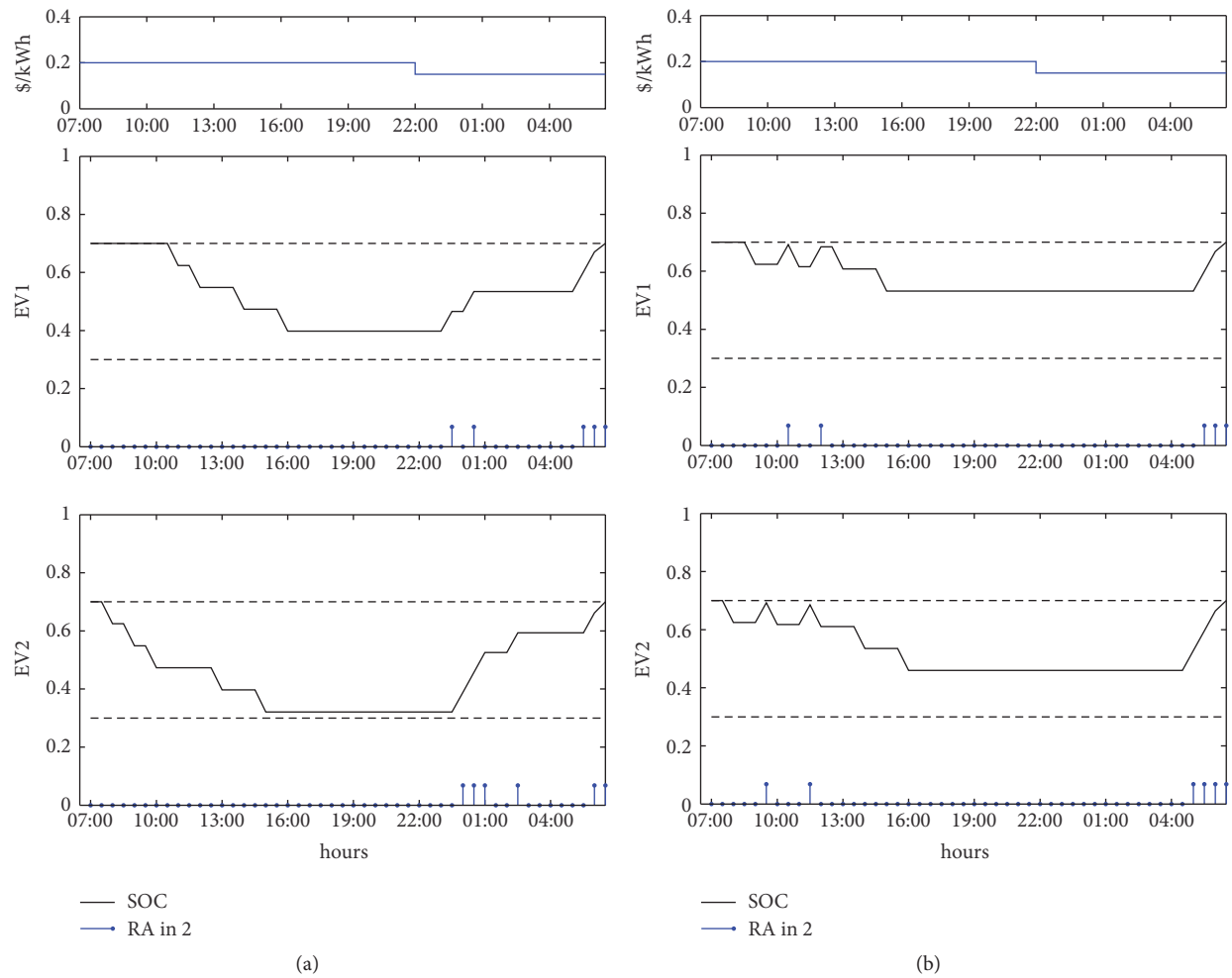


Figure 8: Charge scheduling for two BEVs operating in SC 1 (a) and SC2 (b).

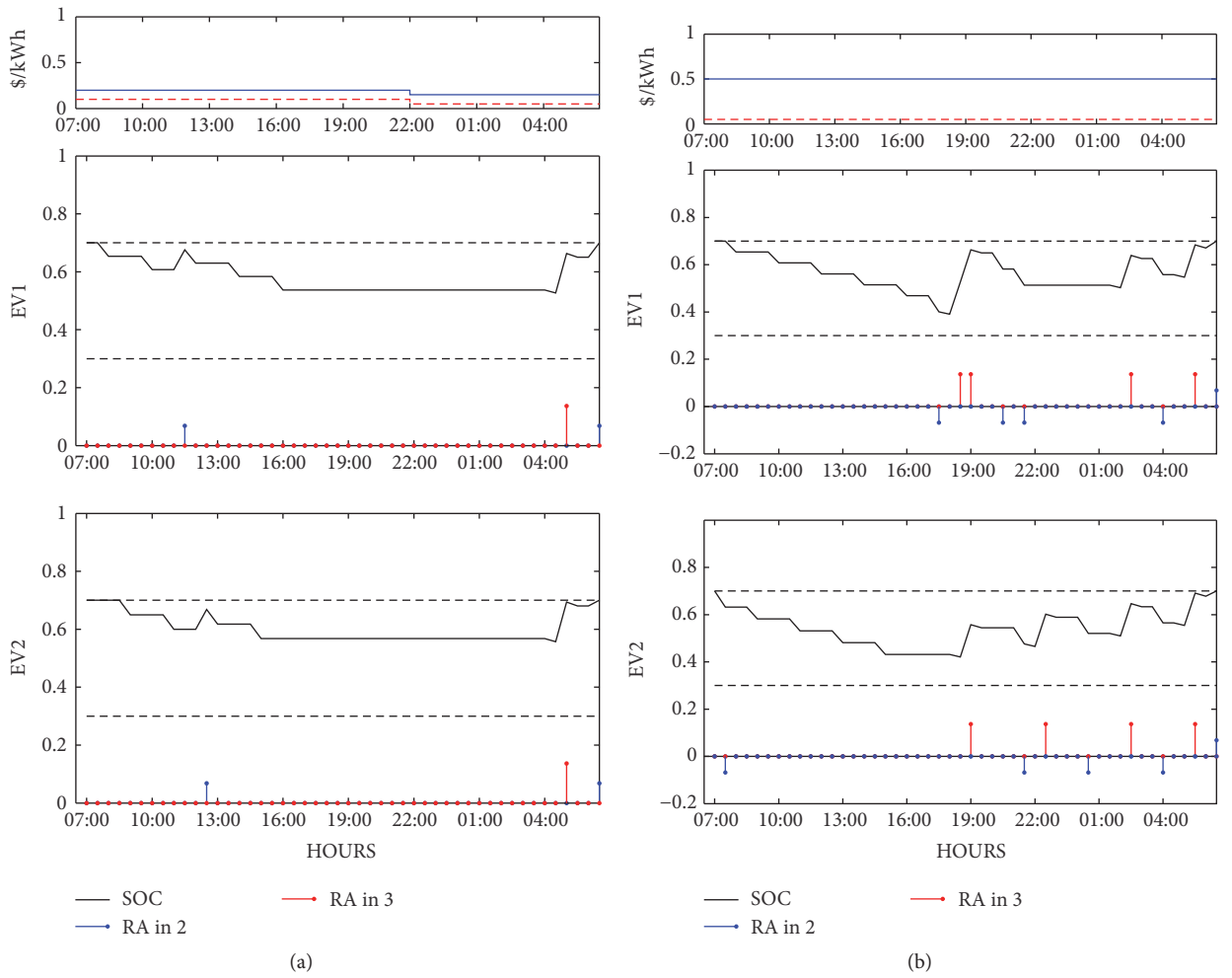


Figure 9: Charge scheduling for two BEVs operating in SC3 (a) and SC4 (b).

Table 4: Energy consumption matrix.

	Energy consumption [kWh]					
Depart\arrive	Airport 1	Airport 2	Hotel	Public RS	Mall	Terminal
Airport 1	—	0.128	0.574	1.073	0.79	0.719
Airport 2	0.285	—	0.567	1.076	0.634	0.656
Hotel	0.593	0.773	—	0.658	0.183	0.429
Public RS	0.688	0.828	0.51	—	0.443	0.727
Mall	0.805	0.876	0.321	0.497	—	0.536
Terminal	0.839	0.925	0.28	0.72	0.407	—

the same. Consequently, the recharging schedules produce a similar battery lifespan for both cases: 6434 cycles for DE and 6322 cycles for GA. Regarding the computational cost, the simulation times obtained were 14.57 for the DE and 17.91 for the GA. For SC1, the use of DE instead of GA led to a reduction of 18.6% in the computational cost. The computational cost difference obtained can be important when a large fleet is considered.

Figure 10 shows the convergence curves obtained with DE and GA. Both algorithms presented expected behaviors. The DE convergence curve is monotonously decreasing. The GA convergence curve presents some oscillations before converging. The advantage of DE is the selection process performed between generations. The choice of GA as benchmark takes advantage of its capability to reach global solutions. From this point of view, the results of the benchmark performed suggest that the DE is converging to the global minimum.

Table 6 presents the results obtained for the comparison under SC2 conditions. For this scenario, the total daily cost was computed. The total cost considers the recharging cost and the battery degradation cost. The difference between the average convergence values obtained with DE and GA for the

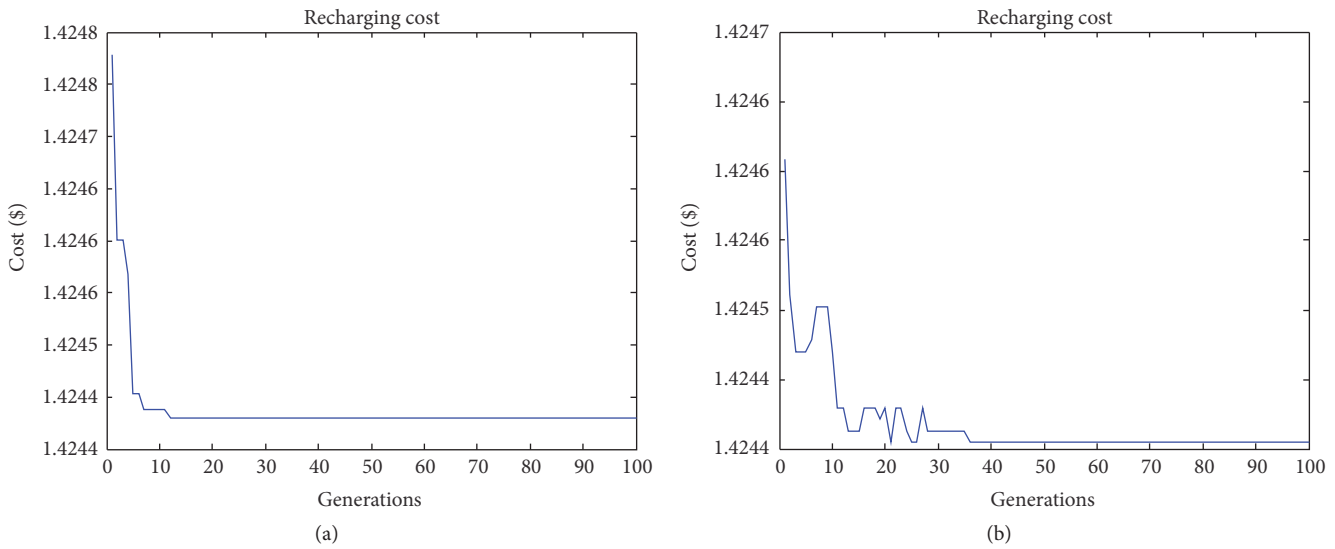


Figure 10: Convergence curves of DE (a) and GA (b) in SC1.

Table 5: Results of the simulation obtained with the DE and GA algorithms in SC1.

	Average (Std. Dev)	
	DE	GA
Recharge cost [USD]	1.42 (§)	1.42 (§)
Battery lifespan [cycles]	6434 (16)	6322 (137)
Simulation time [s]	14.57 (0.201)	17.91 (0.126)

§ represents a standard deviation under 0.01 USD.

Table 6: Results of the simulation obtained with the DE and GA algorithms in SC2.

	Average (Std. Dev)	
	DE	GA
Total cost [USD] (recharging + battery degradation)	5.21 (0.05)	5.29 (0.07)
Battery lifespan [cycles]	8368 (185)	8042 (281)
Simulation time [s]	15.13 (0.23)	17.42 (0.357)

total cost is 1.4%. The difference between the average convergence values obtained with DE and GA for the battery lifespan is 3.9%.

7. Conclusions and Future Research

A methodology to plan the energy charge and route assignation for a fleet of BEVs providing a passenger transport service has been presented. This methodology considers the search of optimal routes and the minimization of operational costs. It has been found that the charge scheduling and route assignment have effects on the battery lifetime. The result obtained for the charge scheduling allows increasing the battery lifespan. Additionally, it is observed that the consideration of battery degradation modifies the charge patterns.

In this work, some conditions to perform a V2B operation have been reported. These conditions are related to the battery technology, the battery degradation model, and the energy rate. It was found that batteries with lithium-ion technology studied in this work do not meet the requirements to provide the V2B operation as their degradation cost is high. Therefore, the V2B operation is profitable only for the BEV owner when the difference in energy rate between the recharge station and the discharge station is approximately 0.5 USD/kW. Nonetheless, this difference in energy rates is difficult to achieve in a real scenario.

For future work, both models for the estimation of the energy consumption and for battery degradation should be refined. The energy consumption model should consider regenerative braking. The battery degradation model should consider the detailed behavior of different types of batteries. Also, the performance of additional metaheuristics should be investigated.

Notations

Notation Used in the BEV Routing Problem

$\mathcal{G}_s$: Simplified energy graph
$\mathcal{V}_s$: Vertices or nodes of the energy graph $\mathcal{G}_s$
$\mathcal{E}_s$: Edges of the energy graph $\mathcal{G}_s$
v_{d_1}, v_{d_2}: Depot nodes
C: Set of customer nodes
R_s: Set of nodes on recharge stations
P: Set of pick-up nodes
D: Set of drop-off nodes
c_{ij}: Energy consumed by traveling from node i to node j
t_{ij}: Time elapsed during the trip from node i to node j
$[a_i, b_i]$: Time window to pick up a passenger at node i

q_i: Number of passengers at node i
Q: Maximum passenger capacity of the BEV
B: Battery capacity
M: Constant, higher than any value of w_j^h
$e_{\min}$: Lower bound of battery level
H: Set of routes
x_{ij}^h: Binary flow variable to specify that route h travels between nodes i and j
y_j^h: Number of passengers picked up or dropped off at node j when traveling on route h
w_j^h: Instant at which a passenger i has to be picked up or dropped off along route h
e_j^h: Remaining level of battery charge at the end of node i on route h.

Notation Used in the Description of the Charge Scheduling Problem

$\mathcal{N}$: Programming horizon
N: Final time slot of programming horizon
a_k^h: Route assignation variable
$d_h^k(n)$: Unavailability variable
t_h^s: Starting time for a trip along route h
t_h^e: Ending time for a trip along route h
$\aleph$: Assignation profile
$\mathrm{SOC}^k(n)$: State of charge of BEV k battery at instant n
η_z: Charge efficiency of a charge station z
r_z: Charge rate of charge station z
$u_z^k(n)$: Charge action variable of a BEV k at instant n in charge station z
$e_h^k(n)$: Variable of energy consumed by BEV k at instant n in route h
$\mathbf{u}^k$: Charge profile of BEV k
$p_z(n)$: Energy cost in charge station z at instant n
c_{deg}^k: Battery degradation cost of BEV k.

Notation Used in the Description of the Battery Degradation Model

c_{bat}: Battery cost
$L_{Q,T}$: Battery degradation due to temperature
$L_{Q,\mathrm{SOC}}$: Battery degradation due to state of charge
$L_{Q,\mathrm{DOD}}$: Battery degradation due to depth of discharge
n_{hy}: Number of hours in a year
$l_y(\cdot)$: Function of the lifespan of the battery in years
T_{amb}: Ambient temperature
R_{th}: Thermal resistance
$P_t(\cdot)$: Charge power
$t_{\max}$: Time available to perform a charge in hours
t_{ch}: Charge time in hours
$\mathrm{CF}_{\max}$: Battery charge capacity
y_p: Battery lifespan estimated in years
$N_l(\cdot)$: Battery lifespan estimated in cycles.

Notation Used in the Description of the Differential Evolution Algorithm

$\mathbf{X}^{\mathbf{U}}$: Vector containing the charge parameters or the information about the charge actions
$\mathbf{X}^{\mathbf{a}}$: Vector containing the route assignation parameters
X_{i,G_e}: Vector containing the parameters of subject i of generation G_e
V_{i,G_e}: Donor vector that contains the parameters of subject i of generation G_e
U_{i,G_e}: Trial vector that contains the parameters of subject i of generation G_e
N_p: Number of parameters of the vector
C_r: Crossover rate
m_v: Number of elements in a parameter vector.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by three sources: Project CIFI 2012, Facultad de Ingeniería Universidad de Los Andes; Codensa S.A. ESP through Silice III project, under Grant P12.245422.006.2012; and the Colombian Administrative Department of Science, Technology, and Innovation (Colciencias), Asoingeniería Ltda., and Universidad de Los Andes, under Grant 256-2013.

References

[1] A. Emadi, "Transportation 2.0," in *IEEE Power and Energy Magazine*, vol. 9, pp. 18–29, IEEE, 2011.

[2] C. Ozkurt, F. Camci, V. Atamuradov, and C. Odorry, "Integration of sampling based battery state of health estimation method in electric vehicles," *Applied Energy*, vol. 175, pp. 356–367, 2016.

[3] S. Vazquez, S. M. Lukic, E. Galvan, L. G. Franquelo, and J. M. Carrasco, "Energy storage systems for transport and grid applications," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 12, pp. 3881–3895, 2010.

[4] A. Millner, "Modeling lithium ion battery degradation in electric vehicles," in *Proceedings of the 2010 IEEE Conference on Innovative Technologies for an Efficient and Reliable Electricity Supply, CITRES 2010*, pp. 349–356, Waltham, MA, USA, September 2010.

[5] L. Gaines and R. Cuenca, "Cost of Lithium-Ion Batteries for Vehicles," Center for Transportation Research, Energy Systems Division, Center for Transportation Research, Energy Systems Division, 2000.

[6] M. Sachenbacher, M. Leucker, A. Artmeier, and J. Haselmayr, "Efficient Energy-Optimal Routing for Electric Vehicles," in *Proceedings of theTwenty-Fifth AAAI Conference on Artificial Intelligence*, pp. 1402–1407, 2011.

[7] N. Touati-Moungla and V. Jost, "Combinatorial optimization for electric vehicles management," *Journal of Energy and Power Engineering*, vol. 6, pp. 738–743, 2012.

[8] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[9] D. B. Johnson, "Efficient algorithms for shortest paths in sparse networks," *Journal of the ACM*, vol. 24, no. 1, pp. 1–13, 1977.

[10] A. Artmeier, J. Haselmayr, M. Leucker, and M. Sachenbacher, "The Optimal Routing Problem in the Context of Battery- Powered Electric Vehicles," in *proceedings of the Workshop CROCS at CPAIOR-10, Second International Workshop on Constraint Reasoning and Optimization for Computational Sustainability*, pp. 1–13, 2010.

[11] S. Erdogan and E. Miller-Hooks, "A Green Vehicle Routing Problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 1, pp. 100–114, 2012.

[12] D. Goeke and M. Schneider, "Routing a mixed fleet of electric and conventional vehicles," *European Journal of Operational Research*, vol. 245, no. 1, pp. 81–99, 2015.

[13] G. Hiermann, J. Puchinger, S. Ropke, and R. F. Hartl, "The electric fleet size and mix vehicle routing problem with time windows and recharging stations," *European Journal of Operational Research*, vol. 252, no. 3, pp. 995–1018, 2016.

[14] R. Roberti and M. Wen, "The Electric Traveling Salesman Problem with Time Windows," *Transportation Research Part E: Logistics and Transportation Review*, vol. 89, pp. 32–52, 2016.

[15] G. Desaulniers, F. Errico, S. Irnich, and M. Schneider, "Exact algorithms for electric vehicle-routing problems with time windows," *Operations Research*, vol. 64, no. 6, pp. 1388–1405, 2016.

[16] R. G. Conrad and M. A. Figliozzi, "The Recharging Vehicle Routing Problem," in *Proceedings of the 2011 Industrial Engineering Research Conference*, 2011.

[17] M. Schneider, A. Stenger, and D. Goeke, "The electric vehicle-routing problem with time windows and recharging stations," *Transportation Science*, vol. 48, no. 4, pp. 500–520, 2014.

[18] Z. Ma, D. Callaway, and I. Hiskens, "Decentralized charging control for large populations of plug-in electric vehicles: Application of the Nash certainty equivalence principle," in *Proceedings of the 2010 IEEE International Conference on Control Applications, CCA 2010*, pp. 191–195, Yokohama, Japan, September 2010.

[19] Z. Ma, D. S. Callaway, and I. A. Hiskens, "Decentralized charging control of large populations of plug-in electric vehicles," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 1, pp. 67–78, 2013.

[20] L. Gan, U. Topcu, and S. Low, "Optimal decentralized protocol for electric vehicle charging," in *Proceedings of the 2011 50th IEEE Conference on Decision and Control and European Control Conference, CDC-ECC 2011*, pp. 5798–5804, Orlando, FL, USA, December 2011.

[21] L. Gan, U. Topcu, and S. H. Low, "Optimal decentralized protocol for electric vehicle charging," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 940–951, 2013.

[22] K. Clement, E. Haesen, and J. Driesen, "Coordinated charging of multiple plug-in hybrid electric vehicles in residential distribution grids," in *Proceedings of the IEEE/PES Power Systems Conference and Exposition (PSCE '09)*, pp. 1–7, Seattle, WA, USA, March 2009.

[23] K. Clement-Nyns, E. Haesen, and J. Driesen, "The impact of charging plug-in hybrid electric vehicles on a residential distribution grid," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 371–380, 2010.

[24] E. Sortomme and M. A. El-Sharkawi, "Optimal charging strategies for unidirectional vehicle-to-grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 1, pp. 131–138, 2011.

[25] A. Hoke, A. Brissette, D. Maksimović, A. Pratt, and K. Smith, "Electric vehicle charge optimization including effects of lithium-ion battery degradation," in *Proceedings of the 7th IEEE Vehicle Power and Propulsion Conference, VPPC 2011*, Chicago, IL, USA, September 2011.

[26] T. Markel, K. Smith, and A. Pesaran, "Improving Petroleum Displacement Potential of PHEV's Using Enhanced Charging Scenarios," in *proceedings of the EVS-24, 24th International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium*, NREL/CP-540-45730, pp. 1–9, 2009.

[27] J. Hof, M. Schneider, and D. Goeke, "Solving the battery swap station location-routing problem with capacitated electric vehicles using an AVNS algorithm for vehicle-routing problems with intermediate stops," *Transportation Research Part B: Methodological*, vol. 97, pp. 102–112, 2017.

[28] E. S. Rigas, S. D. Ramchurn, and N. Bassiliades, "Managing Electric Vehicles in the Smart Grid Using Artificial Intelligence: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1619–1635, 2015.

[29] L. E. Munoz, J. C. Blanco, J. P. Barreto, N. A. Rincon, and S. D. Roa, "Conceptual design of a hybrid electric off-road vehicle," in *Proceedings of the 2012 IEEE International Electric Vehicle Conference, IEVC 2012*, Greenville, SC, USA, March 2012.

[30] P. Toth and D. Vigo, Eds., *The vehicle routing problem*, vol. 9 of *SIAM Monographs on Discrete Mathematics and Applications*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002.

[31] V. Marano, S. Onori, Y. Guezennec, G. Rizzoni, and N. Madella, "Lithium-ion batteries life estimation for plug-in hybrid electric vehicles," in *Proceedings of the 5th IEEE Vehicle Power and Propulsion Conference, VPPC '09*, pp. 536–543, Dearborn, MI, USA, September 2009.

[32] R. Storn and K. Price, "Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, Article ID 341359, pp. 341–359, 1997, http://dx.doi.org/10.1023/A:1008202821328.

[33] S. Das and P. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

[34] R. Storn, "On the Usage of Differential Evolution for Function Optimization," in *Proceedings of North American Fuzzy Information Processing*, pp. 519–523, 1996.

[35] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[36] S. J. Orfanidis, *Introduction to Signal Processing*, Prentice Hall, 2010.

[37] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 4627856, 15 pages
https://doi.org/10.1155/2017/4627856

Research Article

Mode-Based versus Activity-Based Search for a Nonredundant Resolution of the Multimode Resource-Constrained Project Scheduling Problem

Daniel Morillo, Federico Barber, and Miguel A. Salido

Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain

Correspondence should be addressed to Miguel A. Salido; msalido@dsic.upv.es

Received 7 April 2017; Revised 18 June 2017; Accepted 5 September 2017; Published 11 October 2017

Academic Editor: Jorge Magalhaes-Mendes

This paper addresses an energy-based extension of the Multimode Resource-Constrained Project Scheduling Problem (MRCPSP) called MRCPSP-ENERGY. This extension considers the energy consumption as an additional resource that leads to different execution modes (and durations) of the activities. Consequently, different schedules can be obtained. The objective is to maximize the efficiency of the project, which takes into account the minimization of both makespan and energy consumption. This is a well-known NP-hard problem, such that the application of metaheuristic techniques is necessary to address real-size problems in a reasonable time. This paper shows that the Activity List representation, commonly used in metaheuristics, can lead to obtaining many redundant solutions, that is, solutions that have different representations but are in fact the same. This is a serious disadvantage for a search procedure. We propose a genetic algorithm (GA) for solving the MRCPSP-ENERGY, trying to avoid redundant solutions by focusing the search on the execution modes, by using the Mode List representation. The proposed GA is evaluated on different instances of the PSPLIB-ENERGY library and compared to the results obtained by both exact methods and approximate methods reported in the literature. This library is an extension of the well-known PSPLIB library, which contains MRCPSP-ENERGY test cases.

1. Introduction

The energy consumption in the industry sector is growing by leaps and bounds. Based on the U.S. Energy Information Administration report, in 2016, the industry sector, including manufacturing, consumed approximately a third of the total delivered energy in the world [1]. The environmental implications of the industrial process are gaining more and more importance. Therefore, energy consumption reduction in resource-allocation projects is a critical aspect in the industry sector [2]. For this reason, the interest of researchers is increasingly focused on the development of methodologies for obtaining energy-sustainable solutions. The energy-efficiency oriented scheduling is an actual challenge and a feasible way to save energy in process planning [3].

The Multimode Resource-Constrained Project Scheduling Problem (MRCPSP) is one of the most studied scheduling problems due to the fact that many problems can be modeled as variants of it. One extension of this problem, which incorporates energy consumption in activities, is the so-called MRCPSP-ENERGY that was proposed by Morillo Torres et al. [4]. It includes an additional resource, the energy, that gives rise to different execution modes of activities, and the objective is to maximize the efficiency of the project. This efficiency criterion is managed by combining the *makespan* and the energy consumption criteria into a new combined objective. The objective function of the MRCPSP-ENERGY is more sensitive than the traditional function of the MRCPSP, since solutions that generate the same objective function value for the MRCPSP can generate different values for the MRCPSP-ENERGY. This is because the traditional function of the MRCPSP does not distinguish between solutions with different execution modes if they do not affect the *makespan*; instead, these solutions may lead to different energy consumption in the MRCPSP-ENERGY. This paper addresses the MRCPSP-ENERGY for two main reasons: (1) the wide

interest of reaching energy-efficient solutions in scheduling processes and (2) because the impact of redundant solutions can be deeply analyzed in the presence of a highly sensitive objective function.

Solving MRCPSP-ENERGY instances has an NP-hard complexity, and thus exact methods can only find the optimal solution to small-size instances in a reasonable time. Therefore, metaheuristic methods have become more important because they can find near-optimal solutions in a short time. Most metaheuristic methods use a solution representation based on the Activity List and apply movement rules based on order changes in this list of activities to explore the neighborhood of a solution. This paper shows that this commonly used representation (the Activity List) can produce a large number of redundant solutions, which has a negative impact on the search effort. In order to avoid this disadvantage, a new genetic algorithm (GA) is proposed to solve the MRCPSP-ENERGY that includes two optimization phases. The first is an optimization phase over the Mode List, in which the mutation operator plays the major role, since most of the population undergoes a mutation to improve the exploration. The second is an optimization phase over the Activity List, which includes an operator of mutation based on multiple insertions to decrease the number of redundant solutions.

In order to perform an assessment, four versions of the proposed GA are considered: the ML-GA only includes the optimization phase over the Mode List; the AL-GA only includes the optimization phase over the Activity List; the MIX-GA uses the two phases previously mentioned, simultaneously; and finally, the TP-GA considers the two phases separately. In addition, two fitness functions are considered: relative efficiency, in accordance with the MRCPSP-ENERGY proposal, and a weighted normalized function of the objectives. All algorithms are evaluated by using the PSPLIB-ENERGY library instances. The PSPLIB-ENERGY library [4] is an extension of the well-known PSPLIB library proposed by Kolisch and Sprecher [5] in order to provide MRCPSP-ENERGY instances. The PSPLIB-ENERGY includes four sets of problems ($j30$, $j60$, $j90$, and $j120$), which allows evaluating the performance of search algorithms with different sizes of problems. In addition, the results obtained by the proposed GA are compared with the results given by IBM CPLEX CP optimizer. This is a well-known toolbox that uses constraint programming for solving combinatorial optimization problems.

The main contribution is to show that performing the search through the Mode List is a different way to explore the solution space, which can achieve as competitive solutions or even better ones as the search through the Activity List. Moreover, both search procedures can be combined to achieve even better solutions.

The paper is organized as follows. Section 2 presents the problem description. Section 3 describes main methodologies applied for solving the MRCPSP. Section 4 shows some examples of redundant solutions of the Activity List-based representation. In Section 5, the new genetic algorithm is described, and then Section 6 gives the computational experiments and the result analysis. Section 7 summarizes some concluding remarks and Section 8 points out some future work.

2. Problem Description

The MRCPSP-ENERGY [4] is an extension of the well-known Resource-Constrained Project Scheduling Problem (RCPSP). In the MRCPSP-ENERGY, the activities have different execution modes, associated with different energy consumption levels. Activities also require an amount of renewable resources for their execution; these are resources that can be used by any activity and they are renewed every time the activity that uses them has ended, leaving those units of resources again available for being used by another activity. The goal is to maximize the relative efficiency of the project, which minimizes both the energy consumption and the *makespan* ($C_{\max}$). Formally, the problem can be described as a project that consists of a set of n activities $I = \{0, \ldots, i, \ldots, n\}$, a set B of K^ρ shared renewable resources $B = \{1, \ldots, b, \ldots, K^\rho\}$, and an available amount R_b^ρ of every renewable resource. Each activity i has M_i execution modes, where each mode $m \in M_i$ requires a nonpreemptive execution time d_{im}, a total of r_{ib}^ρ renewable resources of type b, and an amount of energy e_{im} for its realization. Activities are subject to precedence constraints, which indicate that each activity cannot be started before all its predecessor activities are completed. The different energy consumption for each activity gives rise to different execution modes and, consequently, different execution times.

Figure 1 shows an example of a MRCPSP-ENERGY instance. It has 11 activities; the first and the last activity are dummy activities that represent the beginning and the end of the project. There are 3 renewable resources with a maximum availability of 4 units for each of them. At the top of each node, the execution time and energy consumption are presented for each mode, and at the bottom, its resource usage is presented. The arrows show the precedence relations between activities.

Let the total energy consumption of a project (CETP) be the sum of the energy consumption e_{im} for each activity i in a schedule. $LB0_{\min}$ is the critical path with the shortest execution time, $e_{\min}$ is the sum of energy consumption with lower consumption value, and P_j is the set of immediate predecessor activities of an activity j. Given an upper bound T for the project *makespan*, the latest start time (ls_i) and earliest start time (es_i) can be calculated by applying the forward and backward pass method. The binary decision variable ξ_{imt} takes the value 1 when the activity i is executed in mode m and starts at time t, and 0 otherwise. Therefore, the MRCPSP-ENERGY problem can be formulated as follows:

$$\max \quad \eta\left(C_{\max}, \mathrm{CETP}\right) \tag{1}$$

$$\text{Subject to} \quad \sum_{m=1}^{M_i} \sum_{t=\mathrm{es}_i}^{\mathrm{ls}_i} \xi_{imt} = 1 \quad \forall i \in I \tag{2}$$

$$\sum_{m=1}^{M_i} \sum_{t=\mathrm{es}_j}^{\mathrm{ls}_j} t * \xi_{jmt} \geq \sum_{m=1}^{M_i} \sum_{t=\mathrm{es}_i}^{\mathrm{ls}_i} \left(t + d_{im}\right) * \xi_{imt} \quad \forall i \in P_j,\ \forall j \in I \tag{3}$$

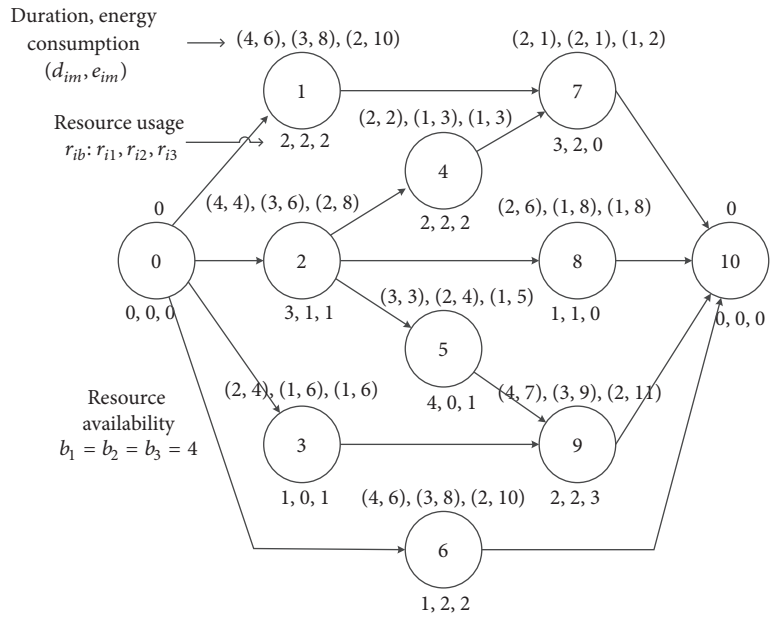


Figure 1: A MRCPSP-ENERGY example.

$$\sum_{i=0}^{n} r_{ib}^{\rho} * \sum_{m=1}^{M_i} \sum_{s=\max\{t-d_{im},\mathrm{es}_i\}}^{\min\{t-1,\mathrm{ls}_i\}} \xi_{ims} \leq R_b^{\rho} \quad \forall b \in B,\ t = 0, \ldots, T \tag{4}$$

$$\xi_{imt} \in \{0, 1\}, \tag{5}$$

where

$$\eta\left(C_{\max}, \mathrm{CETP}\right) = \frac{1}{\mathrm{CSR}} * \frac{1/C_{\max}}{\mathrm{CETP}}. \tag{6}$$

$$\mathrm{CSR} = \frac{1/LB0_{\min}}{e_{\min}} \tag{7}$$

Expression (2) ensures that each activity starts only once. Expression (3) represents the precedence constraints. Expression (4) ensures that the capacity of resources is not exceeded. Finally, expression (1) shows the optimization function where the objective is to maximize the relative efficiency $\eta(C_{\max}, \mathrm{CETP})$, which has been defined by expressions (6) and (7). This criterion considers both the energy consumption and the *makespan*, simultaneously. Overall, the relative efficiency is interpreted as the efficiency of the project regarding an upper bound of the project performance (CSR) (expression (7)).

The MRCPSP-ENERGY is a strongly NP-hard problem, because this problem is a generalization of the standard RCPSP which is well known to be NP-hard.

3. Literature Review

Most of the related academic literature is dedicated to the MRCPSP with both renewable and nonrenewable resources and less attention has been given to addressing the MRCPSP with only renewable resources. However, the solution methods of both problems share many features. The main difference between them is that when nonrenewable resources are considered, infeasible solutions can be reached, and therefore the solution methods often include a penalty function. Following the academic literature, solution methods can be classified into two groups: exact approaches and metaheuristic approaches.

Elmaghraby [6] was the first to consider different execution modes for activities in the project scheduling problem. Later, Talbot [7] proposed a branch and bound (B&B) algorithm for solving the MRCPSP. This consisted of two stages. In the first one, activities, resources, and modes are organized based on several established priority rules. In the second stage, a heuristic based on a priority rule is used to calculate an upper bound and then a backward B&B is implemented. Patterson et al. [8] proposed another enumeration scheme-based procedure; it is a B&B based on the precedence tree to guide the search in the set of all precedence-feasible sequences of activities. Speranza and Vercellis [9] proposed a depth-first B&B, but Hartmann and Sprecher [10] showed that this algorithm might be unable to find the optimal solution for instances with two or more renewable resources.

Since then, several methods have been proposed based on the B&B with different variants. Sprecher et al. [11] proposed a B&B in which an enumeration scheme, called mode-and-delay alternatives, is used as an extension of the alternative concept of delay proposed by Christofides et al. [12], which is also used by Demeulemeester and Herroelen [13] for the RCPSP. The main differences between this approach and the traditional B&B are that at each level more than one activity can be scheduled and that decisions made at previous levels can be undone at the current level.

More recently, similar but more efficient methods have been developed; for example, Zhu et al. [14] proposed a

branch and cut algorithm. Although it was proposed for the multimodal version of the RCPSP with resources partially renewable, it can be used for the traditional MRCPSP. In this approach, the linear relaxation of the linear integer programming model is used to obtain a lower bound of the project duration in each node of the search tree. If the search tree node has a fractional solution, then the algorithm tries to find cuts, that is, valid inequalities that are violated by the fractional solution but are satisfied by the feasible integer solutions represented by that node in the search tree. If no cuts are found in the node, then the branch is carried out, creating new nodes in the tree.

However, in spite of the encouraging results obtained by exact methods, it is important to highlight that exact algorithms in general are unable to optimally solve problems with more than 30 activities, thus leaving the metaheuristic methods as the unique alternative.

Metaheuristic approaches are search algorithms that include escaping strategies from local optima, with the aim of exploring and finding a good approximation to a global optimum. Following the definition stated by Van Peteghem and Vanhoucke [15], the metaheuristic procedures for solving the MRCPSP can be classified into schedule and mode representations, Schedule Generation Schemes (SGS), metaheuristic algorithms, and local search procedures.

Schedule and Mode Representations. A representation stands for how to code a complete solution (i.e., the execution mode and start time of each activity). The coding consists of one schedule representation and one mode representation. Kolisch and Hartmann [16] distinguished five different schedule representations but the *Activity List* representation and the Random Key representation are the most used. The *Activity List* representation consists of a vector of n activities; the order of these elements indicates the priority of an activity to be scheduled. A precedence-feasible list is generally used. The Random Key representation is also a vector with n elements but each of them contains a priority value of the activity in that position. On the other hand, there are mainly two ways of representing modes: a Mode List and a mode vector. The difference between them is that the Mode List represents the execution modes in an ascending order, while the mode vector does it according to the order of the Activity List.

Schedule Generation Schemes. In order to decode a schedule and mode representation in a complete solution, a Schedule Generation Scheme (SGS) is used. Kolisch and Hartmann [16] mainly distinguished two SGS: the serial and the parallel scheme. Both schemes produce feasible solutions. In the serial scheme, solution building is done through a single set of eligible activities that is updated at each step. In the parallel scheme, there are many sets of eligible activities determined by the span in which resources are available. The serial scheme produces a set of schedules that always contain the optimum, while the parallel scheme produces a set of schedules that may exclude it. In spite of this fact, the parallel scheme is also used in the literature because it usually builds more compact solutions than the serial scheme [17].

Metaheuristic Algorithms. There are several metaheuristic methods for solving the MRCPSP. One of the first ones, proposed by Kolisch and Drexl [18], consists of a local search that tries to find a feasible solution and then perform a single neighborhood search on the set of feasible mode assignments. Özdamar [19] proposed two versions of GAs: pure GA and hybrid GA. In the first one, the *Activity List* representation and the serial SGS are used. In the second, a Random Key representation and a parallel SGS are used. The experimental results show that the hybrid GA outperforms all other algorithms tested in that research. Later, Hartmann [20] proposed a GA, which uses a precedence-feasible *Activity List* representation and the serial SGS as the decoding rule. The algorithm includes two local search methods: one was used to deal with the feasibility problem and the other was used to improve the schedules. Józefowska et al. [21] proposed Simulated Annealing (SA) first considering and later disregarding a penalty function, with the latter being the alternative with the best results. They also used the *Activity List* representation and the serial SGS. Bouleimen and Lecocq [22] presented another implementation of SA, where neighbor solutions are generated by using two phases. In the first, a mode-feasible solution is searched and secondly it is improved by random shifts of activities. The first Tabu Search was proposed by Nonobe and Ibaraki [23]. There, the *Activity List* representation is implemented and the neighbor solutions are generated either by a change on the Mode List or by shifting some activities on the Activity List. The authors proposed a new solution building procedure, but the optimal solutions might not be reached by it. In the research made by Alcaraz et al. [24], a GA with equal schedule and mode representation was proposed, but an additional element was included: the forward/backward gene, which indicates the direction of serial SGS to generate a schedule. In the forward direction, the solutions are generated according to the precedence constraints, and in the backward direction they are generated by changing the precedence constraints by successor constraints. The results show that this algorithm outperforms the SA approach proposed by Józefowska et al. [21] but does not exceed the GA proposed by Hartmann [20]. Alternatively, Zhang et al. [25] proposed particle swarm optimization (PSO), which uses two particles as a solution representation. The first particle contains a vector with priority values for each activity (like the Random Key representation), and the second particle contains information about the activity execution modes. Based on their results, the PSO achieves competitive results, although the GA proposed by Hartmann [20] outperforms the PSO. Later, a combinatorial particle swarm optimization (CPSO) was proposed by Jarboui et al. [26]. It generates mode-feasible particles (coded solutions) and then, with a fixed mode assignment, a local search to improve the sequence associated with each particle is performed. The authors compared the algorithm performance with that of the PSO and SA, with CPSO being the algorithm with the best results.

More recently, Li and Zhang [27] proposed an ant colony optimization (ACO) which uses two levels of pheromones: the first level is used to make the selection of an activity and insert it into a sequence and the second level is used

Activity List:	10	7	6	1	4	9	8	5	2	3	0
Activities:	0	1	2	3	4	5	6	7	8	9	10
Mode List:	0	2	2	3	2	1	2	1	3	1	0

Figure 2: Example of an Activity List and a Mode List.

to perform the execution mode assignment. They defined the ACO solution: it is an adaptation of the *Activity List* and the *Mode List* representation to be used in the ACO procedure. The serial SGS was also adapted to generate a complete solution. The results show a high performance but the ACO does not outperform the CPSO. Tseng and Chen [28] proposed a genetic local search with two phases. In the first one, an initial population is generated, and the best solutions are grouped into an elite set. In the second phase, a deep search is carried out in regions defined by the elite set. The solutions in this algorithm are encoded by using the *Activity List* representation. Lova et al. [29] developed an hybrid GA (MM-HGA), which uses, in addition to the *Activity List* representation, two additional genes: forward/backward gene and a serial/parallel gene. The first gene is related to the decoding direction and the second gene is related to the decoding algorithm. The authors also proposed a new normalized fitness function that relates the *makespan* to the units of resources from the excess of nonrenewable resources. Their results show that MM-HGA outperforms the other heuristics. Finally, a bipopulation version of GA was proposed by Van Peteghem and Vanhoucke [30]. The main difference with other genetic algorithms is the use of two different populations: one contains schedules only right-justified and the other contains schedules only left-justified. They adapted the genetic operators to be used with two populations and proposed an extended serial SGS with a local search for improving the mode execution. This algorithm uses the Random Key representation. The obtained results show that this algorithm achieves one of the best solutions in the literature.

To summarize, there are several proposed metaheuristic procedures to solve the MRCPSP and, based on the reported computational experiments, population-based algorithms as well as hybrid metaheuristics are those which achieve the best results. For a wide study on these methodologies, we refer the reader to [31, 32].

Local Search Procedures. The purpose of local search procedures is to iteratively improve the current solution. They often focus on achieving feasibility or improving the *makespan*. One of the most relevant local searches is the forward-backward improvement (FBI) proposed by Tormos and Lova [33] and adapted to be used in the MRCPSP by Lova et al. [29]. The FBI consists of a backward pass and a forward pass. In the first one, the activities are listed from the right to the left and scheduled at the latest feasible time possible. Then, in the forward pass, the activities are listed from the left to the right and scheduled at the earliest feasible time possible. In this way, more compact schedules are often achieved.

4. Redundant Solutions in *Activity List*-Based Representations

It can be deduced from the conclusions of the previous section that the *Activity List*-based representation has a fundamental role in metaheuristic procedures to solve MRCPSP instances. Formally, the *Activity List* representation is an array with n elements that represents activities. This list is feasible with respect to precedence constraints. The position of each activity in the array represents the priority of the activity to be scheduled by using an SGS. On the other hand, the Mode List is also an array with n elements but the elements represent the execution mode of the activities in an ascending order; that is, the element i represents the execution mode of the activity i.

To decode a complete solution, both an Activity List and a Mode List are needed. The decoding can be carried out in several ways: by using the serial or the parallel SGS and the forward or the backward direction. In the backward direction, the predecessor and successor activities are swapped. In Figure 2, an example of an *Activity List* representation and a *Mode List* representation of a feasible solution for the problem represented in Figure 1 is shown.

Metaheuristic procedures use this representation scheme for coding the solutions and then apply movement rules to do the search procedure in the neighborhood. Most procedures do the search through modifications in the Activity List. However, it can be easily deduced that different Activity Lists can obtain equal solutions when they are decoded. For example, Figure 3 shows two different Activity Lists (Activity List 1 and Activity List 2) and one Mode List of feasible solutions for the problem presented in Figure 1. Both Activity Lists were decoded by using the serial SGS in the forward direction. Although both Activity Lists are different (see the underlined activities in Figure 3), the solutions obtained are exactly the same. It is worth noting that the differences between the Activity Lists are not just a simple activity swap. These solutions are named redundant solutions.

Redundancy in solutions can occur at any search procedure when permutations are generated over the Activity List. Estimating the number of redundant solutions of the MRCPSP is not possible without, first, generating, decoding, and checking all permutations of the Activity List. This is because redundant solutions depend on the sequence of scheduled activities and their use of resources at each time point of each solution. Nevertheless, based on the results of Section 6, the number of redundant solutions is much larger than the number of unique solutions. This is one of the first conclusions of our experiments and it is of relevance to the efficiency of the search process. There have been important

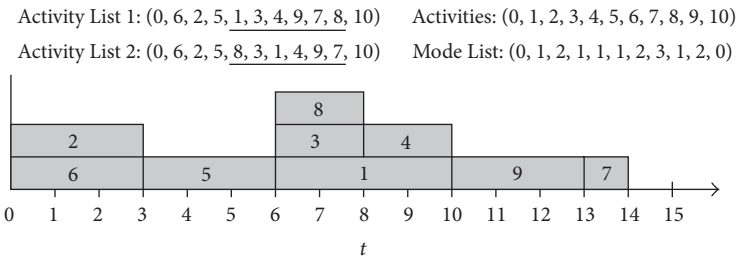


Figure 3: Example of redundant solutions using the *Activity List* representation.

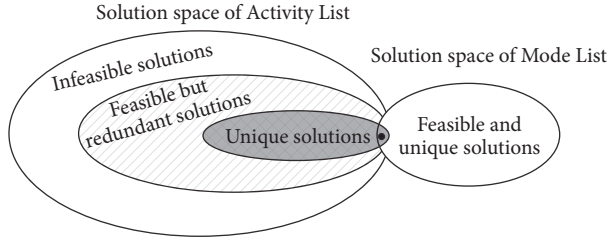


Figure 4: The solution space related to the *Activity List* and the *Mode List* representation.

developments in the literature that address these issues. In particular, Debels et al. [34] proposed a scattered search algorithm that includes four mechanisms to avoid the generation of redundant solutions in Random Key representation. Subsequently, Paraskevopoulos et al. [35] proposed a new representation of solutions called event list. In their proposal, first, a real feasible solution is created and then it is coded into sets of activity events, such as sets of activities that can be performed at the same time. Both approaches address the single-mode version of the RCPSP.

On the other hand, only two conditions must be fulfilled so that any change in the Mode List guarantees a different solution (a complete schedule), even when the Activity List remains constant. The first condition is that the duration of the execution modes of the modified activity is different, and the second condition is that there are other activities that can consume the released resources or cannot be executed at that time due to the lack of them. In fact, if an activity i changes its execution mode, the duration of that activity will also change, and this will cause the serial or parallel decoding algorithms to either have or not have available resources at different times of the scheduling. Furthermore, the successor activities will also be available at different time points.

The space of solutions for a representation based only on a list of n activities is bounded by the number of permutations ($n!$). Similarly, the solution space for a representation based on a list of m execution modes is bounded by the number of combinations (m^n). Of course, the permutation space is rather larger than the combination space, as the number of activities increases. Figure 4 shows an illustration about the solution space of the *Activity List* representation and the *Mode List* representation. In this figure, the solution space of the Mode List can be seen as the solutions related to the combination of the execution modes of a list of activities.

Therefore, the problem lies in the fact that the computational effort of a search based only on permutations of the Activity List could be wasted on redundant solutions. Taking into account the fact that most metaheuristics are mainly focused on the permutations of the Activity List, this paper proposes an in-depth study of searching over the Mode List, instead.

5. A *Mode* and *Activity List*-Based Genetic Algorithm

In this section, a new genetic algorithm (GA) to solve MRCPSP-ENERGY that uses two optimization phases, based on the Activity List and the Mode List, is described. First, we describe the codification scheme, the fitness function, the selection criteria, and the replacement process. Then, the genetic operations of crossover and mutation, related to the two proposed optimization phases (based on the Activity List and the Mode List), are detailed.

5.1. Basic Elements of the Genetic Algorithm

Solutions Encoding and Fitness Function. Following the codification proposed by Lova et al. [29], we used an Activity List and a Mode List plus two genes that are used as codification of solutions in MRCPSP-ENERGY. The two additional genes are the SGS gene and the direction gene. The SGS gene represents the method to be used to decode the solution: its value will be 0 when a serial scheme is used or 1 when a parallel scheme is used. The direction gene can be forward or backward (with a value equal to 0 or 1, resp.).

Regarding the fitness function, two different alternatives to take into account the two objectives (minimizing *makespan* and energy consumption) are considered: first

maximizing the relative efficiency of the project (expression (1)), as it was described in Section 2, and second minimizing normalized convex combination of these two objectives. Expression (8) shows the definition of the former alternative where $C_{\max}$ is the *makespan* of a project, CETP is the total energy consumption of a project, α represents the priority of minimizing the *makespan* assigned by the decision-maker, and $(1 - \alpha)$ represents the priority of minimizing the total energy consumption, also assigned by the decision-maker. The parameters $LB0_{\min}$, T, and $e_{\min}$ were described in Section 2, and $e_{\max}$ is the sum of energy consumption with the highest value. Under these criteria, the objective is to minimize $F(C_{\max}, \text{CETP})$.

$$F\left(C_{\max}, \text{CETP}\right) = \alpha * \frac{C_{\max} - LB0_{\min}}{T - LB0_{\min}} + (1 - \alpha) * \frac{\text{CETP} - e_{\min}}{e_{\max} - e_{\min}}. \tag{8}$$

Initial Population. To generate the initial population, the Regret Based Biased Random Sampling (RBBRS) with the latest start time (LST) as a priority rule is applied; this is one of the best sampling methods to create an initial population [16, 36]. Each solution is obtained in the following way: at the beginning, an eligible set (ele) of activities, where all their predecessors have already been scheduled, is computed. Afterwards, the probability of selecting each activity i of ele is calculated through RBBRS, and one of them is chosen (expressions (9) and (10)). Next, a new eligible set of activities is again computed until no task remains. The values of the SGS gene, direction gene, and execution mode are randomly chosen. The p_i value is called *regret value* of the activity i, which compares the value of the activity priority rule $v(i)$ with the worst value of the priority rule for all activities in the set ele. In this case, the worst priority value is the activity with the maximum LST.

$$p_i = \max_{j \in \text{ele}} v\left(j\right) - v\left(i\right), \tag{9}$$

where $v(i)$ is a priority rule value to be minimized.

$$\text{Probability}\left(i\right) = \frac{\left(p_i + \epsilon\right)^{\alpha}}{\sum_{j \in \text{ele}} \left(p_j + \epsilon\right)^{\alpha}}. \tag{10}$$

The parameters of expressions (9) and (10) are $\epsilon \geq 0$ and $\alpha \geq 0$. A value of $\epsilon \neq 0$ allows a positive probability to the activity with the worst value in the priority rule to be selected. The α parameter determines the selection mechanism: a large value makes a deterministic selection and a value of zero makes a completely random selection.

Population Size. There is not a standard method to estimate the best population size in this kind of problems, although it is known that the population should be related to the complexity and the problem size. Some studies state that the population size should decrease with the increasing number of activities [37, 38]. However, most authors carry out computational experiments to estimate these parameters in a GA.

Because an iteration was defined as a complete solution, it was found that the population size should also be related to the number of iterations available. Thus, if the number of iterations is similar to the population size, there will be very few generations that can be produced. Based on our experiments, we apply expressions (11) to compute population and generation number of the proposed GA.

$$\text{Population} = \left(\frac{1}{n} * \text{iterations}\right) + 15$$
$$\text{generations} = \frac{\text{iterations}}{\text{population}}. \tag{11}$$

Selection. Stochastic sampling with replacement is used, on the basis of the fitness value of the solutions. Therefore, each individual in the population has a probability of being chosen according to its fitness value. When an individual is chosen, a replica of that individual is included in the next selection.

Replacement. Replacement refers to how to create the next population $P3$ from the parent population $P1$ and the offspring population $P2$. First, $P1$ and $P2$ are sorted based on their fitness value. Then, $P3$ is built with 50% of the best $P1$ individuals and 50% of the best $P2$ individuals.

Local Improvement. Two naive local improvements are used in the proposed GA. The first one is the well-known forward-backward improvement (FBI) described in Section 3 and proposed by Tormos and Lova [33]. It is applied over the initial population. The second local improvement consists of reviewing all activities, checking whether they can be executed with less energy consumption (longer execution time) without breaking precedence and resource constraints as long as the *makespan* remains unchanged. It is applied on the best final solution found.

5.2. Optimization Phases. As it was pointed out, the proposed GA divides the search process into two optimization phases, on the Mode List and on the Activity List. The first optimization phase uses crossover and mutation operators over the Mode List in order to expand the search. This plays the most important role in our proposed GA algorithm. The second optimization phase uses crossover and mutation operators over the Activity List, and it is done at the end of the algorithm to intensify the search. Each phase has specific genetic operators (crossover and mutation), which are detailed below.

It is important to note that although each optimization phase focuses on the Activity List or the Mode List, both lists are needed to decode a solution using an SGS. These lists are obtained from the initial population.

5.2.1. Genetic Operators for the Optimization over the Mode List

Crossover. Crossover allows building new solutions from two selected parents. These solutions must share features from both parents. We use a two-point crossover operator only on the Mode List based on the operator proposed by

Alcaraz et al. [24]. Initially, two random integers q_1 and q_2 with $0 < q_1 < q_2 < N$ are generated. The first genes from 0 to q_1 are taken from parent 1, the next genes from q_1 to q_2 are taken from parent 2, and the remaining genes are taken from parent 1 again. The Activity List is randomly inherited only from one of the two parents. The SGS gene and direction gene are inherited when they are of the same value in both parents; otherwise, they are randomly generated.

Mutation. Although mutation does not generally have a main role in genetic operators, it introduces new genetic material, which encourages the exploration. Particularly, a slight perturbation over the Mode List would cause great changes over the solution. The mutation consists of randomly selecting an activity and changing its execution mode. The mutation is applied on each solution (an individual of the population) rather than on each activity, and thus the probability of mutation is independent of the number of project activities. Based on the experimentation performed, the probability value that obtained the best results was 90%.

5.2.2. Genetic Operators for the Optimization over the Activity List

Crossover. A modified two-point crossover is used over the Activity List. Thus, two random integers q_1 and q_2 with $0 < q_1 < q_2 < N$ are generated. Thus, the first genes from 0 to q_1 are taken from parent 1, and the next genes from q_1 to q_2 are taken from parent 2. Here, these genes inherited (q_1 to q_2) might not be those that are in the positions q_1 to q_2 of parent 2, such that they must be searched from the beginning of the Activity List of parent 2, and the first activities that are not repeated in the child are inherited. In the same way, the remaining ones (q_2 to N) are taken from parent 1 again, without repeating genes from both parent 1 and parent 2. The modes of activities are inherited from their corresponding parents. For SGS and direction genes, the gene value is inherited when it is the same in both parents; otherwise, it is randomly generated.

Mutation. A multi-insertion mutation operator based on the research conducted by Boctor [39] is proposed to minimize the probability of producing the same solution. It consists of randomly selecting an activity i and then inserting it in a randomly chosen position. The new position must be higher than that of its predecessors and lower than that of its successors. Usually, the mutation is applied on each activity in a solution with a fixed probability; this implies that the average number of mutated activities is dependent on the total number of activities: the more the activities, the greater the probability of a mutation occurring. In contrast, our experimental results show that the average number of mutated activities should be independent of the total number of project activities. That value was estimated by using the PSPLIB-ENERGY library; an appropriate number of three mutated activities per solution was obtained. Similarly, the probability of mutation is also fixed per solution, independent of the total number of activities, and the estimated value of that probability is 90%.

Finally, the GA's parameters were fixed through experimentation on the PSPLIB-ENERGY library. The number of iterations to change from the optimization phase over the Mode List to the optimization phase over the Activity List was determined as 2/3 of the total number of iterations. This value represents the importance of the optimization phase over the Mode List in relation to the optimization phase over the Activity List.

Figure 5 shows an example of how the offspring is generated by using both the Activity List optimization and the Mode List optimization. In this example, the serial SGS and the forward direction are used. Schedule (a) is parent 1 (genes with overline) and schedule (b) is parent 2 (underlined genes), which were used to generate schedules (c) and (d). Schedule (c) is created by the optimization over the Activity List. To this end, a two-point crossover was used; the first 5 activities are inherited from parent 1; then, the following 3 activities are sought from the beginning of the Activity List of parent 2 and are inherited in only those that are not repeated in the Activity List of the child (activities inherited are 1, 4, and 7); finally, the 3 remaining activities are inherited again from parent 1; these are searched from the beginning of the Activity List of parent 1 and only those that are not repeated are taken. As can be seen in this figure, despite using a two-point crossover from different parents, schedule (c) has the same Activity List as parent 1. The only reason why solutions are different is because the modes of activities inherited from parent 2 have different durations to the same activities of parent 1. Schedule (d) is created by the optimization over the Mode List; it inherits the Activity List from parent 2 and the Mode List is created by a two-point crossover, where the fifth and sixth positions of the Mode List are inherited from parent 2; the remaining activities' modes are taken from parent 1. It can be observed that although the Activity List of parent 2 and the Activity List of schedule (d) are exactly the same, the corresponding solutions are very different, due to the different Mode Lists. In fact, schedule (d) has the best fitness value among the four schedules.

Usually, it is assumed that modifications in a Mode List only affect the execution modes of a preestablished order of activities in the Activity List. But it is not like that. In fact, that preestablished order also changes, as shown in Figure 5.

6. Empirical Assessment and Results

The performed empirical assessment of the proposed GA and the obtained computational results are divided into two groups: the first group is focused on the impact of the redundant solutions of the *Activity List*-based representation and the second group is focused on the performance assessment of the proposed GA by using the PSPLIB-ENERGY library.

This PSPLIB-ENERGY library [4] is a set of MRCPSP-ENERGY instances. It consists of four sets of problems: $j30$, $j60$, $j90$, and $j120$. Each of them has 480 instances, except the last one with 600 instead. Each set has 30, 60, 90, and 120 jobs, respectively. All problems are composed of activities with three execution modes.

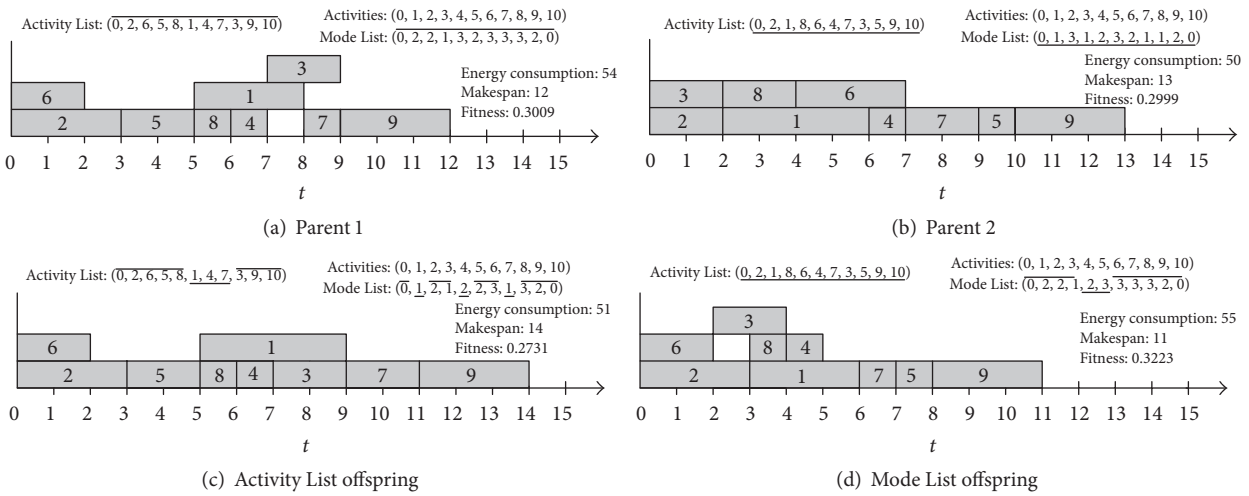


Figure 5: An example of offspring, by using the optimization phases on the Activity List (offspring (c)) and on the Mode List (offspring (d)).

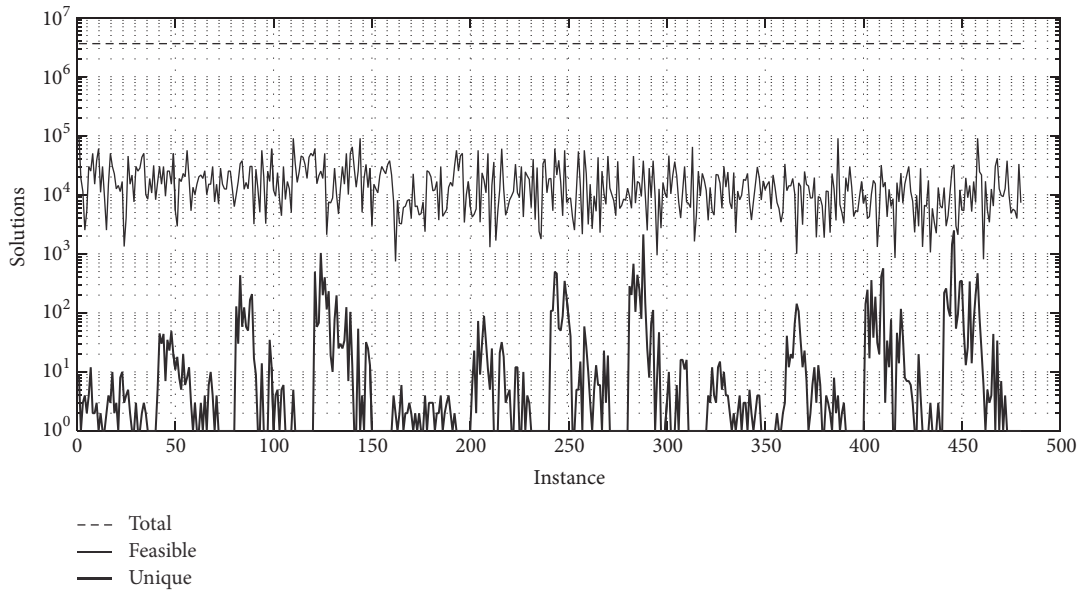


Figure 6: Activity List permutations for the set $j10$.

6.1. Impact of Redundant Solutions in the Activity List-Based Representation. To carry out this evaluation, an exhaustive search was made through all the permutations of the Activity List and all the combinations of the Mode List. Due to the high computational cost of this exhaustive search in problems with many activities, a set of 480 instances was used, each with 10 activities. These instances are based on the set $J30$ of the PSPLIB-ENERGY.

Figures 6 and 7 show (i) the total number of solutions (total), (ii) the number of feasible solutions including redundant ones (feasible), and (iii) the number of unique solutions (unique), of all permutations generated by the Activity List (Figure 6) and the full set of combinations generated by the Mode List (Figure 7).

In Figure 6, the total number of permutations (10! = 3,628,800) is the same for all instances. Due to the huge number of permutations, Figure 6 is shown in a logarithmic scale. As can be seen, the number of redundant solutions is rather higher than that of the unique ones. Indeed, there are some instances with only one unique solution.

In Figure 7, the total number of combinations for each Activity List (3^{10} = 59,049) is the same for all instances, and all combinations are feasible. There are some instances with a number of unique solutions different from the number of feasible solutions. This is because some activities have equal modes. For instance, in the problem shown in Figure 1, activity 3 has mode 2, which corresponds to a duration of 1 time unit and energy consumption of 6. This activity cannot be executed in a lower time than 1, and thus mode 3 is equal to mode 2.

Therefore, although the solution space of the *Activity List*-based representation is higher than the solution space

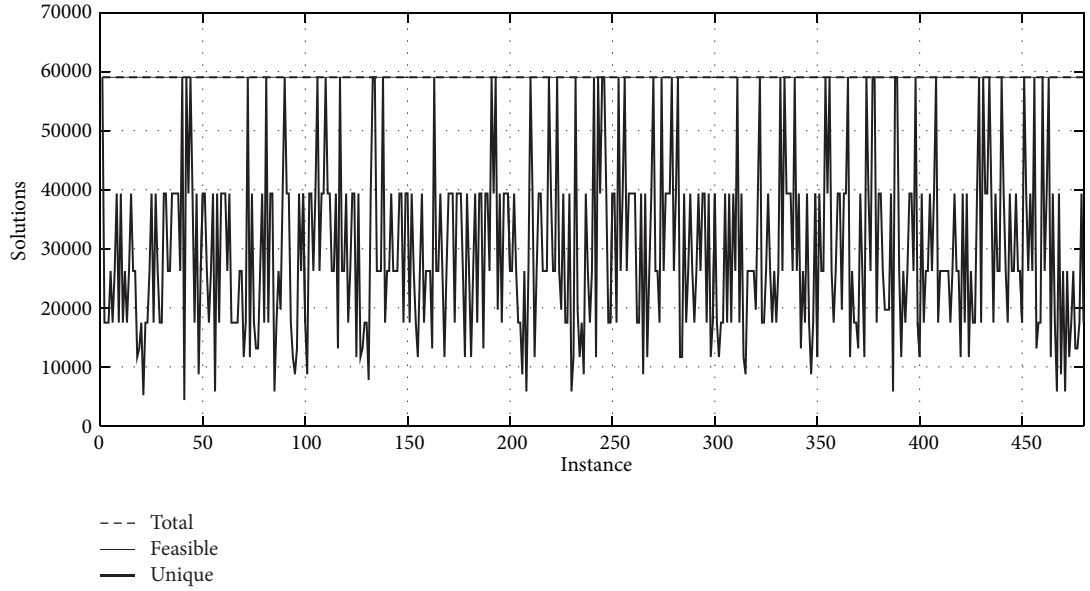


Figure 7: Mode list combinations for the set $j10$.

of the *Mode List*-based representation, most of its solutions are redundant, which makes the search unsuccessful. In contrast, the solution space of the *Mode List* representation is composed of feasible and unique solutions. There are only redundant solutions when there are activities with duplicated execution modes.

6.2. Assessment of the Proposed Genetic Algorithm. In this point, we assess the performance of the proposed GA by using the PSPLIB-ENERGY library. Since this library is based on the well-known PSPLIB library, instance sets were created based on two sets of variable parameters. The first set was used to create the 480 instances of each set: $j30$, $j60$, and $j90$. The second set of parameters was used to create 600 instances for the set $j120$. As is usual in the evaluation of the RCPSP, in this work, the first two instance sets ($j30$, $j60$) will be taken as a reference for small instances, and the set $j120$ will be taken as a reference of large instances. To the best of our knowledge, Morillo Torres et al. [4] are the only authors that have reported results for the instances in this library. Therefore, the proposed GA was compared with that algorithm (called in this paper Basic-GA). Basic-GA uses a coding based on the list of activities and modes, a crossover operator at a single point, and a mutation based on the insertion of Boctor. When an activity is selected, its position is changed in the list of activities, as well as its execution mode. The main difference between this algorithm and the proposed GA is that the Basic-GA takes the standard elements of a genetic algorithm for the MRCPSP, focusing the genetic operators on modifying the Activity List, leaving the mutation as the main tool to change the execution modes.

On the other hand, one of the most used stop criteria in the literature for RCPSP and MRCPSP is the maximum number of iterations. Kolisch and Hartmann [32] defined an iteration as a complete scheduling and showed the reason to use this termination condition as a comparing rule with other solving methods. In this paper, the same stop criteria are used.

The empirical assessment was done considering four variants of the proposed GA. In this way, objective comparisons can be made to identify the elements that contribute to the search. The variants of the proposed algorithm are described below.

(i) The genetic algorithm based on the optimization phase over the Activity List (AL-GA): this algorithm is focused on performing the search by exploring the permutations of the Activity List by using the operators described in Section 5.2.2. The mode of the activities of the population remains constant after the creation of the initial population. The offspring inherits the activities with the execution mode of the corresponding parent.

(ii) The genetic algorithm based on the optimization phase over the Mode List (ML-GA): the search performed by this algorithm is focused on modifying the Mode List of the population (over all combinations) by using the operators described in Section 5.2.1. Activity lists of the population remain constant after the creation of the initial population.

(iii) The genetic algorithm based on the simultaneous mixing of the optimization phases (MIX-GA): this algorithm mixes the two optimization approaches: the optimization phase over the Mode List and over the Activity List. In this way, the operators of both phases are used simultaneously. In this case, neither activity nor Mode Lists remain constant.

(iv) The genetic algorithm based on the two separate optimization phases (TP-GA): this algorithm incorporates the two optimization phases separately, but

Table 1: $\overline{\eta}$ obtained by using the proposed genetic algorithms for solving MRCPSP-ENERGY library.

$j\#$	Algorithm	Iterations/$\overline{\eta}$		
		1000	5000	50,000
$j30$	*TP-GA*	*0.6397*	*0.6517*	*0.6564*
	ML-GA	0.6381	0.6506	0.6559
	AL-GA	0.6345	0.6489	0.6551
	MIX-GA	0.6290	0.6398	0.6469
	Basic-GA	0.5966	0.6091	0.6293
$j60$	*TP-GA*	0.6565	*0.6803*	*0.6919*
	ML-GA	*0.6576*	0.6794	0.6907
	AL-GA	0.6418	0.6700	0.6890
	MIX-GA	0.6498	0.6672	0.6773
	Basic-GA	0.6029	0.6182	0.6424
$j120$	*TP-GA*	0.5192	0.5382	*0.5590*
	ML-GA	*0.5211*	*0.5397*	0.5589
	AL-GA	0.5092	0.5237	0.5523
	MIX-GA	0.5158	0.5310	0.5477
	Basic-GA	0.4760	0.4875	0.5032

with greater emphasis on the search for execution modes by using 2/3 of the total number of available iterations in the optimization phase over the Mode List and in the end 1/3 of the total number of available iterations in the optimization phase over the Activity List.

Table 1 shows the average relative efficiency of different variants of the proposed GA and the Basic-GA to solve three sets of instances of the PSPLIB-ENERGY for 1000, 5000, and 50,000 iterations. This table is sorted by the 50,000-iteration column.

As it can be seen from Table 1, the ML-GA outperforms the AL-GA. Note that the objective is to maximize the relative efficiency of the project. It is interesting to remark that, in the ML-GA, the Activity Lists of all instances have remained constant once the initial population is created, and then only the modes have changed. On the other hand, in AL-GA, the Mode Lists of all instances have remained constant, and only the Activity Lists were modified. This indicates that searching through Mode Lists can achieve high quality solutions, as compared to searching through Activity Lists.

Due to redundant solutions with different Activity Lists, an optimal solution can be represented by a Mode List and several Activity Lists. Thus, by fixing a list and searching over the other, optimal solutions can be excluded. However, the proposed TP-GA searches over the Mode List and, in the final phase, over the Activity List. This combined search procedure achieves better results. In fact, Table 1 shows that the TP-GA outperforms all other algorithms in respect of maximizing the relative efficiency of the project.

On the other hand, we also compared the proposed algorithms based on the minimization of the normalized value of the convex combination for both objectives: the *makespan* and the total energy consumption (expression (8)). Without losing generality, three different values of α are selected: 0.25, 0.5, and 0.75. Table 2 shows a summary of the results. The first row shows the alpha value and the second row shows the iterations number. The first, fifth, and ninth columns show the algorithm version used. The remaining columns show the value of the normalized objective function ($F(C_{\max}, \text{CETP})$).

Based on the results of Table 2, it can be seen that the TP-GA is still the algorithm that obtains better results on average; it outperforms the other algorithms in 7 out of 9 experimental sets. In addition, in the sets $j60$ with $\alpha = 0.25$ and $j60$ with $\alpha = 0.5$, the differences regarding the first position are 0.09% and 0.27%, respectively. In addition, it can be observed that the search by means of the optimization over the Mode List (ML-GA) can reach solutions similar to those reached by the search using the optimization over the Activity List (AL-GA) and outperforms it in some cases. On the other hand, simultaneously mixing the two phases of search does not seem appropriate because it obtains worse results on average; this could be because the mixture can generate too much diversity in the population, thus slowing the convergence.

Since there are no reported optimal solutions for the test cases provided in this library, it is useful to obtain results by an exact approach. Thus, IBM ILOG CPLEX CP optimizer 12.6.2 was used to solve some instances of the MRCPSP-ENERGY library. This toolbox uses constraint programming because it has shown significant results in combinatorial problems such as scheduling problems.

Since the MRCPSP-ENERGY is an NP-hard problem, it is for now impossible to determine optimal solutions in a reasonable time for real-size instances. Therefore, we set a deadline of 30 minutes for solving each problem. We consider this as a reasonable time to address scheduling problems with an exact approach. When the deadline is reached, the best solution is returned. In addition, considering that the optimal solutions for unimodal RCPSP instances with 60 activities are unknown, trying to find the optimal solutions for the MRCPSP-ENERGY with 60 activities is actually more difficult, and we only run the set $j30$.

The exact approach has been able to obtain 70% of the optimal solutions for set $j30$, with a maximum time of 30 minutes for each problem. Table 3 shows a summary of such results. The first column shows the set of instances used. The second column presents the method used to resolve the instances. The third column shows the average value of the objective function (relative efficiency of the project). The fourth column shows the number of optimal solutions found and the total number of instances. The fifth column shows the average execution time for all instances. Finally, the last column shows the limit of execution time.

The exact approach (CPLEX) reached an average value of 65.97% of relative efficiency of the project with an average computational time of 601.15 seconds. The proposed GA reaches an average value of 65.75% of relative efficiency with an average computational time of 2.51 seconds. Thus, the difference between the objective function average provided by CPLEX and that of the proposed GA is 0.3335%, taking into account the fact that the proposed GA requires 99.58% less time than CPLEX to achieve these results. As it can be

Table 2: Normalized value obtained by using the proposed genetic algorithms for solving MRCPSP-ENERGY library.

$\alpha = 0.25$				$\alpha = 0.5$				$\alpha = 0.75$			
Alg.	Iterations/F			Alg.	Iterations/F			Alg.	Iterations/F		
	1000	5000	50,000		1000	5000	50,000		1000	5000	50,000
Set $j30$											
TP-GA	*0.0728*	*0.0531*	*0.0525*	*TP-GA*	*0.1185*	*0.1044*	*0.1029*	*TP-GA*	*0.1329*	*0.1247*	*0.1225*
AL-GA	0.0907	0.0550	0.0525	AL-GA	0.1282	0.1059	0.1031	ML-GA	0.1335	0.1254	0.1227
ML-GA	0.0781	0.0542	0.0529	ML-GA	0.1207	0.1055	0.1034	AL-GA	0.1364	0.1260	0.1230
MIX-GA	0.1011	0.0693	0.0571	MIX-GA	0.1358	0.1170	0.1088	MIX-GA	0.1417	0.1329	0.1281
Set $j60$											
AL-GA	0.1390	0.0605	*0.0331*	*AL-GA*	0.1316	0.0828	*0.0651*	*TP-GA*	0.1034	*0.0866*	*0.0828*
TP-GA	0.0965	*0.0356*	0.0331	TP-GA	0.1064	*0.0685*	0.0653	AL-GA	0.1137	0.0928	0.0832
ML-GA	*0.0937*	0.0363	0.0339	ML-GA	*0.1037*	0.0693	0.0664	ML-GA	*0.1023*	0.0870	0.0833
MIX-GA	0.1061	0.0538	0.0367	MIX-GA	0.1142	0.0819	0.0705	MIX-GA	0.1095	0.0949	0.0886
Set $j120$											
TP-GA	0.1599	0.0551	*0.0304*	*TP-GA*	0.1442	0.078	*0.0604*	*TP-GA*	0.1239	0.0981	*0.0866*
ML-GA	0.1467	*0.0467*	0.0322	ML-GA	*0.1361*	*0.0743*	0.0632	ML-GA	*0.1207*	*0.0960*	0.0878
MIX-GA	*0.1443*	0.0641	0.0343	AL-GA	0.1718	0.1267	0.0641	AL-GA	0.1358	0.1177	0.0894
AL-GA	0.2059	0.1322	0.0355	MIX-GA	0.1380	0.0885	0.0669	MIX-GA	0.1261	0.1055	0.0928

Table 3: Results obtained by using IBM ILOG CPLEX CP optimizer for solving set $j30$ of the MRCPSP-ENERGY library.

Method	$\bar{\eta}$	# optimal	Average time (s)	Deadline (s)
Set $j30$				
CPLEX	0.6597	340 (480)	601.15	1800
TP-GA	*0.6575*	*218 (480)*	*2.51*	—

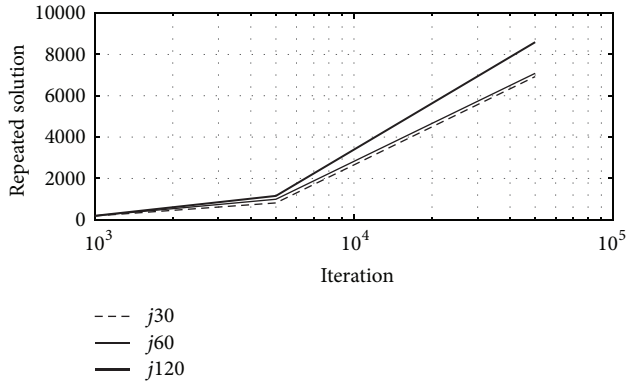


Figure 8: The average number of redundant offspring.

seen, the GA achieves a beneficial trade-off between time and accuracy.

Finally, in order to experimentally assess the impact of redundant solutions, we calculate the average number of redundant offspring generated by the genetic operators in the optimization phase of the Activity List, where both parents are different from each other. The offspring from equal parents were not counted. The estimates were worked out for sets $j30$, $j60$, and $j120$, for 1000, 5000, and 50,000 iterations, respectively. The results are shown in Figure 8. As expected, the number of redundant solutions is related to the number of iterations and, to a lesser degree, to the activities number. From the results, the average number of redundant solutions that do not contribute with new information to the GA is at least 15% of the total number of solutions.

7. Conclusions

In this paper, the Multimode Resource-Constrained Project Scheduling Problem, particularly the MRCPSP-ENERGY, has been addressed with the aim of showing the importance of the solution representation in the metaheuristic techniques. Particularly, we analyze the redundant solutions issue of one of the most important solution representations: the *Activity List*.

A new genetic algorithm with two phases of optimization (TP-GA) has been proposed, whose main contribution is to show that there is an undervalued search alternative in multimode resource scheduling problems. Here, the search is focused on Mode Lists instead of doing it on Activity Lists. This proposal is based on the fact that although genetic operators can modify the Activity List, the resulting solution (complete scheduling) can be exactly the same, even if they come from different Activity Lists. Therefore, the computational effort of a search when it is based only on changes of the Activity List might be wasted on redundant solutions. While a change on the Activity List may lead to the same scheduling, a change on the Mode List always guarantees a different scheduling as long as activities have a different execution time and there exist activities that can take advantage of the availability or lack of resources released by a change in the execution mode.

The results show that when the objective function is the relative efficiency, the proposed genetic algorithm achieves the best results in comparison with the other algorithms for solving the MRCPSP-ENERGY in all problem sets. With regard to minimizing the weighted normalized value of both *makespan* and total energy consumption, the proposed GA also obtained one of the best results; it outperforms the other algorithms in seven out of nine experimental sets and in the other cases achieves the second place. The results also show that the search when modifying only Mode Lists can achieve the best results compared to the search when modifying only Activity Lists. In addition, a comparison of the results between the proposed genetic algorithm and an exact approach was performed. For this, we used IBM ILOG CPLEX CP optimizer. Although the exact approach produces slightly better results, the proposed algorithm uses 99.58% less time. We can conclude that the proposed genetic algorithm achieves highly efficient results.

Based on these results and the fact that the literature about scheduling problems with different execution modes is focused on the optimization of the Activity List, we think there exists a valuable field of research focused not only on the optimization of the Activity List, but also on the optimization of the Mode List. The latter optimization process becomes the most successful search procedure when compared with the former.

8. Future Work

The paper has focused on the analysis of solution representations in the MRCPSP-ENERGY which seeks to minimize both the *makespan* and the energy consumption. The next step would be to consider energy as a nonrenewable resource. In addition, energy consumption can be generalized to the use of other nonrenewable resources, such as budget and fossil fuels. The analysis could be extended to the traditional MRSPCP, which only considers the *makespan*. Here, the impact of the search focused on the *Mode List* representation can be analyzed. Likewise, an analysis on how the existence of redundant solutions affects other solution representations could be performed. Finally, a new solution representation that can take advantage of the search through the Mode List could be proposed, being independent of an Activity List to decode a solution, or a combined representation that avoids redundant solutions.

Disclosure

An earlier version of this work was presented at COPLAS'17 (informal proceedings).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper has been partially supported by the Spanish Research Projects TIN2013-46511-C2-1-P and TIN2016-80856-R.

References

[1] Energy Information Administration (EIA), "Monthly Energy Review - October 2016," Tech. Rep., Office of Energy Statistics, 2016.

[2] Z. Zhang, R. Tang, T. Peng, L. Tao, and S. Jia, "A method for minimizing the energy consumption of machining system: integration of process planning and scheduling," *Journal of Cleaner Production*, vol. 137, pp. 1647–1662, 2016.

[3] G. Mouzon, M. B. Yildirim, and J. Twomey, "Operational methods for minimization of energy consumption of manufacturing equipment," *International Journal of Production Research*, vol. 45, no. 18-19, pp. 4247–4271, 2007.

[4] D. Morillo Torres, F. Barber, and M. A. Salido, "A new model and metaheuristic approach for the energy-based resource-constrained scheduling problem," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, pp. 1–13, 2017.

[5] R. Kolisch and A. Sprecher, "PSPLIB—a project scheduling problem library," *European Journal of Operational Research*, vol. 96, no. 1, pp. 205–216, 1997.

[6] S. E. Elmaghraby, *Activity Networks: Project Planning and Control by Network Models*, John Wiley & Sons Inc, 1977.

[7] F. B. Talbot, "Resource-constrained project scheduling with time-resource tradeoffs: the nompreemptive case," *Management Science*, vol. 28, no. 10, pp. 1197–1210, 1982.

[8] J. H. Patterson, R. Slowinski, F. B. Talbot, and J. Weglarz, "An algorithm for a general class of precedence and resource constrained scheduling problems," *Advances in Project Scheduling*, pp. 3–28, 1989.

[9] M. G. Speranza and C. Vercellis, "Hierarchical models for multi-project planning and scheduling," *European Journal of Operational Research*, vol. 64, no. 2, pp. 312–325, 1993.

[10] S. Hartmann and A. Sprecher, "A note on "hierarchical models for multi-project planning and scheduling"," *European Journal of Operational Research*, vol. 94, no. 2, pp. 377–383, 1996.

[11] A. Sprecher, S. Hartmann, and A. Drexl, "An exact algorithm for project scheduling with multiple modes," *OR Spektrum. Quantitative Approaches in Management*, vol. 19, no. 3, pp. 195–203, 1997.

[12] N. Christofides, R. 'Alvarez-Valdes, and J. M. Tamarit, "Project scheduling with resource constraints: a branch and bound approach," *European Journal of Operational Research*, vol. 29, no. 3, pp. 262–273, 1987.

[13] E. Demeulemeester and W. Herroelen, "A branch-and-bound procedure for the multiple resource-constrained project scheduling problem," *Management Science*, vol. 38, no. 12, pp. 1803–1818, 1992.

[14] G. Zhu, J. F. Bard, and G. Yu, "A branch-and-cut procedure for the multimode resource-constrained project-scheduling problem," *INFORMS Journal on Computing*, vol. 18, no. 3, pp. 377–390, 2006.

[15] V. Van Peteghem and M. Vanhoucke, "An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances," *European Journal of Operational Research*, vol. 235, no. 1, pp. 62–72, 2014.

[16] R. Kolisch and S. Hartmann, "Heuristic algorithms for the resource-constrained project scheduling problem: classification and computational analysis," in *Project Scheduling*, vol. 14 of *International Series in Operations Research & Management Science*, pp. 147–178, Springer, 1999.

[17] J.-L. Kim, "Proposed methodology for comparing schedule generation schemes in construction resource scheduling," in *Proceedings of the 2009 Winter Simulation Conference, (WSC '09)*, pp. 2745–2750, December 2009.

[18] R. Kolisch and A. Drexl, "Local for multi-mode resource-constrained project," *IIE Transactions (Institute of Industrial Engineers)*, vol. 29, no. 11, pp. 987–999, 1997.

[19] L. Özdamar, "A genetic algorithm approach to a general category project scheduling problem," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 29, no. 1, pp. 44–59, 1999.

[20] S. Hartmann, "Project scheduling with multiple modes: a genetic algorithm," *Annals of Operations Research*, vol. 102, pp. 111–135, 2001.

[21] J. Józefowska, M. Mika, R. Rózycki, G. Waligóra, and J. Weglarz, "Simulated annealing for multi-mode resource-constrained project scheduling," *Annals of Operations Research*, vol. 102, pp. 137–155, 2001.

[22] K. Bouleimen and H. Lecocq, "A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version," *European Journal of Operational Research*, vol. 149, no. 2, pp. 268–281, 2003.

[23] K. Nonobe and T. Ibaraki, "Formulation and tabu search algorithm for the resource constrained project scheduling problem," in *Essays and surveys in metaheuristics (Angra dos Reis, 1999)*, vol. 15 of *Oper. Res./Comput. Sci. Interfaces Ser.*, pp. 557–588, Kluwer Acad. Publ., Boston, MA, USA, 2002.

[24] J. Alcaraz, C. Maroto, and R. Ruiz, "Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms," *Journal of the Operational Research Society*, vol. 54, no. 6, pp. 614–626, 2003.

[25] H. Zhang, C. M. Tam, and H. Li, "Multimode project scheduling based on particle swarm optimization," *Computer-Aided Civil and Infrastructure Engineering*, vol. 21, no. 2, pp. 93–103, 2006.

[26] B. Jarboui, N. Damak, P. Siarry, and A. Rebai, "A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems," *Applied Mathematics and Computation*, vol. 195, no. 1, pp. 299–308, 2008.

[27] H. Li and H. Zhang, "Ant colony optimization-based multi-mode scheduling under renewable and nonrenewable resource constraints," *Automation in Construction*, vol. 35, pp. 431–438, 2013.

[28] L. Y. Tseng and S. C. Chen, "Two-phase genetic local search algorithm for the multimode resource-constrained project scheduling problem," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 848–857, 2009.

[29] A. Lova, P. Tormos, M. Cervantes, and F. Barber, "An efficient hybrid genetic algorithm for scheduling projects with resource

constraints and multiple execution modes," *International Journal of Production Economics*, vol. 117, no. 2, pp. 302–316, 2009.

[30] V. Van Peteghem and M. Vanhoucke, "A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem," *European Journal of Operational Research*, vol. 201, no. 2, pp. 409–418, 2010.

[31] J. Weglarz, J. Józefowska, M. Mika, and G. Waligóra, "Project scheduling with finite or infinite number of activity processing modes—a survey," *European Journal of Operational Research*, vol. 208, no. 3, pp. 177–205, 2011.

[32] R. Kolisch and S. Hartmann, "Experimental investigation of heuristics for resource-constrained project scheduling: an update," *European Journal of Operational Research*, vol. 174, no. 1, pp. 23–37, 2006.

[33] P. Tormos and A. Lova, "A competitive heuristic solution technique for resource-constrained project scheduling," *Annals of Operations Research*, vol. 102, pp. 65–81, 2001.

[34] D. Debels, B. De Reyck, R. Leus, and M. Vanhoucke, "A hybrid scatter search/electromagnetism meta-heuristic for project scheduling," *European Journal of Operational Research*, vol. 169, no. 2, pp. 638–653, 2006.

[35] D. C. Paraskevopoulos, C. D. Tarantilis, and G. Ioannou, "Solving project scheduling problems with resource constraints via an event list-based evolutionary algorithm," *Expert Systems with Applications*, vol. 39, no. 4, pp. 3983–3994, 2012.

[36] A. Drexl, "Scheduling of project networks by job assignment," *Management Science*, vol. 37, no. 12, pp. 1590–1602, 1991.

[37] D. Debels and M. Vanhoucke, "A decomposition-based genetic algorithm for the resource-constrained project-scheduling problem," *Operations Research*, vol. 55, no. 3, pp. 457–469, 2007.

[38] M. Cervantes, A. Lova, P. Tormos, and F. Barber, "A dynamic population steady-state genetic algorithm for the resource-constrained project scheduling problem," in *New Frontiers in Applied Artificial Telligence*, L. Thanh Nguyen, A. Borzemski, Grzech., and., and M. Ali, Eds., vol. volume, 502, pp. 611–620, Springer, Berlin, Germany, 2008.

[39] F. F. Boctor, "Resource-constrained project scheduling by simulated annealing," *International Journal of Production Research*, vol. 34, no. 8, pp. 2335–2351, 1996.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 3046830, 20 pages
https://doi.org/10.1155/2017/3046830

Review Article

Bee-Inspired Algorithms Applied to Vehicle Routing Problems: A Survey and a Proposal

Thiago A. S. Masutti and Leandro N. de Castro

Natural Computing and Machine Learning Laboratory (LCoN), Graduate Program in Electrical Engineering and Computing, Mackenzie Presbyterian University, R. da Consolação 930, Higienópolis, 01302-000 São Paulo, SP, Brazil

Correspondence should be addressed to Leandro N. de Castro; lnunes@mackenzie.br

Received 25 January 2017; Revised 29 July 2017; Accepted 22 August 2017; Published 8 October 2017

Academic Editor: Jorge Magalhaes-Mendes

Vehicle routing problems constitute a class of combinatorial optimization tasks that search for optimal routes (e.g., minimal cost routes) for one or more vehicles to attend a set of nodes (e.g., cities or customers). Finding the optimal solution to vehicle routing tasks is an NP-hard problem, meaning that the size of problems that can be solved by exhaustive search is limited. From a practical perspective, this class of problems has a wide and important set of applications, from the distribution of goods to the integrated chip design. Rooted on the use of collective intelligence, swarm-inspired algorithms, more specifically bee-inspired approaches, have been used with good performance to solve such problems. In this context, the present paper provides a broad review on the use of bee-inspired methods for solving vehicle routing problems, introduces a new approach to solve one of the main tasks in this area (the travelling salesman problem), and describes open problems in the field.

1. Introduction

The characteristics of an optimization problem, such as the high number of possible solutions (size of the search space) and number and type of constraints, can prevent its solution by exhaustive search methods in a feasible time, even if large amounts of computational resources are employed. For this reason, instead of using methods that calculate the exact solution to the problem, techniques that propose candidate solutions and consider the history of results in the search process, modifying these solutions iteratively until a satisfactory solution is found, are usually employed. These methods exchange the guarantee of the exact solution for a satisfactory solution that can be obtained with an acceptable computational cost. These methods are normally called metaheuristics [1–3].

The planning of routes used to transport products is one of the ways to improve the use of resources in various transportation modals. For instance, a beverage distributor has its vehicle filled with goods and must distribute them to a certain group of customers. Defining the best sequence of customer visits, known as a route, can help reduce the cost of this operation and, consequently, impact the final product cost. In the literature, the route planning task is defined as a vehicle routing problem (VRP), with the travelling salesman problem (TSP) as one of the most elementary and widely studied cases [4–10]. The VRPs belong to the category of NP-hard problems, making them difficult to solve by exact methods, mainly for larger instances. Therefore, much of the study of this class of problems is based on metaheuristics [11, 12].

One area that has received considerable attention from the scientific community over the past decades is that of nature-inspired algorithms [13–15]. Studies in this area seek inspiration in biological phenomena for the development of algorithms capable of solving problems that are not satisfactorily solved by traditional techniques [16]. In such cases, heuristics inspired by insect behaviors have shown good results. Swarm intelligence (SI) is a term used to describe algorithms that have as inspiration the collective behavior of social insects or other types of social animals [17, 18]. Within swarm intelligence, the agents in the swarm

act with no supervision, being affected by what happens in the surroundings, interacting with the environment and with other agents. An important characteristic of SI systems is the self-organization: with the low-level interactions within the agents, the swarm is capable of providing global responses. Among the most studied algorithms in this area one can mention the Ant Colony Optimization (ACO) [19] and the Particle Swarm Optimization (PSO) [20].

Some social species of bees clearly present characteristics and principles related to swarm intelligence, making them a good inspiration to optimization algorithms. Some of the main collective behaviors of bee colonies during foraging that can be highlighted as inspiration for the design of algorithms are as follows [21]: (1) bees dance to recruit nestmates to a food source; (2) bees adjust the exploration and recovery of food according to the colony state; (3) bees exploit multiple food sources simultaneously; (4) there is a positive linear relationship between the number of bees dancing and the number of bees recruited; (5) recruitment continues until a threshold number of bees is reached; (6) the quality of the food source influences the bee dance; and (7) all bees retire at some point in time, meaning that bees stop recruiting other bees. For generic reviews of bee-inspired algorithms and applications, the reader is invited to refer to Karaboga and Akay [22], Bitam et al. [23], Ruiz-Vanoye et al. [24], Verma and Kumar [25], Karaboga et al. [26], and Agarwal et al. [27].

This paper starts by bringing a chronological review on bee-inspired algorithms applied to vehicle routing problems. The review includes a brief description of the key biological mechanisms of the bee metaphor and a mathematical description of vehicle routing problems. It then follows with a taxonomy of bee-inspired algorithms and a description of the four standard approaches, presenting the main types of (bee) agents in the metaphor, a simplified pseudocode of the algorithm, and a brief description of its main steps. The review of the papers focuses on the problem solved, the types of bee agents or base algorithm used, how the algorithms were assessed, and comments on their overall performance.

The last part of the paper describes a new bee-inspired proposal, named TSPoptBees, originally designed to solve continuous optimization problems [21] and then adapted to solve VRPs, more specifically the travelling salesman problem. The proposed algorithm is applied to 28 TSP instances and its results are compared to the best-known solutions from the literature and with the results of many other bee-inspired algorithms cited in the review part of the paper. The paper is concluded with general comments and a proposal of how to extend the algorithm to the other vehicle routing problems.

The remainder of this paper is organized as follows. Section 2 provides a brief introduction to bee colonies and vehicle routing problems; Section 3 describes the base algorithms used in the literature to solve vehicle routing problems; and Section 4 briefly reviews each work found in the literature related to bee-inspired algorithms to solve vehicle routing problems. The proposed algorithm is detailed in Section 5 and its performance is assessed in Section 6. The paper is concluded in Section 7 with general comments and perspectives for future research for the area.

2. Fundamentals of Bee Colonies and Vehicle Routing Problems

This section provides a brief introduction to the use of bee colonies as metaphors for swarm intelligence and a standardized mathematical description of vehicle routing problems, to be used over the whole paper.

2.1. Bee Colonies as Metaphors for Swarm Intelligence. A notable feature of swarm intelligence is self-organization, whereby the system is able to present responses at a global level through the low-level iterations among the agents themselves and the environment. Bonabeau et al. [17] highlight four characteristics of self-organization in swarms: positive feedback, through simple behavioral rules that promote the creation of appropriate structures; negative feedback, which acts as a counterbalance to positive feedback; oscillations, such as random behaviors, errors, and task switching; and multiple interactions among the agents and the environment, allowing the exchange of information. Millonas [28] highlighted five principles for a swarm to present intelligent behavior: proximity (individuals should be able to perform simple tasks); quality (the swarm must be sensitive to quality factors); diversity (the swarm should not allocate all its resources in a single medium, but distribute them); stability (individuals should not change their behavior in response to all changes in the environment); and adaptability (the swarm must be able to change its behavior when necessary).

Some species of bees clearly present the characteristics described by Bonabeau et al. [17] and the principles defined by Millonas [28], which motivates the use of bee colonies as metaphors for the design of algorithms for solving complex problems. Karaboga and Akay [22] highlighted some tasks performed by bee colonies that are most used in the literature as metaphors for swarm intelligence, as well as the clear division of tasks between bee types:

(i) *Queen bee*: it is the only female of the colony that lays eggs, being the progenitor of all the other bees in the colony. It can live for several years, mating only once. Fertilization can occur for two or more years, using sperm stored during mating. When there is a lack of food sources, it produces more eggs and when the colony is very populated it stops producing them. After consuming the sperm in its spermatheca, it produces eggs that have not been fertilized and one of these descendants will become the new queen.

(ii) *Drones*: they are the male bees of the hive, with reproductive role, being considered the progenitors of the other bees of the colony. They are produced from unfertilized eggs and are fed differently when in the larval stage. Depending on the period there may be hundreds of drones in the colony; however they do not live more than six months. Their main task is to mate with the queen, dying after that.

(iii) *Workers*: they are responsible for various operational tasks of the hive, such as collecting and storing food, removing debris, and protecting the hive. They can

live for a few weeks or months and the tasks to which they are allocated depend on their age and the colony needs. In general, in the second half of their lives they forage for food.

(iv) *Mating flight*: the mating of the queen takes place in the air during the so-called mating flight. This starts with a dance performed by the queen and the drones follow her. The mating between the queen and a drone is probabilistic, according to the speed of the queen's flight and the fitness of the drone and queen. The drone sperm is stored in the spermatheca of the queen and can be used in the descendants fertilized by the queen.

(v) *Foraging*: this is one of the most important tasks for the hive. External factors (odor, location, and presence of other bees in the food source) and internal factors (souvenir and odor of the location) on bees influence this process. It begins with the worker leaving the hive in search of the food source. After finding this source, the nectar is collected and stored in its stomach, and after returning to the nest it deposits the nectar in the combs.

(vi) *Dancing*: dancing is the way bees inform others of good food sources. After unloading the nectar, the bee that found an attractive food source performs a series of movements, called waggle dance. Information such as quality, direction, and distance from the source of food is passed through the dance.

2.2. Optimization and Vehicle Routing Problems. Optimization consists of determining the values of a set of variables that minimize or maximize a given mathematical expression, satisfying all problem constraints [2, 3]. Perhaps the most intuitive way to solve a given optimization problem is to list all possible solutions, evaluate them, and use the best solution. However, this approach, known as brute force, is not efficient depending on the characteristics of the problem. The main drawback in using full enumeration is that it becomes computationally impractical depending on the number of possible solutions to the problem. This means that this exact solution approach would be valid only for simpler problems, which hardly occurs in practical applications.

An example of an optimization problem commonly used in the literature is the travelling salesman problem (TSP) [5, 7, 9]. In a simple way, TSP can be described as follows: given a set of cities, the salesman should visit every city once, coinciding the initial city with the final one, so that the cost of the path travelled is minimal. For the asymmetric TSP, in which the distance between a city A and another city B can be different from the distance between B and A, the number of possible solutions is $(n-1)!$, where n is the number of cities. This means that the number of possible solutions has a factorial growth with the size of the problem.

The TSP can be mathematically described as follows. Given a set of n cities and the cost $\mathbf{C}_{ij}$ $(i, j = 1, 2, 3, \ldots, n)$ of going from city i to city j, the TSP aims at determining a permutation of $\boldsymbol{\pi}$ the cities that minimize

$$\sum_{i=1}^{n-1} \left(\mathbf{C}_{\boldsymbol{\pi}_i \boldsymbol{\pi}_{i+1}}\right) + \mathbf{C}_{\boldsymbol{\pi}_n \boldsymbol{\pi}_1}. \tag{1}$$

TSP represents one of the elementary vehicle routing problems and, despite its simple description, its exact solution becomes complex because of the computational cost required, being part of the class of NP-hard problems [29]. Due to its academic importance and wide application in practical problems, TSP has been receiving great attention for more than 60 years [5, 10]. As a result, well-consolidated TSP-oriented review works can be found in the literature [6, 9, 29], which present formulations, examples of practical applications, and classical solution algorithms.

Vehicle routing problems closer to practical applications appear with the addition of some constraints to the TSP. The Multiple Travelling Salesman Problem (MTSP) is a generalization of TSP in which more than one salesman is used in the solution and a common city, the depot, is used as starting and ending point by all the salesmen. Given a number M of salesmen, a depot i_d, a set $\mathbf{S} = \{i_1, i_2, i_3, \ldots, i_n\}$ with n intermediate cities, and the cost $\mathbf{C}_{ij}$ $(i, j = 1, 2, 3, \ldots, n+1)$ of going from one city to another and also from each city to the depot, the MTSP consists of determining the set of permutations $\boldsymbol{\pi}_m$ $(m = 1, 2, 3, \ldots, M)$ from the elements in $\mathbf{S}$ so as to minimize

$$\sum_{m=1}^{M} \left(\mathbf{C}_{i_d, \boldsymbol{\pi}_{m,1}} + \sum_{i=1}^{n_m - 1} \mathbf{C}_{\boldsymbol{\pi}_{m,i}, \boldsymbol{\pi}_{m,i+1}} + \mathbf{C}_{\boldsymbol{\pi}_{m,n_m}, i_d} \right), \tag{2}$$

where $\boldsymbol{\pi}_{m,i}$ represents the city visited in order i in route m and n_m is the number of intermediary cities visited in route m. The MTSP already presents enough characteristics to represent practical problems, such as school vehicle routing [30]. Bektas [31] presented other examples of practical applications and algorithms for MTSP.

The addition of some constraints in MTSP leads to another classical problem, the capacitated vehicle routing problem (CVRP). The CVRP can be seen as an MTSP with two modifications: (1) each intermediate city represents a customer with a certain demand for a product and (2) each salesman represents a vehicle with a limited capacity. Given a number K of identical vehicles, each with a capacity Q, a depot i_d, a set $S = \{i_1, i_2, i_3, \ldots, i_n\}$ with n customers, each with a demand q_i $(i = 1, 2, 3, \ldots, n)$, and the cost $\mathbf{C}_{ij}$ $(i, j = 1, 2, 3, \ldots, n+1)$ of going from customer i to customer j and from each customer to the depot, the CVRP consists of determining the permutation $\boldsymbol{\pi}$ of the elements in S so as to minimize

$$\sum_{k=1}^{K} \left(\mathbf{C}_{i_d, \boldsymbol{\pi}_{k,1}} + \sum_{i=1}^{n_k - 1} \mathbf{C}_{\boldsymbol{\pi}_{k,i}, \boldsymbol{\pi}_{k,i+1}} + \mathbf{C}_{\boldsymbol{\pi}_{k,n_k}, i_d} \right), \tag{3}$$

subject to

$$\sum_{i=1}^{n_k} q_{\boldsymbol{\pi}_{k,i}} \leq Q \quad \forall k, \tag{4}$$

Table 1: Name of the base algorithm inspired by different behaviors of bees.

Honeybees behavior	Algorithm name
Foraging	Artificial Bee Colony (ABC)
	Bee Colony Optimization (BCO)
	Bee System (BS)
Marriage	Marriage in Honeybees Optimization (MHBO)

where $\pi_{k,i}$ represents the city visited in order i in route k and n_k is the number of customers visited by vehicle k. For a review of practical applications and solution algorithms the works of Laporte et al. [11] and Laporte [10] are suggested.

3. Bee-Inspired Algorithms: A Taxonomy and Standard Approaches

In the literature, there are basically four different algorithms forming the basis of all bee-inspired approaches for solving *vehicle routing problems* (VRPs). Thus, the works to be reviewed here are either the proposal of one algorithm or a modification of it to solve a different problem or to improve its performance. The domain of bee-inspired algorithms is not constrained to the ones presented in this section, but these are the ones so far used to tackle vehicle routing problems. As shown in Table 1, these algorithms can be divided into two subclasses based on the bees' behavior used as inspiration [23]. Each of the base algorithms is described in the following subsections.

3.1. Artificial Bee Colony. The Artificial Bee Colony (ABC) algorithm was introduced by Karaboga [32] and later extended by Basturk and Karaboga [33] to solve numeric function optimization problems. This algorithm is inspired by the honeybees' foraging behavior. The first modification of the ABC to solve a vehicle routing problem was proposed by Banharnsakun et al. [34] and the problem chosen was the *travelling salesman problem* (TSP). This section describes the ABC algorithm as a general-purpose optimization model, as in Karaboga et al. [26]. The modifications proposed in each work to solve the vehicle routing problem are described in Section 4. The ABC algorithm works with three kinds of agents (bees):

(i) *Employed bees*: bees already associated with a food source, which represents a solution

(ii) *Onlooker bees*: bees that will observe the dance of employed bees and follow them to the neighborhood of their food sources

(iii) *Scout bees*: bees that wander around the search space to find new food sources; this can be a random or guided process.

Each of these agents is used during specific phases of the algorithm, as presented and summarized in Box 1.

```
Initialization
while stopping criterion is not met
  Employed bees phase
  Onlooker bees phase
  Scout bees phase
End
```

Box 1: Structure of the ABC algorithm and its main phases.

3.1.1. Initialization. The initial food sources are discovered by scout bees. The number of food sources is the same as the number of employed bees in the hive. Each food source is then associated with an employed bee.

3.1.2. Employed Bees Phase. During this phase, an employed bee will search for better food sources in its neighborhood. Then, based on some neighborhood function, a greedy selection is applied between the previous and the new food source. If a better food source is found within this neighborhood, then this becomes the current food source associated with this employed bee.

3.1.3. Onlooker Bees Phase. During this phase, employed bees advertise their food sources to the onlooker bees by the means of the waggle dance. Onlooker bees will observe the dance of employed bees and probabilistically choose one of them to follow until their food source. Karaboga et al. [26] suggest the roulette wheel as a way to accomplish this selection, using the quality of the food source as the fitness for this process. After defining which employed bee to follow, the onlooker bee will look for good food sources in the neighborhood of the food source of the employed bee. The same operation for the employed bees phase is used here.

3.1.4. Scout Bees Phase. During this phase, an employed bee will become a scout bee if its food source is not improved during a determined number of iterations. Then, the employed bee will abandon its food source and look for a new one.

3.2. Bee Colony Optimization. The Bee Colony Optimization (BCO) algorithm is a model inspired by the foraging behavior of honeybees. It is based on the work proposed by Nakrani and Tovey [35] to solve a server allocation problem in datacenters. Two papers were the first to propose similar modifications to the original algorithm to solve a vehicle routing problem: Wong et al. [36] to solve the TSP and Lu and Zhou [37] to solve the TSP.

Lu and Zhou [37] named their algorithm as Bee Collecting Pollen Algorithm, while in Wong et al. [36] they termed their algorithm as BCO. Besides having different names, the main concepts are very similar and the base work is the same. Here we use the proposal of Wong et al. [36] to provide an overview of the algorithm. The main steps of the BCO algorithm are summarized in Box 2 and detailed in the sequence.

```
Initialization
while stopping criterion is not met
  Observe Waggle Dance
  Perform Foraging
  Perform Waggle Dance
End
```

Box 2: Structure of the BCO algorithm and its main steps.

3.2.1. Initialization. Each bee represents a candidate solution and is created by using a nearest neighbor or a random approach.

3.2.2. Observe Waggle Dance. Before leaving the hive for the foraging process, a bee will decide if it will observe the waggle dance of other bees. A bee is more likely to observe the dance of other bees if its own solution is not as good as the average solution of the hive. By observing the waggle dance, a bee will select another one to follow. The solution of this selected bee will become its preferred solution, to be used during the foraging process. During the first iteration, as bees have no dance to follow, they are allowed to explore the search space without a *preferred solution*.

3.2.3. Perform Foraging. During this process, a bee will iteratively construct a full solution to the TSP by moving from one city to another. A bee selects the next city to visit based on a probabilistic transition rule that takes into account the distance between the two cities and if this arc is present in its preferred solution.

3.2.4. Performing Waggle Dance. The bee that finds a solution better than its previous one will advertise the new route by means of a waggle dance. The dance has a duration defined by the quality of the solution found by the bee and average quality of the hive.

3.3. Bee System. The Bee System (BS) algorithm was initially proposed by Lučić and Teodorović [38] to solve transportation problems. This algorithm is inspired by the honeybees foraging process. The main phases of the BS algorithm are presented in Box 3, as proposed in Lučić and Teodorović [39] to solve the TSP. The original BS algorithm evolved [40] and, together with some structural changes, the authors proposed a new name as the Bee Colony Optimization (BCO). For the present review, the new algorithm is kept under the BS structure and the modifications are discussed in Section 4.

3.3.1. Changing the Hive Location. At each iteration, the hive location is changed by placing it on a different city. The hive will be the initial city for each bee. The main idea behind this is that the initial city of the tour being constructed will be different at each iteration.

3.3.2. Stages. A stage is considered to be a discrete time unit in the bee's environment. Each iteration has a predetermined number of stages on which the bees will perform some of their tasks.

```
while stopping criterion is not met
  change the hive location
  for each stage
    perform the foraging process
    abandon the food source
    waggle dance
  end for
  local-search
end while
```

Box 3: Structure of the BS algorithm and its main steps.

3.3.3. Foraging Process. At each stage, a bee is allowed to visit a predetermined number of cities during the foraging process. This means that, at each stage, it iteratively constructs subtours until reaching a complete solution for the TSP. Not all bees start to forage at the same stage. Those that do not start at the initial stage will follow other bees when they decide to leave the hive. Bees construct subtours by visiting one city after another. The next city to be visited is chosen based on a probabilistic function with the following rules: the greater the distance between the cities, the lower the probability; the greater the number of iterations already spent, the higher the influence of the distance; the greater the number of bees that visited the same link in the past, the higher the probability.

3.3.4. Abandoning the Food Source. A bee will decide if it should continue to forage from the same location in the next stage or if it will abandon its subtour. This is a probabilistic function based on the quality of the subtour: the greater the distance, compared with the solutions found by other bees, the higher the probability of abandoning the current solution.

3.3.5. Waggle Dance. Bees that have chosen to keep their food sources will now decide if they will recruit other foragers by means of the waggle dance. The decision to dance to advertise its solution is a probabilistic function with low possibility of not dancing. If a bee decided to abandon its solution, it will start to follow another forager bee. It will observe the waggle dance from other bees to decide which one to follow. This decision is guided by a probabilistic function: the lower the cost of the advertised subtour, compared with other solutions, the higher the probability of using it; the higher the number of bees advertising the same subtour, the higher the probability.

3.3.6. Local Search. At the end of each iteration, the solution of each bee undergoes a local search. 2opt or 3opt is used, based on the size of the TSP instance.

3.4. Marriage in Honeybees Optimization. The Marriage in Honeybees Optimization (MHBO) algorithm was introduced in 2001 by Abbass [41]. In this work, the author

```
Initialization
while stop criteria is not met
  Perform mating fight
  Generate broods
  Improve broods
  Determine the new set of queens
End
```

Box 4: Structure of the MHBO algorithm and its main steps.

described a general-purpose algorithm inspired by the marriage behavior of honeybees, which was then adjusted and tested to solve the SAT problem.

The first works to propose modifications on the MHBO to solve a vehicle routing problem were Yang et al. [42, 43], solving the TSP, and Marinakis et al. [44], solving the CVRP. These works are described in the following section. Here we describe, at a high level, the main steps of the original general-purpose algorithm [41]. The algorithm has four types of agents (bees), each with a different task during the optimization process:

(i) *Queens*: representing the best solutions found so far and used for crossover

(ii) *Drones*: solutions across the search space that are used for crossover

(iii) *Broods*: resulting solutions from crossover and mutation operations between queens and drones

(iv) *Workers*: set of heuristics used to improve broods.

The main steps of the algorithm are highlighted below and the whole process is then summarized in Box 4.

3.4.1. Initialization. During the initial phase, a set of queens is randomly generated and a random worker is applied to each of them.

3.4.2. Mating Flight. The queen starts the mating flight with an initial energy and speed. At each position in the flight, the queen encounters a different drone, which represents a randomly generated solution. The queen decides if it will mate with the drone based on a probabilistic function as follows: the higher the drone fitness, the higher the probability; the higher the speed, the higher the probability. If the drone is selected for mating, its sperm is added to the queen's spermatheca. A drone sperm represents only part of its solution. During the mating flight, the queen's energy and speed decrease and the mating flight is over when the queen's energy is below a given threshold. The mating flight occurs for every queen in the hive.

3.4.3. Generating Broods. Broods are generated by applying crossover and mutation operators between the queens and sperms in their spermathecal.

3.4.4. Improving the Broods. During this phase, a worker is used to improve each brood. A fitness is assigned to each worker, which guides their selection: the worker that makes the higher improvement in the broods is more likely to be selected. This fitness is updated throughout the search process based on the quality of the improvement performed on each brood.

3.4.5. Determining the New Set of Queens. If the best brood is better than the worst queen, then this brood replaces this queen. This occurs until the best brood is no better than the worst queen.

4. Bee-Inspired Algorithms to Solve Vehicle Routing Problems: A Survey

Works relating bee-inspired algorithms with vehicle routing problems are described in this section. They are segmented depending upon the base algorithm, as described previously, and chronologically ordered.

4.1. Artificial Bee Colony. In Banharnsakun et al. [34], the authors proposed the first modification found in the literature for the ABC to solve a vehicle routing problem, targeting the TSP. An array of permutations of integers is used to represent each candidate solution. During the employed and the onlooker bees phases the authors used a recombination operator called Greedy Subtour Crossover and applied the 2opt local search after a new solution is created. Random solutions were created during the scout bees phase. Tests were conducted using instances from the TSPLIB, ranging from 51 to 318 cities. Results were compared with other works from the literature and the proposed method achieved better solutions.

Karaboga and Gorkemli [45] proposed an ABC algorithm to solve the TSP. The initial solutions were created using the nearest neighbor approach. During the employed and the onlooker bees phase the authors proposed the usage of a method initially proposed as a mutation operator for a genetic algorithm. The algorithm was tested in two instances from the TSPLIB, with 150 and 200 cities. The results were compared with the ones from a genetic algorithm. Although the proposed algorithm was not able to find the best-known solution in any run, it outperformed the genetic algorithm.

In Singh et al. [46], the authors proposed a hybrid ABC and a genetic algorithm to solve the TSP. Candidate solutions are represented using an array of real numbers and operators are chosen based on this representation. Recombination and mutation phases are added before the employed bees phase. The algorithm was tested using 2 randomly created instances with 30 and 60 cities, respectively. The proposed algorithm was able to produce better solutions than an authors' implementation of the genetic algorithm.

Zhang et al. [47] combined the ABC algorithm with the path relinking method to solve the TSP. During the employed and onlooker bees phase, 2opt is used for any new created solution and path relinking is used whenever a better solution is found. The algorithm was tested in 11 instances from the

TSPLIB, ranging from 51 to 127 cities. Results are compared with those published in other works, specially related to algorithms inspired in the self-organizing maps, and the proposed algorithm outperforms the other ones in regard to the quality of the solution.

In Szeto et al. [48], the capacitated vehicle routing problem (CVRP) was resolved by using the ABC. Candidate solutions are represented with an array of integers, with the depot appearing multiple times. Infeasible solutions are allowed during the search process and a penalty is added to the quality of the solution when a constraint is violated. Initial solutions are created by assigning customers to vehicles by choosing the one that would impose the least cost to the overall solution. During the employed and onlooker bees phase, neighboring solutions are created by using one out of 7 possible operators. Solutions found by the onlooker will not replace its employed bee but the one which is not improved for a long time. Solutions related to scout bees are not randomly generated but are neighbor of the previous solution. Tests were conducted using 34 benchmark instances from the TSPLIB, ranging from 50 to 483 customers. The obtained results are good, with the algorithm being able to find the best-known solutions on many runs for different instances.

The Green TSP (GTSP) was resolved in Özceylan et al. [49] with an algorithm hybrid between the Ant Colony Optimization (ACO) and the ABC. ABC is used to determine the best speed between two visited customers. Tests with the hybrid algorithm were performed with 36 instances proposed by the author, ranging from 5 to 40 cities. Results were compared with those proposed by the software LINGO. The proposed algorithm provides better solutions.

Ji and Wu [50] tested the ABC algorithm with different mutation operators as the neighboring operator for the employed and onlooker bees phases: random swap; random insertion; random swap of subsequence; random insertion of subsequence; random reversing of subsequence; random reversing swap of subsequence; and random reversing insertion of subsequence. 2opt and 3opt local search heuristics are used at the end of the process to improve the final solution. Based on the tests made, the best results were obtained with the insertion operators.

In Li et al. [51], the authors proposed a hybrid algorithm combining the ABC algorithm with a nearest neighbor method to solve the TSP. To assess the performance of their proposal they applied the algorithm to a number of instances from the TSPLIB and compared it with the BCO algorithm, showing a superior performance for the instances evaluated.

In Brajevic [52], the author proposed some modifications in the ABC algorithm so that it could be applied to the capacitated vehicle routing problem. The modifications include the proposal of a specific encoding scheme, constraint handling, and specific neighborhood operators. The method was applied to 12 small-scale benchmark CVRP instances and the results were compared to those of the best known solutions.

Shi et al. [53] proposed the ABC-T algorithm for solving the VRPTW. ABC-T is based on the standard ABC but uses a tournament selection mechanism to select food sources, improving the global search capability of the algorithm. Their proposal was assessed using the Solomon's R102 problem instance.

Iqbal and Rahman [54] tackled the Vehicle Routing Problem with Time Windows (VRPTW) using the ABC algorithm. Candidate solutions are represented with an array of integer numbers, where the depot appears as much times as vehicles are used in the solution. Infeasible solutions are allowed during the search, but this implies penalties to the quality of the solution. During the employed and onlooker bees phase, neighboring solutions are created by swap mutation. Also, when a better solution is found during these phases, it does not replace the original solution but the worst one in the population that has not been improved for some time. Onlooker bees create new solutions by using the swap operator on the solution to be discarded. Tests were conducted on 28 randomly generated instances proposed by the authors, ranging from 20 and 300 customers. The modified algorithm was compared with the original one and the new implementation was able to provide better solutions.

In Karabulut and Tasgetiren [55], a modified ABC algorithm was proposed to solve the TSP with Time Windows (TSPTW). Initial solutions were randomly constructed. In the employed and onlooker bees phase, neighboring solutions were chosen using a destruction and construction method and then undergoing a local search. For the onlooker bee phase, a new solution was constructed using the same method as the employed bees. The proposed algorithm was tested in 55 benchmark instances from the literature, ranging from 20 to 200 cities. The results were compared with other two works from the literature, presenting competitive solutions.

Bhagade and Puranik [56] proposed an ABC algorithm to solve the TSP. There is no description about the necessary modifications in the original algorithm. The proposed algorithm was tested in 4 instances created by the authors. The results were compared with another unmentioned method, which was outperformed by the proposed algorithm.

Pathak and Tiwari [57] tackled the TSP by using the original version of the ABC. Since candidate solutions are represented using arrays of real valued numbers, a map function is required to translate this array into a TSP solution. For that purpose they use a technique called shortest position value. Tests were conducted using 2 instances proposed by the authors, with 30 and 60 cities, respectively. Results obtained by the proposed algorithm were better than an authors' implementation of a genetic algorithm.

In Li et al. [58], the authors proposed the use of a swap operator in the ABC algorithm to help the bees in finding better candidate tours for the TSP based on a greedy search. To assess the performance of their algorithm they used 6 TSP instances from the literature and compared the results with the PSO algorithm. They also performed a sensitivity analysis of some ABC algorithm parameters.

In Bin et al. [59], the Vehicle Routing Problem with Soft Time Windows (VRPSTW) was resolved using the ABC algorithm. A candidate solution is represented using a set of arrays with integer values, on which each array represents the route of a vehicle. Initial solutions are randomly created. During the employed and onlooker bees phases,

neighborhood solutions are created by performing inter- and intraroute modifications. Tests were conducted on 56 benchmark instances from the literature. The results are compared with other 5 algorithms from the literature. For most of the tests, the proposed algorithm outperforms the other ones.

Kiran et al. [60] modified the structure of ABC algorithm adding a crossover step between the employed and the onlooker bees phase. The modification is mainly described to tackle continuous optimization problem and later they suggest the same structure to deal with the TSP. Although tests were made and results were posted to the TSP, the specific modifications to deal with this problem are not clear. Tests were conducted with 3 instances proposed by the authors, ranging from 10 to 30 cities, and the proposed algorithm was able to find solutions better than the original algorithm.

Sabet et al. [61] proposed an implementation of the ABC to solve the TSP. Initial solutions are built by using a probabilistic implementation of the nearest neighbor. Mutation operators are used to create neighboring solutions during the employed and onlooker bees phases. Tests were conducted using 5 instances of the TSPLIB, ranging from 51 to 99 cities. The algorithm was capable of finding good results but did not outperform the results from other algorithms from the literature chosen by the authors.

In Yao et al. [62] the authors proposed modifications on the ABC algorithm in order to solve the Periodic Vehicle Routing Problem (PVRP). Recombination and mutation operators are used in order to define new solutions during the search process. 2opt is used to improve the routes found. Tests were made on 9 benchmark instances from the literature with the following characteristics: 50 to 100 customers; 1 to 6 vehicles; and periodic cycles from 2 to 10 days. Results were compared with those published in other works and the algorithm has been shown to be competitive in regard to the quality of the solutions found.

Liu [63] proposed a merge between an Adaptive GA (AGA) and the ABC algorithm to solve the Multidepot Vehicle Routing Problem with Time Window (MDVRPTW). In the description of the algorithm, it is not clear how the algorithm is improved with the ABC concepts. Tests are conducted with 20 benchmark instances from the literature and results are good.

In Singh et al. [46], a solution based on the ABC algorithm is proposed to the 2-dimensional Loading Capacitated Vehicle Routing Problem (2L-CVRP). Candidate solutions are represented using an array of real numbers, which requires a decode process to validate the quality and feasibility. 2opt and 3opt local search algorithms are used to improve the routes. A set of 3 different heuristics are used to deal with the 2-dimensional loading problem. Tests were conducted using 50 benchmark instances from the literature with the following characteristics: 15 to 100 customers; 15 to 310 demanded goods. In most of the tests the proposed algorithm outperformed others from the literature.

Yang and Pei [64] combined the ABC algorithm with the PSO to solve the travelling salesman problem but used PSO as the main algorithm in the problem solution due to its claimed faster convergence.

Pandey and Kumar [65] introduced some enhancements by adding crossover operators to the ABC algorithm and applied it to the TSP, evaluating its performance in terms of efficiency and accuracy. Their hybrid algorithm works in five phases: initialization; employed bee phase; crossover; onlooker bee phase; and scout bee phase. They applied their algorithm for different values of the maximum cycle number and different dimensions of the individuals in the population.

Rekaby et al. [66] proposed an Adaptive Artificial Bee Colony (AABC) algorithm applied to the travelling salesman problem.

In Yuan and Zhu [67], the authors proposed a hybrid ABC with a genetic algorithm to solve the TSP. It is not clear how each phase of the ABC algorithm was modified to tackle the TSP. The GA inspiration was used as an additional step by adding crossover and mutation operators after the scout bees phase. The algorithm was tested using an instance with 30 cities from the TSPLIB. The result was compared with other 5 works from the literature and the proposed algorithm outperformed the other ones.

In Zhang et al. [68], the Environmental Vehicle Routing Problem (EVRP) is resolved using a hybrid algorithm between ABC and genetic algorithms. Employed and onlooker bees phases are modified by adding crossover and mutation operators. Also, 2opt and 3opt are used to improve the routes created. Tests were conducted with 19 benchmark instances from the literature, ranging from 31 to 51 customers and 3 to 7 vehicles.

In Nahum et al. [69], the authors proposed modifications in the ABC to solve the Multiobjective VRPTW (MOVRPTW). The authors proposed a Vector Evaluated ABC (VEABC) so that multiobjective problems can be properly solved. The vector evaluated approach is borrowed from genetic algorithms, where portions of the population are evaluated on each objective. Candidate solutions are represented as an array of integers, with the depot appearing multiple times. During the employed and onlooker bees phase, neighboring solutions are created by using an operator chosen from a pool of 9 different ones. Tests were conducted on 55 benchmark instances from the literature. The algorithm was capable of finding good solutions with regard to the cost of routes and number of vehicles.

Kocer and Akca [70] used a local search method to improve the performance of the ABC algorithm for solving the TSP. They used a loyalty function of BCO as a fitness function for the ABC algorithm, applied their proposal to TSP instances from the TSPLIB, and compared their results with those of other bee-inspired algorithms.

In Chung [71], the authors combined the ABC algorithm with a sweep method to quickly generate near optimal solutions to the capacitated vehicle routing problems. His proposal was compared with a genetic algorithm over 60 instances containing 100 nodes.

In the proposal of Nagaya and Inoie [72], the authors created a hybrid between the ABC and the simulated annealing (SA) algorithm to solve capacitated vehicle routing problems. They assessed their algorithm in 11 benchmark instances and

compared the results with that of the standard ABC, SA, and the best-known solution.

Zhang and Lee [73] introduced a routing directed ABC (RABC) algorithm for solving the capacitated vehicle routing problem (CVRP) and tested their proposal in various benchmark problems from the literature.

Gündüz et al. [74] proposed a hierarchical approach based on the Ant Colony Optimization (ACO) and the ABC algorithms to solve the TSP. In their approach the ACO is used as a path-construction method and the ABC as a path-improvement method. As such, the ACO provides a better initial solution for the ABC algorithm, which becomes responsible for seeking the optimal solution. The authors used 10 TSP instances from the literature and showed that their hybrid proposal achieves better quality solutions than the individual use of ACO or ABC, with less computational time.

Alzaqebah et al. [75] used a modified ABC algorithm for solving Vehicle Routing Problems with Time Windows (VRPTW). Their modification included the use of abandoned solutions by scout bees, a roulette wheel selection of candidate solutions, and the generation of new random candidate solutions to be inserted in the swarm. The proposal was applied to the Solomon benchmark datasets and the results were compared with that of the standard ABC.

4.2. Bee Colony Optimization. One of the first proposals using the BCO algorithm to resolve a vehicle routing problem, the TSP, is described in Lu and Zhou [37]. There are only a few specificities from this work to the algorithm described in the previous section: during the iterations, the starting position of the bees changes randomly; new bees are introduced in the population during the search process. Tests are made using two instances from the TSPLIB, with 51 and 70 cities, respectively.

Wong et al. [36] is one of the first works to relate the BCO to a vehicle routing problem, the TSP. This is the work used in the previous section to describe the BCO algorithm. Tests were conducted using 15 instances from the TSPLIB, ranging from 48 to 318 cities. Results were compared with other 6 works from the literature which was unable to provide better results.

In Wong et al. [76–79], the authors tackled the TSP by proposing some modifications on Wong et al. [36] to improve the efficiency of the algorithm. The modification resides on the local search used after the foraging phase. A method named frequency-based pruning strategy (FBPS) chooses which solutions should undergo a local search, which is an efficient implementation of the 2opt, called fixed-radius near neighbor (FRNN) 2-opt. Tests were made using 84 instances from the TSPLIB, ranging from 14 to 1379 cities. The results are good, decreasing the computational time without compromising the quality of the solution. The authors make an extensive comparison of the quality of the solutions with other works from the literature and the proposed algorithm is capable of finding competitive solutions. A generic framework for combinatorial optimization problems, including vehicle routing problems, is proposed in Wong et al. [78, 79].

In Wong et al. [77], the authors expanded the work in Wong et al. [76] to solve the TSP, by changing the transition rule in order to make it quicker, in terms of computational time. Instead of visiting one city at a time, the bee is now allowed to visit a set of cities (a subroute). The proposed algorithm is tested on 84 instances from the TSPLIB, ranging from 14 to 1379 cities. The modification decreased the running time without compromising the quality of the solution.

In Häckel and Dippold [80] the authors proposed a two-stage BCO-like algorithm to tackle the Vehicle Routing Problem with Time Windows (VRPTW). During the first stage it determined the number of vehicles and the cities to be visited. The route of each vehicle was created during the second stage. The proposal was tested using 56 benchmark instances from the literature, with 100 customers each. The results were compared with other works from the literature and the proposed algorithm was not able to find competitive solutions, in comparison to other algorithms.

In Özceylan et al. [49], the authors proposed a modification on the BCO to solve the TSP. 3opt local search is added after the foraging phase in order to improve the routes built. Tests were conducted with 16 instances proposed by the authors and the results were compared with those from the nearest neighbor to represent the improvement made by the proposed algorithm.

In Girsang et al. [81], the authors proposed modifications in the BCO to solve the TSP. The modifications were to enhance the computational performance by using a method called pattern reduction. Two pattern reduction operators were added and the modified algorithm was compared with the original BCO in 7 instances from the TSPLIB, ranging from 51 to 1002 cities. The modifications significantly reduced the computational time with a small reduction in the average quality of the solutions.

4.3. Bee System. In Lučić and Teodorović [39] and in Lučić and Teodorović [82], the authors proposed the Bee System algorithm to solve the TSP. These are the works referred in the previous section to describe the main steps of this algorithm. In Lučić and Teodorović [39], tests were conducted using 8 instances from the TSPLIB, ranging from 51 to 280 cities. Results were good, with the algorithm being able to find the best-known solution for 6 instances, with size lower than 102 cities. In Lučić and Teodorović [82], the tests were conducted on 10 instances, ranging from 51 to 1002 cities.

In Lučić and Teodorović [83], the authors expanded the work in Lučić and Teodorović [39] so that the BS is applicable to the Stochastic Vehicle Routing Problem (SVRP). The proposal uses the original BS to create a "giant tour," such as a solution to the TSP. Based on fuzzy rules, the giant tour is then divided into subtours, one for each vehicle. TSP instances from the TSPLIB were used to create instances for the stochastic VRP. The authors used 10 instances, ranging from 51 to 1002 customers. The algorithm was able to find good average solutions.

In Teodorovic et al. [40], the Bee System is described as a variation of a concept the authors call Bee Colony Optimization (BCO). The BS is slightly modified so that all the operations are conducted within two major steps: (1)

forward pass and (2) backward pass. During the forward pass bees will construct partial solutions and in the backward pass bees will communicate with each other and decide if they will continue with their own solutions or become followers. The algorithm was tested on 6 TSP instances ranging from 51 to 280 cities.

Nikolić et al. [84] follows the concept described in Teodorovic et al. [40] and proposes the use of the BCO to solve the VRPTW. In this implementation, the structure of the algorithm has evolved from a constructive to an improvement algorithm. In the current concept, each bee is associated with a solution which is iteratively modified targeting the improvement of the solution. The structure of the forward and backward passes is maintained. However, in the forward pass, instead of building a solution, modification operators are used targeting the improvement of the current solution. The algorithm was tested on Solomon's benchmark instances and the algorithm was capable of finding satisfactory solutions. The authors suggest the results could be improved by using local search techniques together with the proposed algorithm.

In Jawarneh and Abdullah [85], the authors proposed an improvement in the algorithm introduced in Nikolić et al. [84] to solve the VRPTW. The two main improvements are the use of a sequential insertion heuristic to create the initial population and the online tuning of parameter's values. The proposed algorithm was tested on Solomon's benchmark instances. A direct comparison made by the authors with the original algorithm showed that the proposed method is capable of finding better results.

4.4. Marriage in Honeybees Optimization. In Yang et al. [42], the authors proposed an MHBO algorithm to solve the TSP. The main addition highlighted by the authors is the usage of the Wolf Pack Search (WPS) heuristic to improve the queen and the drone solutions. The algorithm was tested on two instances from the TSPLIB, with 16 and 48 cities, respectively. The results were compared with the original algorithm without the WPS and an authors' implementation of a genetic algorithm. The proposed algorithm was able to find better solutions.

In Yang et al. [43], the authors tackled the TSP with an implementation of the MHBO. It uses the Nelder-Mead method as the heuristic for the local search phase. Tests were performed on 2 instances from the TSPLIB, with 16 and 48 cities, respectively.

In Marinakis et al. [86], the authors used the proposal in Marinakis et al. [44] to solve the TSP. The initial population is created using the Multiple Phase Neighborhood Search-GRASP (MPNS-GRASP). To generate new broods, a crossover inspired on the adaptive memory procedure is proposed, which selectively copies information from the queen and the drones. As the workers, they use an approach called Expanding Neighborhood Search, which applies a set of local search strategies to the target solution. The algorithm was tested on 74 instances from the TSPLIB, ranging from 51 to 85900 cities. From 51 to 783 cities, the algorithm has found the best solutions on all runs. For the other instances, the algorithm was capable of finding either the best solution or a very close one on all runs.

In Celik and Ulker [87], the authors applied the MHBO to solve the Asymmetric Travelling Salesman Problem (ATSP), that is, the one in which the cost from A to B is different from the cost from B to A. The algorithm was applied to three different instances of the ATSP and the results were compared with that of a genetic algorithm and a simulated annealing method. They showed that the MHBO algorithm was capable of finding the best-known solution in most of the runs.

A solution using the MHBO algorithm to the capacitated vehicle routing problem with three-dimensional loading constraints (3L-CVRP) is proposed in Ruan et al. [88]. The initial population is built by using Multiple Phase Neighborhood Search Greedy Randomized Adaptive Search Procedure (MPNS-GRASP) with the best solution being chosen as the queen. The mating of queen and drones is made using the 2-point crossover. As workers, local search algorithms are used to make inter- and intraroute changes to improve the overall solution. To tackle the loading part of the problem, 6 different loading heuristics are used to determine if packages can be loaded on vehicles. Tests were conducted on 27 instances from the literature, ranging from 16 to 101 customers. Results were compared with other 3 works from the literature and, in most cases, the proposed algorithm outperforms the other ones.

Table 2 presents a chronological summary of the works reviewed, organized in terms of the base algorithm and target problem.

5. TSPoptBees: A Bee-Inspired Algorithm for Solving the Travelling Salesman Problem

The optBees algorithm was originally proposed by Maia et al. [21, 90] to solve continuous optimization problems inspired by the collective behavior of honeybees during the foraging process. It was later applied to other optimization problems, such as data clustering [91] and the training of artificial neural networks [92].

The TSPoptBees is a modification of the original optBees algorithm to solve the travelling salesman problem. It was originally introduced in a short conference paper by the same authors [93] and we now bring a much more complete explanation of the algorithm, plus a new set of experiments to validate the proposal.

In the TSPoptBees, each bee represents a route to the TSP, which is a candidate solution to the target problem. The quality of the food source explored by a bee is related to the quality of the associated TSP route in such a way that the higher the quality, the lower the total travelling cost. As in the original algorithm, there are three types of bees: (1) *recruiters*, which are responsible for recruiting other bees to exploit promising regions of the search space; (2) *scouts*, that are responsible for exploring new regions of the search space; and (3) *recruited*, which follow the recruiter bees to exploit new solutions in their surroundings.

The bees fly around the space of candidate solutions searching for the high quality ones. According to the quality

Table 2: Reviewed works using bee-inspired algorithms to solve vehicle routing problems.

Base algorithm	Target problem	Year	Authors
Artificial Bee Colony (ABC)	TSP	2010	Banharnsakun et al. [34]
		2011	Karaboga and Gorkemli [45]
			Singh et al. [46]
			Zhang et al. [47]
			Li et al. [51]
		2012	Bhagade and Puranik [56]
			Pathak and Tiwari [57]
			Li et al. [58]
		2013	Kiran et al. [60]
			Verma and Kumar [25]
			Sabet et al. [61]
			Pandey and Kumar [65]
			Rekaby et al. [66]
			Yang and Pei [64]
		2014	Yuan and Zhu [67]
			Kocer and Akca [70]
		2015	Gündüz et al. [74]
	GTSP	2011	Özceylan et al. [49]
	TSPTW	2012	Karabulut and Tasgetiren [55]
	CVRP	2011	Szeto et al. [48]
			Brajevic [52]
		2014	Chung [71]
			Nagaya and Inoie [72]
		2015	Zhang and Lee [73]
	EVRP	2014	Zhang et al. [68]
	PVRP	2013	Yao et al. [62]
	TDVRP	2011	Ji and Wu [50]
	VRPTW	2012	Shi et al. [53]
		2016	Alzaqebah et al. [75]
	VRPSTW	2012	Iqbal and Rahma [54]
	MOVRPTW	2014	Nahum et al. [69]
	MDVRPTW	2013	Liu [63]
	2L-CVRP	2013	Bin et al. [59]
Bee Colony Optimization (BCO)	TSP	2008	Lu and Zhou [37]
			Wong et al. [36]
		2009	Wong et al. [76, 77]
		2010	Wong et al. [78, 79]
		2012	Singh and Narayan [89]
			Girsang et al. [81]
	VRPTW	2009	Häckel and Dippold [80]
Bee System (BS)	TSP	2002	Lučić and Teodorović [39]
		2003	Lučić and Teodorović [82]
		2006	Teodorovic et al. [40]
	SVRP	2003	Lučić and Teodorović [83]
	VRPTW	2013	Nikolić et al. [84]
		2015	Jawarneh and Abdullah [85]
Marriage in Honeybees Optimization (MHBO)	TSP	2007	Yang et al. [42]
			Yang et al. [43]
		2012	Celik and Ulker [87]
	VRP	2011	Marinakis et al. [86]
		2008	Marinakis et al. [44]
		2013	Ruan et al. [88]

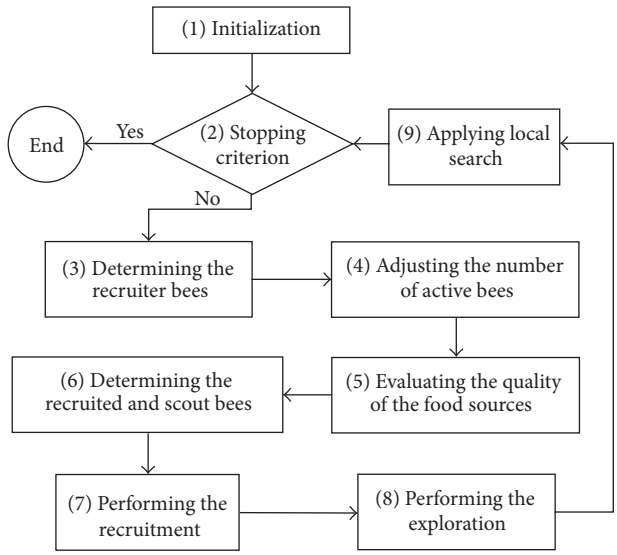


Figure 1: Flowchart of the TSPoptBees algorithm.

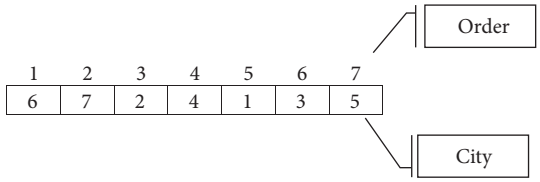


Figure 2: Solution representation for the TSP.

of the current solutions, multiple bees may be classified as recruiters. Some of the other bees are recruited by the recruiter bees to exploit their corresponding food sources, while the other ones become scout bees and explore new regions of the search space. If the number of promising regions is high, new bees can be activated to participate in the recruitment process. A high level flowchart of the TSPoptBees is presented in Figure 1 and each of the steps is then described.

5.1. Representation of a Candidate Solution. Each bee represents a candidate solution for the TSP, which is represented by the permutation of n integers. The example in Figure 2 represents a candidate solution that passes by the cities in the following order: 6, 7, 2, 4, 1, 3, 5, and returning to 6.

5.2. Initialization. In this step, the initial population of bees is built by using a set of construction heuristics [29], as follows:

(i) One solution is created using a Greedy algorithm.

(ii) One solution is created using the Cheapest Insertion algorithm.

(iii) Multiple solutions are created using the nearest neighbor algorithm. Each solution is created using a different initial city, meaning that it can create up to n different solutions.

(iv) If necessary, additional solutions are created by taking random routes created by the above algorithms after

applying one of the modification operators described in the Exploration Step.

5.3. Stopping Criterion. The stopping criterion terminates the search process when it completes a certain number of iterations without improving the best solution; that is, it is based on the stabilization of the search process.

5.4. Determining the Recruiter Bees. In this step there is a probabilistic selection of the recruiter bees based on the quality of the food source being explored by each bee. The main idea is that the better the quality of the food source, the greater the probability of the bee exploring this food source to become a recruiter.

The probability p_i of bee i, associated with a solution cost sc_i, becoming a recruiter is determined by

$$p_i = \left(\frac{(1 - p_{\min})}{(\mathrm{sc}_{\max} - \mathrm{sc}_{\min})} \right) * (\mathrm{sc}_{\max} - \mathrm{sc}_i) + p_{\min}. \quad (5)$$

In (5), $p_{\min}$ is the minimum probability of a bee to become a recruiter and $\mathrm{sc}_{\max}$ and $\mathrm{sc}_{\min}$ represent, respectively, the maximum and minimum solution cost among all bees. Then, the probability p_i is compared to a random number n_{random} within the interval [0, 1] in such a way that if $n_{\mathrm{random}} < p_i$, then bee i becomes a recruiter. To avoid multiple recruiter bees acting around the same promising region, an *inhibition radius* is used. Then, each recruiter is considered, in ascending order with regard to the solution cost sc_i, and the remaining recruiters that are within the inhibition radius ρ are labeled as nonrecruiters. In other words, given the distance $d_{i,j}$ between bees i and j, if $d_{i,j} < \rho$, then bee j is not a recruiter, assuming that bee i has a higher fitness than bee j.

The distance between two bees is calculated by a function based on their solution cost of the two bees being compared as in

$$d_{i,j} = \frac{\mathrm{abs}\left(\mathrm{sc}_i - \mathrm{sc}_j\right)}{\max\left(\left[\mathrm{sc}_i; \mathrm{sc}_j\right]\right)}. \quad (6)$$

In (6), sc is the cost of a specific solution, abs(·) is a function that returns the absolute value of the argument, and max(·) is a function that returns the highest value from a set of arguments.

5.5. Adjusting the Number of Active Bees. In this step, the number of active bees needed during the current iteration is determined, increasing or decreasing it based on the variety of food sources being explored. Let n_r be the number of recruiter bees, n_{act} the number of active bees in the current iteration, and n_m the expected number of nonrecruiters (recruited and scout) for each recruiter. Hence, $n_d = (n_r + 1) * n_m$ is the expected number of active bees in the current iteration. If $n_d > n_{\mathrm{act}}$, then $n_{ad} = n_d - n_{\mathrm{act}}$ is the number of bees to become active. If $n_d < n_{\mathrm{act}}$, then $n_{ad} = n_{\mathrm{act}} - n_d$ is the number of bees to be deactivated; $n_d \in [n_{\min}, n_{\max}]$. When necessary, new bees are included in the population (activated) by randomly selecting n_{ad} bees already in the population and applying one of the modification operators described in the Exploration Step. When bees are deactivated, the n_{ad} bees with the highest cost (worst solutions) are removed from the population.

5.6. Evaluating the Quality of the Food Sources. The algorithm aims to maximize the quality of the food source explored by each bee. In this step, a function that calculates the cost of all solutions and transforms them into a quality factor is used, as in (7), meaning that the lower the cost, the higher the quality.

$$q_i = 1 - \frac{(\mathrm{sc}_i - \mathrm{sc}_{\min})}{(\mathrm{sc}_{\max} - \mathrm{sc}_{\min})}. \quad (7)$$

In (7), q_i is the quality associated with the solution of bee i, sc_i is the cost of the solution associated with bee i, and $\mathrm{sc}_{\max}$ and $\mathrm{sc}_{\min}$ are, respectively, the maximum and minimum solution costs among all bees.

5.7. Determining the Recruited and Scout Bees. In this step the recruiter and scout bees are determined. Moreover, for each recruited bee, its recruiter is also defined such that the number of recruited bees for each recruiter is proportional to the quality of the food source associated with the recruiter. Consider $n_{\mathrm{nr}} = n_{\mathrm{act}} - n_r$ as the number of nonrecruiter bees. The number of recruited bees in the current iteration is $n_{\mathrm{rtd}} = \mathrm{round}(p_{\mathrm{rec}} * n_{\mathrm{nr}})$, in which round(·) is a function that gives the nearest integer and p_{rec} is the percentage of nonrecruiter bees that are going to be recruited. Thus, the number of scout bees is $n_{\mathrm{ex}} = n_{\mathrm{nr}} - n_{\mathrm{rtd}}$. Consider Q_r as the sum of the qualities of all recruiter bees; then $\mathrm{nr}_i = \mathrm{round}((q_i/Q_r) * n_r)$ is the number of recruited bees associated with recruiter bee i. Nonrecruiter bees are then associated with the recruiters using a roulette wheel approach based on the quality of the nonrecruiter bees, being tagged as recruited bees.

5.8. Performing the Recruitment. The recruitment consists of exploiting new food sources in the neighborhood of the recruiter bees by attracting the recruited ones. This is accomplished by combining the solution associated with the recruited bee with the solution associated with its recruiter, creating a new solution around this neighborhood. The combination of two solutions was implemented with a set of *crossover operators*, borrowing ideas from evolutionary algorithms. Three variations of the Order Crossover [94] are used:

(i) Type 1 (see Figure 3(b)): 50% to 80% of the recruited bee is copied as a single fragment to the final solution, while the remaining cities are copied, in order, from the recruiter.

(ii) Type 2 (see Figure 3(c)): 50% to 80% of the recruited bee is copied as a single fragment to the final solution, while the remaining cities are copied, based on the edges, from the recruiter.

(iii) Type 3 (see Figure 3(d)): 50% to 80% of the recruited bee is copied from two separated fragments to the

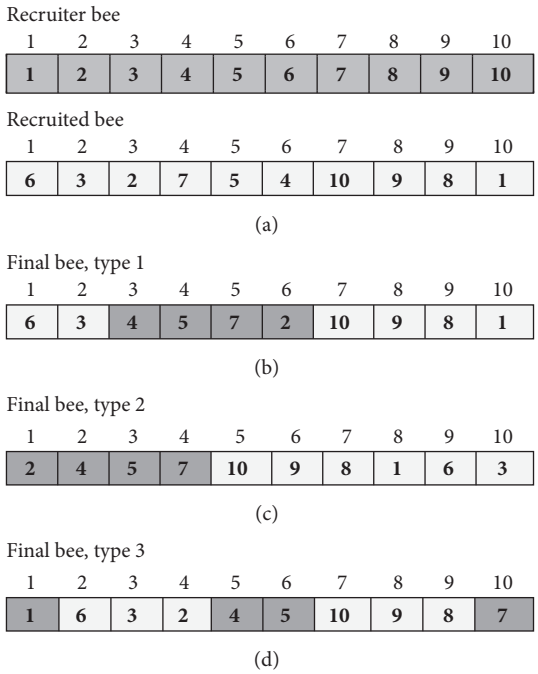


Figure 3: Example of a final solution after the possible crossover-like operators using the same recruiter and recruited bees.

final solution, while the remaining cities are copied, in order, from the recruiter.

For each pair of recruiter and recruited bees, the above operators are randomly chosen with the same probability.

After a new solution is created, it will be used by the recruited bee, replacing its current solution, only if there was an improvement. In addition, if the quality of the new solution associated with the recruited bee is better than the one associated with its recruiter, this solution is assigned to the local search, as will be explained later.

5.9. Performing the Exploration. In this step, the scout bees look for new food sources. Exploration operators that provide a new food source were implemented as a set of mutation operators, also borrowing ideas from evolutionary algorithms [95]. The operators are as follows:

(i) Swap of cities: randomly choose two cities and switch their positions.

(ii) City insertions: randomly choose a city and insert it in a new, random position.

(iii) Subroute inversion: randomly choose two cities and invert the subroute between them.

(iv) 3opt movement: perform a 3opt movement by randomly choosing the breaking and joining points.

For each scout bee, a random solution is chosen from the population and goes through one of the above operators, which is chosen randomly with the same probability.

5.10. Applying Local Search. In this step, the 2opt local search with the "first improvement" option is used to improve promising solutions [29]. This is limited to a maximum of $2 * n^2$ iterations at each run, in which n is the number of cities of the instance being solved.

Only a few selected recruited bees are assigned to the local search at each iteration, which occurs during the recruitment phase when the recruited bee finds a food source better than the one associated with its recruiter.

6. Performance Assessment

To assess the performance of TSPoptBees, several experiments were conducted using benchmark instances from the TSPLIB (available online at http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/) [7]. In total, 28 instances, ranging from 48 to 439 cities, were used and the optimal solution is known for each of them. The proposed algorithm was implemented in MATLAB and executed on an Intel Core i5 1.9 GHz with 8 GB RAM. For each instance, the algorithm was executed 30 times.

Before running the TSPoptBees, the value for a set of parameters must be defined. Further details on how these parameters can be tuned were presented in Masutti and de Castro [96] and the values used in the present experiments are presented below:

(i) $n_{\min} = 50$: minimum number of active bees, which is also the size of the initial population.

(ii) $n_{\max} = 300$: maximum number of active bees.

(iii) $n_m = 18$: average number of nonrecruiter bees (recruited and scout) for each recruiter.

(iv) $p_{\min} = 1\%$: minimum probability of an active bee to become a recruiter.

(v) $p_{\text{rec}} = 70\%$: percentage of recruited bees among the nonrecruiters (recruited and scout).

(vi) $\rho = 0.1$: social inhibition radius.

(vii) maxIt = 1000: maximum number of iterations without an improvement of the best solution.

6.1. Assessing the Computational Cost Based on the Number of Customers. The computational cost analysis of TSPoptBees was done empirically when executing the algorithm for two instances of TSPLIB: fl417 and pr439 were divided into another 20 instances, 10 for each original TSPLIB instance, varying in size according to sets [25 50 75 100 150 200 250 300 400 417] and [25 50 75 100 150 200 250 300 400 439], respectively. Using city sets from the same instance for this test presented more reliable results than using different instances with different sizes, since each instance may have particular characteristics that influence the convergence of the search process. Two different instances, but of similar sizes, were used to reduce the bias due to the characteristics of a single instance.

For each instance described above, TSPoptBees was run 30 times and only the characteristics related to the computational effort of the algorithm were used in the

Table 3: TSPoptBees computational cost evaluation for TSP. n is the number of cities in the instance; Iterations is the average number of iterations for convergence; n_{act} is the average number of bees in the final swarm; Time is the average time, in seconds, for running the algorithm.

n	fl417			pr439		
	Iterations	n_{act}	Time	Iterations	n_{act}	Time
25	1402.63	170.13	14.98	1402.63	207.53	19.10
50	1657.60	203.87	28.50	1560.57	242.73	34.06
75	2034.80	253.73	55.55	2170.07	259.60	63.78
100	2402.63	277.93	95.29	2234.60	264.00	83.72
150	3110.63	311.67	195.43	3692.37	287.47	200.83
200	4182.17	369.60	381.15	4065.37	282.33	293.47
250	4927.83	426.07	617.87	5299.03	300.67	474.90
300	5636.97	473.73	1006.08	4742.80	309.47	576.66
400	6085.43	486.20	1726.61	4821.03	321.20	961.55
417	5746.30	495.00	1638.32	—	—	—
435	—	—	—	7591.30	289.67	1719.78

analysis of the results. These characteristics are (1) the average number of iterations for convergence, (2) the mean size of the final swarm, and (3) the average running time. It is worth mentioning that the obtained results are specific to the implementation used, since each operator can present a different computational complexity.

Table 3 presents the results obtained for the two instances used. For each instance, the average results of the number of iterations for convergence of the search process, the number of active bees in the final swarm, and the execution time of the algorithm are presented. According to the results presented, the number of active bees in the final swarm tends to increase along with the growth of the number of cities in the instance. However, growth seems to stabilize from a certain value of n. This behavior is observed for the two instances used in this experiment. For the fl417 instance, the growth is more accentuated for smaller values of n, but it seems to stabilize around $n = 300$. For the instance pr439, growth exists, however, in a milder fashion. An interesting point is that the value of n_{act} decreased for instance pr439 between $n = 400$ and $n = 439$ (actual instance size).

For the number of iterations for convergence (Iterations in Table 3), the behavior is also similar for the two instances. The number of iterations for $25 \leq n \leq 250$ is very close between the two instances, presenting a marked increase. From this value, there is a different behavior between the two instances. For fl417, there is a smaller growth between $n = 250$ and $n = 400$ and a slight decrease for $n = 417$ (actual instance size). For pr439, there is a decrease in the number of iterations between $n = 250$ and $n = 300$, almost constant between 300 and 400, and an abrupt increase between 400 and 439.

The runtime also has similar characteristics between the two instances and can be seen as a composition of the two attributes described above. For the two instances, the values are very similar between $n = 25$ and $n = 150$, presenting a certain growth in the execution time. From this point, there is a more pronounced growth for the fl417 instance, which follows a behavior similar to that seen for the number of bees. However, the runtime decreases between $n = 400$ and $n = 417$, the same behavior seen for the number of iterations. For the pr439 instance, there is a less pronounced growth, which changes noticeably between $n = 400$ and $n = 439$, with an abrupt increase, as well as the observed number of iterations.

It is evidenced by the results that, in relation to the computational effort, the algorithm can present different values for different instances. However, what can be observed is a behavior trend that is independent of the instance being solved. Figure 4 shows the runtime-related trend for the two instances. It is observed that the running time can be approximated by a second-degree polynomial. Given the factorial nature of the problem, with the number of possible solutions increasing in factorial scale in relation to the number of cities, this tendency of quadratic growth is good.

6.2. Assessing the Quality of the Solutions. For each of the 28 instances, the algorithm was run 30 times. The results obtained are presented in Table 4. For each instance, the result is presented with the following attributes: (1) the cost of the best solution found (Best); (2) the number of times the best-known solution (BKS) was found; (3) the average cost of the best solution found; (4) the percentage deviation of the best solution found for BKS (PDb); (5) the percentage deviation of the mean solution for BKS (PDav); (6) the average number of iterations for the convergence; (7) the average number of bees in the final swarm; and (8) the average time (Time), in seconds, of the algorithm execution.

Evaluating the best solution obtained for each of the instances in the 30 runs (Best and PDb in Table 4), TSPoptBees was able to find the best-known solution at least once for 19 of the 28 instances of TSP. Out of these 19 instances, the TSP was found more than once for 17 of them. Among the 9 instances for which the BKS was not found, the highest value for PDb was 0.44% for the pr439 instance. On average, the best solution found by TSPoptBees deviates by 0.08% from the BKS (average value for PDb).

Regarding the average solution obtained during the 30 runs for each instance (PDav and PDb in Table 4), TSPoptBees was not able to find the BKS in all runs for any instance.

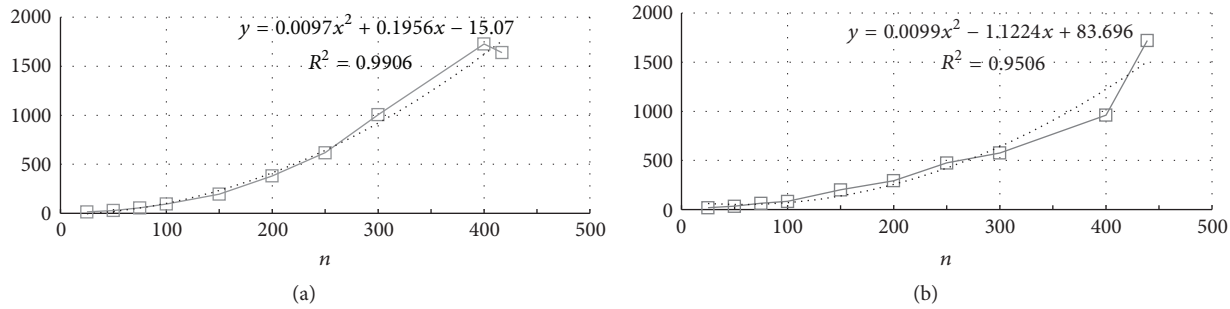


Figure 4: Running time of TSPoptBees for two TSP instances: (a) fl417 and (b) pr439.

Table 4: Computational results of TSPoptBees.

Instance	BKS	Best	# BKS	Average	PDb	PDav	Iterations	# bees	Time
att48	**10628**	10628	15	10646.20	0.00	0.17	2159.50	180.40	49.45
eil51	**426**	427	0	427.37	0.23	0.32	1954.63	137.87	40.13
berlin52	**7542**	7542	29	7545.83	0.00	0.05	1486.20	161.33	33.56
st70	**675**	675	8	679.73	0.00	0.70	2270.53	186.27	78.49
eil76	**538**	538	12	541.20	0.00	0.59	3322.10	168.67	98.24
pr76	**108159**	108159	23	108368.90	0.00	0.19	2426.93	192.13	81.91
kroA100	**21282**	21282	16	21296.63	0.00	0.07	2061.30	219.27	86.89
kroB100	**22141**	22141	17	22176.93	0.00	0.16	2575.17	225.13	123.98
kroC100	**20749**	20749	16	20797.33	0.00	0.23	2706.03	223.67	132.27
kroD100	**21294**	21294	3	21456.37	0.00	0.76	2992.10	238.33	139.68
kroE100	**22068**	22106	0	22144.40	0.17	0.35	2627.53	226.60	122.91
rd100	**7910**	7910	3	7997.07	0.00	1.10	2557.03	214.13	132.98
eil101	**629**	629	8	630.93	0.00	0.31	2606.13	179.67	91.90
lin105	**14379**	14379	29	14380.23	0.00	0.01	2352.03	250.80	129.81
pr107	**44303**	44303	18	44345.60	0.00	0.10	1853.17	319.73	183.47
pr124	**59030**	59030	20	59107.07	0.00	0.13	2296.20	309.47	181.00
bier127	**118282**	118282	4	118650.20	0.00	0.31	2901.80	167.20	120.26
pr136	**96772**	96785	0	97671.70	0.01	0.93	4030.93	225.13	275.90
kroA150	**26524**	26583	0	26827.27	0.22	1.14	3586.33	243.47	291.64
kroB150	**26130**	26130	3	26237.90	0.00	0.41	4075.33	242.00	371.65
rat195	**2323**	2331	0	2352.53	0.34	1.27	3724.30	247.87	538.12
kroA200	**29368**	29368	1	29527.93	0.00	0.54	4063.77	282.33	564.80
kroB200	**29437**	29489	0	29771.77	0.18	1.14	4489.53	282.33	464.05
tsp225	**3916**	3916	1	3984.37	0.00	1.75	4743.40	254.47	711.53
a280	**2579**	2579	3	2613.50	0.00	1.34	4424.30	268.40	939.57
lin318	**42029**	42196	0	42589.33	0.40	1.33	6341.37	267.67	1428.00
fl417	**11861**	11902	0	12003.13	0.35	1.20	5907.13	492.80	3026.35
pr439	**107217**	107685	0	110352.80	0.44	2.92	6524.4	299.20	2651.69

The largest deviation of the mean solution from the BKS was 2.92% for the pr439 instance. The average solution obtained by TSPoptBees for the set of instances used in this test had a deviation of 0.70% for the BKS (mean value for PDb).

Table 5 presents a comparison of the quality of the solutions obtained by TSPoptBees with those published in works that relate the TSP solution with algorithms inspired by bee behavior. The algorithms listed in this table are (1) BS + 2opt [82, 83], (2) BCO [36], (3) ABC-GSX [34], (4) ABC & PR [47], (5) CABC [45], (6) BCOPR [81], and (7) ABC + 2opt [60]. The best solutions in the comparison are shown in italic in the table and a dash (—) appears when the result is not published for that specific instance.

The HBMO-TSP uses in internal steps two techniques that already present, by themselves, competitive results for the TSP: (1) the heuristic called MPNS-GRASP (Marinakis et al., 2009) for the definition of the initial solutions and (2) the heuristic called ENS [97] for the local search stage. The BCO does not present any additional peculiarity to the structure of the algorithm itself that would favor the presented results.

Table 5: Comparison of the computational results of TSPoptBees with other bee-inspired algorithms from the literature.

Instance	TSPoptBees		BS + 2opt		BCO		ABC-GSX		ABC & PR		CABC		BCOPR		ABC + 2opt	
	PDb	PDav	PDb	PDav	PDb	PDav	PDb	PDav	PDb	PDav	PDb	PDav	PDb	PDav	PDb	PDav
att48	*0.00*	*0.17*	—	—	0.31	0.83	—	—	—	—	—	—	—	—	—	—
eil51	0.23	0.32	0.53	1.14	0.47	0.85	*0.00*	0.94	—	*0.00*	—	—	*0.00*	1.1	*0.00*	0.39
berlin52	*0.00*	0.05	*0.00*	1.19	—	—	*0.00*	0.03	—	—	—	—	—	—	*0.00*	*0.00*
st70	*0.00*	0.70	0.22	1.06	—	—	*0.00*	0.71	—	*0.00*	—	—	0.74	2.46	*0.00*	0.56
eil76	*0.00*	0.59	—	—	0.19	2.01	*0.00*	2.41	—	*0.00*	—	—	—	—	—	—
pr76	*0.00*	*0.19*	0.58	1.19	—	—	*0.00*	0.46	—	—	—	—	—	—	*0.00*	0.45
kroA100	*0.00*	*0.07*	0.73	1.36	2.26	3.43	*0.00*	0.20	—	—	—	—	—	—	*0.00*	1.04
kroB100	*0.00*	*0.16*	—	—	2.24	3.10	—	—	—	—	—	—	—	—	—	—
kroC100	*0.00*	*0.23*	—	—	0.50	1.50	—	—	—	—	—	—	—	—	—	—
kroD100	*0.00*	*0.76*	—	—	1.64	3.25	—	—	—	—	—	—	—	—	—	—
kroE100	*0.17*	*0.35*	—	—	1.73	2.20	—	—	—	—	—	—	—	—	—	—
rd100	*0.00*	*1.10*	—	—	—	—	—	—	—	1.12	—	—	—	—	—	—
eil101	*0.00*	*0.31*	0.35	3.97	0.95	2.29	—	—	—	1.56	—	—	1.27	3.31	1.50	2.90
lin105	*0.00*	*0.01*	—	—	0.32	1.24	—	—	—	1.24	—	—	—	—	—	—
pr107	*0.00*	*0.10*	—	—	—	—	—	—	—	0.94	—	—	—	—	—	—
pr124	*0.00*	*0.13*	—	—	—	—	—	—	—	0.36	—	—	—	—	—	—
bier127	*0.00*	*0.31*	—	—	—	—	—	—	—	1.45	—	—	—	—	—	—
pr136	*0.01*	*0.93*	—	—	—	—	—	—	—	1.56	—	—	—	—	—	—
kroA150	*0.22*	*1.14*	—	—	5.03	6.39	—	—	—	—	—	—	—	—	—	—
kroB150	*0.00*	*0.41*	—	—	1.55	3.68	—	—	—	—	0.45	0.98	—	—	—	—
rat195	*0.34*	*1.27*	—	—	—	—	—	—	—	2.30	—	—	—	—	—	—
kroA200	*0.00*	*0.54*	—	—	2.02	4.26	—	—	—	—	0.18	0.62	—	—	—	—
kroB200	*0.18*	*1.14*	—	—	3.10	6.36	—	—	—	—	—	—	—	—	—	—
tsp225	*0.00*	*1.75*	5.35	6.60	—	—	—	—	—	—	—	—	—	—	8.51	12.41
a280	*0.00*	*1.34*	5.95	7.66	—	—	—	—	—	—	—	—	3.95	5.30	21.83	23.92
lin318	*0.40*	*1.33*	—	—	6.32	7.55	—	—	—	—	—	—	—	—	—	—

7. Conclusions and Future Trends

Vehicle routing problems are of great importance in scientific research, since they present a wide set of practical applications and are difficult to solve in the optimality by exact methods, being one of the most studied classes of combinatorial optimization problems. Finding optimal solutions to these problems is unfeasible for large instances due to the factorial growth in the number of possible solutions. Thus, optimal solution methods are replaced by heuristics capable of finding good solutions in reasonable time. Among the many heuristics that have been applied to such problems, swarm-based algorithms, including bee-inspired methods, have received a great deal of attention over the past decade.

Due to the relevance of vehicle routing problems and swarm intelligence algorithms, the first contribution of this paper was to provide a thorough review of bee-inspired methods designed to solve this class of discrete optimization tasks. A taxonomy of methods was presented and the review followed in chronological order based on the taxonomy used and taking into account the problem solved and modifications introduced in the algorithm. It then followed with a presentation of the TSPoptBees algorithm for the solution of the most well-known vehicle routing problem: the travelling salesman problem. The proposed approach presented competitive results, obtaining, in average, solutions of better quality when compared to other algorithms inspired by the behavior of bees.

The survey presented showed that the Artificial Bee Colony (ABC) algorithm is by far the most explored proposal in the literature, but for obtaining good results in vehicle routing problems it invariably requires modifications, hybridizations, and improvements in its standard structure. The same holds true for the BCO, the Bee System (BS), and the MHBO algorithms. In the particular case of TSPoptBees, its former algorithm (optBees) was introduced to solve continuous optimization tasks and, thus, it had to be adapted to solve discrete optimization problems.

For virtually all bee-inspired algorithms, a number of issues open avenues for future research in the context of combinatorial optimization problems and others, such as methods to automatically define the input parameters, including self-adaptive approaches; the use of initialization, selection, and local search techniques specially designed for combinatorial optimization problems; the investigation and improvements of convergence times and properties; empirical studies about the dynamics of the swarm; the design of a general-purpose framework for bee-inspired

algorithms; the inclusion of diversity control mechanisms; a better understanding of the classes of problems these algorithms perform better and why; and the parallelization of the methods. Last but not least, bee-inspired approaches have a great potential to be adapted and applied to various other contexts, from pattern recognition to autonomous navigation.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank FAPESP, CNPq, Capes, and MackPesquisa for the financial support. As a Machine Learning Center of Excellence, special thanks also go to Intel for all its support.

References

[1] S. Luke, *Essentials of Metaheuristics*, Lulu, 2nd edition, 2013.

[2] Z. Michalewicz and D. Fogel, *How to Solve It: Modern Heuristics*, Springer Science & Business Media, 2013.

[3] A. G. Cunha, R. Takahashi, and C. H. Antunes, *Evolutionary Computing and Metaheuristics Manual*, Imprensa da Universidade de Coimbra, Coimbra, Portugal, 2012.

[4] C. B. da Cunha, "Practical aspects of the application of vehicle routing models to real world problems," *Transportes*, vol. 8, no. 2, pp. 1–23, 2000.

[5] E. Lawler, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, John Wiley & Sons, Incorporated, 1985.

[6] G. Laporte, "The traveling salesman problem: an overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, no. 2, pp. 231–247, 1992.

[7] G. Reinelt, "TSPLIB: A traveling salesman problem library," *ORSA Journal on Computing*, vol. 3, no. 4, pp. 376–384, 1991.

[8] P. Toth and D. Vigo, *The Vehicle Routing Problem*, Society for Industrial and Applied Mathematics (SIAM), 2001.

[9] G. Gutin and A. Punnen, *The Traveling Salesman Problem And Its Variations*, Springer Science & Business Media, 2002.

[10] G. Laporte, "Fifty years of vehicle routing," *Transportation Science*, vol. 43, no. 4, pp. 408–416, 2009.

[11] G. Laporte, M. Gendreau, J. Potvin, and F. Semet, "Classical and modern heuristics for the vehicle routing problem," *International Transactions in Operational Research*, vol. 7, no. 4-5, pp. 285–300, 2000.

[12] P. Toth and D. Vigo, *Vehicle Routing*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2014.

[13] L. N. de Castro, *Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications and Applications*, CRC Press, 2006.

[14] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, 2010.

[15] H. Zang, S. Zhang, and K. Hapeshi, "A review of nature-inspired algorithms," *Journal of Bionic Engineering*, vol. 7, pp. S232–S237, 2010.

[16] L. N. de Castro, "Fundamentals of natural computing: an overview," *Physics of Life Reviews*, vol. 4, no. 1, pp. 1–36, 2007.

[17] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems. [S.l.]*, Oxford University Press, 1999.

[18] C. Blum and D. Merkle, *Swarm Intelligence: Introduction and Applications, Natural Computing Series*, Springer, 2008.

[19] M. Dorigo, M. Birattari, and T. Stützle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[20] R. Eberhart and Y. Shi, "Particle Swarm Optimization: Developments, Applications and Resources," in *Proceedings of the Congress on Evolutionary Computation, IEEE 2001*, pp. 81–86, 2001.

[21] R. D. Maia, L. N. de Castro, and W. M. Caminhas, "Collective decision-making by bee colonies as model for optimization - the OptBees algorithm," *Applied Mathematical Sciences*, vol. 7, no. 85-88, pp. 4327–4351, 2013.

[22] D. Karaboga and B. Akay, "A survey: algorithms simulating bee swarm intelligence," *Artificial Intelligence Review*, vol. 31, no. 1–4, pp. 61–85, 2009.

[23] S. Bitam, M. Batouche, and E.-G. Talbi, "A survey on bee colony algorithms," in *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing, Workshops and Phd Forum (IPDPSW '10)*, pp. 1–8, IEEE, New York, NY, USA, April 2010.

[24] J. A. Ruiz-Vanoye, O. Díaz-Parra, F. Cocón et al., "Meta-heuristics algorithms based on the grouping of animals by social behavior for the traveling salesman problem," *International Journal of Combinatorial Optimization Problems and Informatics*, vol. 3, no. 3, pp. 104–123, 2012.

[25] B. K. Verma and D. Kumar, "A review on artificial bee colony algorithm," *International Journal of Engineering & Technology*, vol. 2, no. 3, pp. 175–186, 2013.

[26] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, pp. 21–57, 2014.

[27] D. Agarwal, A. Gupta, and P. K. Singh, "A systematic review on artificial bee colony optimization technique," *International Journal of Control Theory and Applications*, vol. 9, no. 11, pp. 5487–5500, 2016.

[28] M. M. Millonas, "Swarms, Phase Transitions, and Collective Intelligence," in *LANGTON, C. G. Artificial Life III: Proceedings of the Workshop on Artificial Life*, pp. 417–445, LANGTON, C. G. Artificial Life III: Proceedings of the Workshop on Artificial Life, 1994.

[29] D. S. Johnson and A. McGeoch, "The traveling salesman problem: A case study in local optimization," *Local search in Combinatorial Optimization*, vol. 1, pp. 215–310, 1997.

[30] R. D. Angel, W. L. Caudle, R. Noonan, and A. Whinston, "Computer-assisted school bus scheduling," *Management Science*, vol. 18, no. 6, pp. 279–288, 1972.

[31] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209–219, 2006.

[32] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005, vol. 200.

[33] B. Basturk and D. Karaboga, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing Journal*, vol. 8, no. 1, pp. 687–697, 2008.

[34] A. Banharnsakun, T. Achalakul, and B. Sirinaovakul, "ABC-GSX: A hybrid method for solving the traveling salesman problem," in *Proceedings of the 2010 2nd World Congress on Nature and Biologically Inspired Computing, NaBIC 2010*, pp. 7–12, jpn, December 2010.

[35] S. Nakrani and C. Tovey, "On honey bees and dynamic server allocation in internet hosting centers," *Adaptive Behavior*, vol. 12, no. 3-4, pp. 223–240, 2004.

[36] L. P. Wong, M. Y. H. Low, and C. S. Chong, "A bee colony optimization algorithm for traveling salesman problem," in *Proceedings of the Second Asia International Conference on Modelling & Simulation*, pp. 818–823, 2008.

[37] X. Lu and Y. Zhou, "A novel global convergence algorithm: bee collecting pollen algorithm," in *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence. ICIC 2008. Lecture Notes in Computer Science*, D. S. Huang, D. C. Wunsch, D. S. Levine, and Jo. K. H., Eds., vol. 5227, Springer, Berlin, Heidelberg, 2008.

[38] P. Lučić and D. Teodorović, "Bee System: Modeling Combinatorial Optimization Transportation Engineering Problems by Swarm Intelligence," in *Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis*, pp. 441–445, 2001.

[39] P. Lučić and D. Teodorović, "Transportation modeling: An artificial life approach," *Proceedings of the International Conference on Tools with Artificial Intelligence*, pp. 216–223, 2002.

[40] D. Teodorovic, P. Lucic, G. Markovic, and M. Dell'Orco, "Bee colony optimization: principles and applications," in *In Neural Network Applications in Electrical Engineering*, pp. 151–156, NEUREL, 2006.

[41] H. Abbass, "MHBO: Marriage in honey bees optimization-A haplometrosis polygynous swarming approach," in *Proceedings of the, 2001 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 207–214, 2001.

[42] C. Yang, J. Chen, and X. Tu, "Algorithm of marriage in honey bees optimization based on the nelder-mead method," in *Proceedings of the International Conference on Intelligent Systems and Knowledge Engineering (ISKE 2007). Advances in Intelligent Systems Research*, 2007.

[43] C. Yang, X. Tu, and J. Chen, "Algorithm of marriage in honey bees optimization based on the wolf pack search," in *Proceedings of the 2007 International Conference on Intelligent Pervasive Computing, IPC 2007*, pp. 462–467, kor, October 2007.

[44] Y. Marinakis, M. Marinaki, and G. Dounias, "Honey bees mating optimization algorithm for the vehicle routing problem," *Studies in Computational Intelligence*, vol. 129, pp. 139–148, 2008.

[45] D. Karaboga and B. Gorkemli, "A combinatorial Artificial Bee Colony algorithm for traveling salesman problem," in *Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications (INISTA '11)*, pp. 50–53, Istanbul, Turky, June 2011.

[46] V. Singh, R. Tiwari, D. Singh, and A. Shukla, "RGBCA-genetic bee colony algorithm for travelling salesman problem," in *Proceedings of the 2011 World Congress on Information and Communication Technologies, WICT 2011*, pp. 1002–1008, ind, December 2011.

[47] X. Zhang, Q. Bai, and X. Yun, "A new hybrid artificial bee colony algorithm for the traveling salesman problem," in *Proceedings of the 2011 IEEE 3rd International Conference on Communication Software and Networks, ICCSN 2011*, pp. 155–159, chn, May 2011.

[48] W. Y. Szeto, Y. Wu, and S. C. Ho, "An artificial bee colony algorithm for the capacitated vehicle routing problem," *European Journal of Operational Research*, vol. 215, no. 1, pp. 126–135, 2011.

[49] E. Özceylan, M. S. Kiran, and Y. Atasagun, "A new hybrid heuristic approach for solving green traveling salesman problem," in *Proceedings of the 41st International Conference on Computers and Industrial Engineering*, pp. 23–26, 2011.

[50] P. Ji and Y. Wu, "An improved artificial bee colony algorithm for the capacitated vehicle routing problem with time-dependent travel times," in *Proceedings of the Tenth International Symposium On Operations Research And Its Applications (ISORA 2011*, pp. 75–82, 2011.

[51] W. Li, W. Li, Y. Yang, H. Liao, J. Li, and X. Zheng, "Artificial bee colony algorithm for traveling salesman problem," *Advanced Materials Research*, vol. 314-316, pp. 2191–2196, 2011.

[52] I. Brajevic, "Artificial bee colony algorithm for the capacitated vehicle routing problem," in *Proceedings of the European Computing Conference, ECC '11*, pp. 239–244, 2011.

[53] Y.-J. Shi, F.-W. Meng, and G.-J. Shen, "A modified artificial bee colony algorithm for vehicle routing problems with time windows," *Information Technology Journal*, vol. 11, no. 10, pp. 1490–1495, 2012.

[54] S. Iqbal and M. S. Rahman, "Vehicle routing problems with soft time windows," in *Proceedings of the 2012 7th International Conference on Electrical and Computer Engineering, ICECE 2012*, pp. 634–638, 2012.

[55] K. Karabulut and M. F. Tasgetiren, "A discrete artificial bee colony algorithm for the traveling salesman problem with time windows," in *Proceedings of the 2012 IEEE Congress on Evolutionary Computation, CEC 2012*, aus, June 2012.

[56] A. S. Bhagade and P. V. Puranik, "Artificial Bee Colony (ABC) Algorithm For Vehicle Routing Optimization Problem," *International Journal of Soft Computing and Engineering*, vol. 2, pp. 329–333, 2012.

[57] N. Pathak and S. P. Tiwari, "Traveling Salesman Problem Using Bee Colony with SPV," *International Journal of Soft Computing and Engineering*, vol. 2, no. 3, pp. 410–414, 2012.

[58] L. Li, Y. Cheng, L. Tan, and B. Niu, "A discrete artificial bee colony algorithm for TSP problem," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6840, pp. 566–573, 2012.

[59] W. Bin, C. Hong, and Z.-Y. Cui, "Artificial bee colony algorithm for two-dimensional loading capacitated vehicle routing problem," in *Proceedings of the 2013 20th International Conference on Management Science and Engineering, ICMSE 2013*, pp. 406–412, chn, July 2013.

[60] M. S. Kiran, H. Işcan, and M. Gündüz, "The analysis of discrete artificial bee colony algorithm with neighborhood operator on traveling salesman problem," *Neural Computing and Applications*, vol. 23, no. 1, pp. 9–21, 2013.

[61] S. Sabet, F. Farokhi, and M. Shokouhifar, "A hybrid mutation-based artificial bee colony for traveling salesman problem," *Lecture Notes on Information Theory*, vol. 1, no. 3, pp. 99–103, 2013.

[62] B. Yao, P. Hu, M. Zhang, and S. Wang, "Artificial bee colony algorithm with scanning strategy for the periodic vehicle routing problem," *Simulation*, vol. 89, no. 6, pp. 762–770, 2013.

[63] C.-Y. Liu, "An improved adaptive genetic algorithm for the multi-depot vehicle routing problem with time window," *Journal of Networks*, vol. 8, no. 5, pp. 1035–1042, 2013.

[64] W. Yang and Z. Pei, "Hybrid ABC/PSO to solve travelling salesman problem," *International Journal of Computing Science and Mathematics*, vol. 4, no. 3, pp. 214–221, 2013.

[65] S. Pandey and S. Kumar, "Enhanced artificial bee colony algorithm and its application to travelling salesman problem," *HCTL Open International Journal of Technology Innovations and Research*, vol. 2, pp. 137–146, 2013.

[66] A. Rekaby, A. A. Youssif, and A. Sharaf Eldin, "Introducing adaptive artificial bee colony algorithm and using it in solving traveling salesman problem," in *Proceedings of the Science and Information Conference (SAI '13)*, pp. 502–506, 2013.

[67] Y. Yuan and Y. Zhu, "A hybrid artificial bee colony optimization algorithm," in *Proceedings of the 2014 10th International Conference on Natural Computation, ICNC 2014*, pp. 492–496, chn, August 2014.

[68] S. Zhang, C. K. M. Lee, K. L. Choy, W. Ho, and W. H. Ip, "Design and development of a hybrid artificial bee colony algorithm for the environmental vehicle routing problem," *Transportation Research Part D: Transport and Environment*, vol. 31, pp. 85–99, 2014.

[69] O. E. Nahum, Y. Hadas, and U. Spiegel, "Multi-objective vehicle routing problems with time windows: A vector evaluated artificial bee colony approach," *International Journal of Computer and Information Technology*, vol. 3, pp. 41–47, 2014.

[70] H. E. Kocer and M. R. Akca, "An improved artificial bee colony algorithm with local search for traveling salesman problem," *Cybernetics and Systems*, vol. 45, no. 8, pp. 635–649, 2014.

[71] Y.-T. Chung, "A Hybrid Artificial Bee Colony Algorithm to Solve the Capacitated Vehicle Routing Problem in Logistics Management, Master's Dissertation," in *Department of Industrial Management*, p. 81, National Taiwan University of Science and Technology, 2014.

[72] R. Nagaya and A. Inoie, "Hybrid ABC algorithm for the capacitated vehicle routing problem," in *Proceedings of the 8th International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, 383, 382 pages, 2014.

[73] S. Z. Zhang and C. K. M. Lee, "An Improved Artificial Bee Colony Algorithm for the Capacitated Vehicle Routing Problem," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015*, pp. 2124–2128, hkg, October 2015.

[74] M. Gündüz, M. S. Kiran, and E. Özceylan, "A hierarchic approach based on swarm intelligence to solve the traveling salesman problem," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 23, no. 1, pp. 103–117, 2015.

[75] M. Alzaqebah, S. Abdullah, and S. Jawarneh, "Modified artificial bee colony for the vehicle routing problems with time windows," *SpringerPlus*, vol. 5, no. 1, article no. 1298, 2016.

[76] L. P. Wong, M. Y. Hean Low, and C. S. Chong, "An efficient bee colony optimization algorithm for traveling salesman problem using frequency-based pruning," in *Proceedings of the 7th IEEE International Conference on Industrial Informatics (INDIN '09)*, pp. 775–782, IEEE, Wales, UK, June 2009.

[77] L. P. Wong, M. Y. H. Low, and C. S. Chong, "A bee colony optimization algorithm with the fragmentation state transition rule for traveling salesman problem," in *Proceedings of the 2009 Conference on Innovative Production Machines and Systems (IPROMS 2009)*, pp. 399–404, 2009.

[78] L.-P. Wong, M. Y. Hean Low, and C. S. Chong, "A generic bee colony optimization framework for combinatorial optimization problems," in *Proceedings of the Asia Modelling Symposium 2010: 4th International Conference on Mathematical Modelling and Computer Simulation, AMS2010*, pp. 144–151, 2010.

[79] L.-P. Wong, M. Y. H. Low, and C. S. Chong, "Bee colony optimization with local search for traveling salesman problem," *International Journal on Artificial Intelligence Tools*, vol. 19, no. 3, pp. 305–334, 2010.

[80] S. Häckel and P. Dippold, "The Bee Colony-inspired Algorithm (BCiA): A two-stage approach for solving the vehicle routing problem with time windows," in *Proceedings of the 11th Annual Genetic and Evolutionary Computation Conference, GECCO-2009*, pp. 25–32, can, July 2009.

[81] A. S. Girsang, C.-W. Tsai, and C.-S. Yang, "A fast bee colony optimization for traveling salesman problem," in *Proceedings of the 3rd International Conference on Innovations in Bio-Inspired Computing and Applications, IBICA 2012*, pp. 7–12, 2012.

[82] P. Lučić and D. Teodorović, "Computing with bees: Attacking complex transportation engineering problems," *International Journal on Artificial Intelligence Tools*, vol. 12, no. 03, pp. 375–394, 2003.

[83] P. Lučić and D. Teodorović, "Vehicle routing problem with uncertain demand at nodes: the bee system and fuzzy logic approach," in *Fuzzy Sets Based Heuristics for Optimization*, vol. 126 of *Studies in Fuzziness and Soft Computing*, pp. 67–82, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[84] M. Nikolić, D. Teodorović, and M. Šelmić, "Solving the vehicle routing problem with time windows by bee colony optimization metaheuristic," in *In Proceedings of the 1st Logistics International Conference*, pp. 44–48, 2013.

[85] S. Jawarneh and S. Abdullah, "Sequential insertion heuristic with adaptive bee colony optimisation algorithm for vehicle routing problem with time windows," *PLoS ONE*, vol. 10, no. 7, 2015.

[86] Y. Marinakis, M. Marinaki, and G. Dounias, "Honey bees mating optimization algorithm for the Euclidean traveling salesman problem," *Information Sciences*, vol. 181, no. 20, pp. 4684–4698, 2011.

[87] Y. Celik and E. Ulker, "A marriage in honey bee optimisation approach to the asymmetric travelling salesman problem," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 6, pp. 4123–4132, 2012.

[88] Q. Ruan, Z. Zhang, L. Miao, and H. Shen, "A hybrid approach for the vehicle routing problem with three-dimensional loading constraints," *Computers & Operations Research*, vol. 40, no. 6, pp. 1579–1589, 2013.

[89] A. Singh and D. Narayan, "Augmentation of travelling salesman problem using bee colony optimization," *International Journal of Innovative Technology and Exploring Engineering*, 2012.

[90] R. D. Maia, L. N. De Castro, and W. M. Caminhas, "Bee colonies as model for multimodal continuous optimization: The OptBees algorithm," in *Proceedings of the 2012 IEEE Congress on Evolutionary Computation, CEC 2012*, aus, June 2012.

[91] D. P. F. Cruz, R. D. Maia, A. Szabo, and L. N. de Castro, "A bee-inspired algorithm for optimal data clustering," in *Proceedings of the 2013 IEEE Congress on Evolutionary Computation, CEC 2013*, pp. 3140–3147, mex, June 2013.

[92] D. P. F. Cruz, R. D. Maia, L. A. da Silva, and L. N. de Castro, "BeeRBF: A bee-inspired data clustering approach to design RBF neural network classifiers," *Neurocomputing*, vol. 172, Article ID 15835, pp. 427–437, 2016.

[93] T. A. S. Masutti and L. N. De Castro, "TSPoptBees: A bee-inspired algorithm to solve the traveling salesman problem," in *Proceedings of the 5th IIAI International Congress on Advanced Applied Informatics, IIAI-AAI 2016*, pp. 593–598, jpn, July 2016.

[94] L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.

[95] P. Larrañaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, "Genetic algorithms for the travelling salesman problem: A review of representations and operators," *Artificial Intelligence Review*, vol. 13, no. 2, pp. 129–170, 1999.

[96] T. A. S. Masutti and L. N. de Castro, "Parameter analysis of a bee-inspired algorithm to solve the traveling salesman problem," in *Proceedings of the 15th IASTED International Conference on Intelligent Systems and Control, ISC 2016*, pp. 245–252, 2016.

[97] Y. Marinakis, A. Migdalas, and P. M. Pardalos, "Expanding neighborhood GRASP for the traveling salesman problem," *Computational Optimization and Applications*, vol. 32, no. 3, pp. 231–257, 2005.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 4723863, 16 pages
https://doi.org/10.1155/2017/4723863

Research Article

Memetic Computing Applied to the Design of Composite Materials and Structures

Jose Ignacio Pelaez,[1,2] Jose Antonio Gomez-Ruiz,[1,2] Jorge Veintimilla,[3] Gustavo Vaccaro,[2,4] and Patricia Witt[4]

[1] *Department of Languages and Computer Sciences, University of Malaga, Malaga, Spain*
[2] *Institute of Biomedical Research in Malaga (IBIMA), University of Malaga, Malaga, Spain*
[3] *Department of Civil Engineering, Materials and Production, University of Malaga, Malaga, Spain*
[4] *University of Guayaquil, Guayaquil, Ecuador*

Correspondence should be addressed to Jose Antonio Gomez-Ruiz; janto@lcc.uma.es

Received 5 April 2017; Revised 30 July 2017; Accepted 16 August 2017; Published 28 September 2017

Academic Editor: David Greiner

Presently, there exists an important need for lighter and more resistant structures, with reduced manufacturing costs. Laminated polymers are materials which respond to these new demands. Main difficulties of the design process of a composite laminate include the necessity to design both the geometry of the element and the material configuration itself and, therefore, the possibilities of creating composite materials are almost unlimited. Many techniques, ranging from linear programming or finite elements to computational intelligence, have been used to solve this type of problems. The aim of this work is to show that more effective and dynamic methods to solve this type of problems are obtained by using certain techniques based on systematic exploitation of knowledge of the problem, together with the combination of metaheuristics based on population as well as on local search. With this objective, a memetic algorithm has been designed and compared with the main heuristics used in the design of laminated polymers in different scenarios. All solutions obtained have been validated by the ANSYS® software package.

1. Introduction

A composite material is formed by the aggregation of two or more distinct materials to form a new one with enhanced properties: an agglomerate material known as the matrix and reinforcement materials that may be made up of continuous fibers, short fibers, or particles [1, 2]. Well-designed material adopts the best properties of its constituents and even some that none of these possess. The aim of composite materials design is to generate new cheaper materials with improved strength and lightness. Not all of them can be improved simultaneously; therefore, the design objective is to obtain a new material that offers the best possible adaptation to the required specifications.

The exceptional strength and lightness of these materials have led to the development of a vast number of applications, particularly in the aeronautical and space industries due to the economic significance of these properties. Figure 1 shows the markets for composite materials.

Laminate is a particularly important type of composite material made up of laminas of the same composition with unidirectional reinforcement fibers, stacked, and bound together by the same material that forms the matrix, but with distinct fiber orientation. Within this category, symmetric laminates are worthy of special attention, with both geometrical and structural symmetry relative to the mid-plane.

One of the main difficulties of the design process of a laminate is to design both the geometry of the element and the material configuration itself as to best exploit the qualities of the constituent materials. Designing process must also evaluate the deterioration of the laminate properties over time due to stress, which can lead to unanticipated behaviour (cracks) or failure of the structural element in question.

Synthesis and analysis have been traditionally carried out using empirical knowledge based methods [3]. This is partly because the number of possible combinations of composites is almost unlimited and also because characterization by experimentation is very expensive.

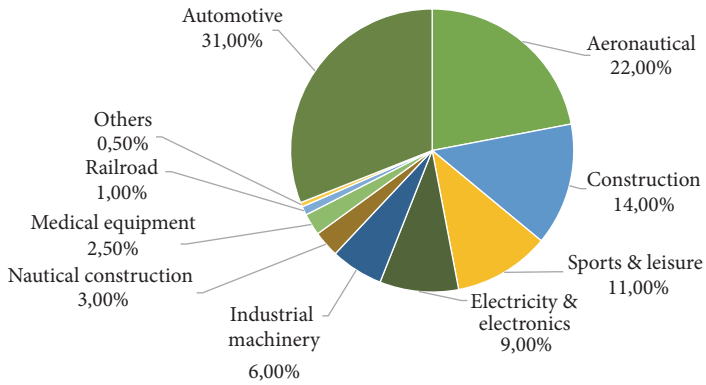


Figure 1: Markets for composite materials.

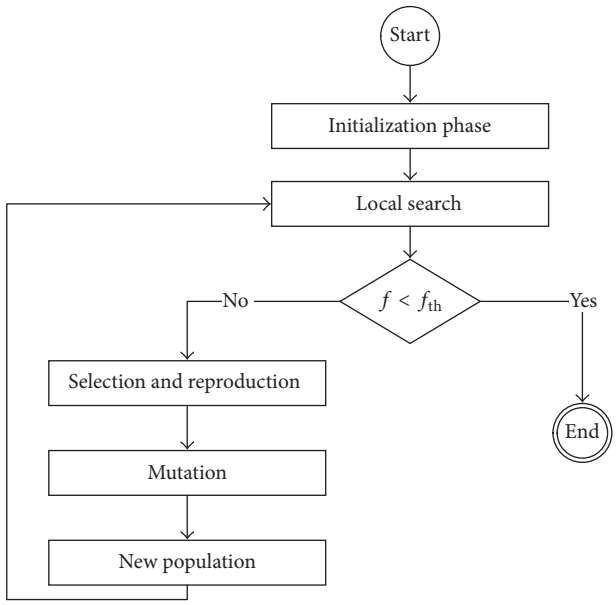


Figure 2: Scheme of the memetic algorithm. f_{th} denotes a fixed threshold value used for the stopping criterion.

Since the 1990s, different design systems which aim to overcome these limitations have been proposed. These proposals have involved approaches ranging from traditional techniques such as classical nonlinear optimization procedures combined with finite element modelling [4, 5], through generic task methods and case-based reasoning [6], to modern artificial intelligence techniques [7–16].

Memetic Computing or Memetic Algorithms (MA) have proven to be efficient at numerous situations [17–21]. Figure 2 shows the scheme for a typical MA [22]. Local search is performed between each generation, in addition to the techniques used by Genetic Algorithms to explore the search space, namely, recombination/crossover and mutation. It is performed to improve the fitness of the population (in a localized region of the solution space) so that the next generation has "better" genes from its parents.

The aim of this work is to show that more effective and dynamic methods to solve problems of laminated polymers design are obtained by using certain techniques based on systematic exploitation of knowledge of the problem, together with the combination of metaheuristics based on population as well as on local search. With this objective, a MA has been designed and thoroughly compared with the main heuristics used in the design of laminated polymers in different scenarios. This work has been organized as follows: in Section 2, a studio of the main metaheuristics which have been used in the design of composite laminates is performed; in Section 3, a model for the design and optimization of symmetric laminates based on a MA is presented; in Section 4, this model is subjected to an exhaustive analysis and is compared with other methods to prove its usefulness; in Section 5, the MA-based model is applied to specific cases. Finally, conclusions are presented in Section 6.

2. State of the Art of Composite Materials Design

2.1. Genetic Algorithms. Genetic Algorithms (GA) have been the most popular method in the design of laminated polymers and optimizing piling sequence [23]. Callahan and Weeks [24], Nagendra et al. [25], Le Riche and Haftka [26], and Ball et al. [27] have been the first to adopt and use GA for the design of piling sequences of composite laminated materials. GA has also been used in problems with different objective functions, such as strength [28, 29], buckling loads [9, 28, 30–34], dimensional stability [35], strain energy absorption [36], weight (either as a restriction or as an objective to minimize) [37, 38], bending/torsion connection, stiffness [36, 39], basic frequencies [34, 40–42], distortion [43], or finding laminate reference parameters [44].

GA have also been applied in the design of a variety of composite structures ranging from simple rectangular plates to complex geometrical sheets, such as sandwich panels [45], rigid sheets [46], bolt joints [47], and laminated cylindrical panels [34]. Similarly, GA have been combined with finite elements packages which analyse the response to tension and deformation of the composite material structure [43, 48, 49].

A combination of methods is sometimes used; for example, Park et al. [50] uses an approximate memory combined with a permutation operator and with the local learning/random mixture, in order to reduce function number and improve rate of convergence.

Some of the main problems regarding GA are deep computational necessity and premature convergence. In order to solve these drawbacks, increase the rate of convergence, reduce risk of premature convergence, and lessen function evaluation time, several modifications have been proposed, including the use of parallel computation [51, 52]; optimization of several levels (thick and thin level codification) [51]; introduction of problem-dependable operators [26]; layer adding or layer deleting, permutation, and interlaminar change [9, 46]; generalized elitist and mutation of thickness of laminate/material/angle of fiber [49]; recovery of previously evaluated solutions [10, 50]; using approximation methods for the evaluation function [33, 53, 54]; training artificial neural network [34, 42]; initial blood-related population; or an aging hierarchic structure [55].

Sargent et al. [56] compared GA with some raging algorithms (i.e., random search, raging search, and simulated annealing) and observed that GA obtained better solutions than raging search, but in some cases they were unable to find a solution. Sivakumar et al. [57] compared David, Fletcher, and Powell (DFP)'s Quasi-Newton Method and GA, applied to reduce weight of a laminated composite limited by its basic frequencies. It was concluded that DFP converged in a reduced number of iterations when restriction number was small. However, finding a feasible number was difficult when restrictions number increased. Also considering that DFP could not handle discreet variables, it was concluded that "GA was a better tool to optimize laminated composites."

Even though GA has been widely used in optimization of piling sequence, an important flaw is its low rate of convergence. GA is an evolutionary algorithm based on population and might require several generations before converging into a solution [58]. Each generation consists of a great number of function evaluations; therefore, it could require plenty of computational time besides being very expensive.

2.2. Simulated Annealing. After Genetic Algorithms, simulated annealing (SA) is the second most popular method for the optimization of piling sequence in laminated composite materials [59–62].

The main problem of this technique is the generation of a sequence of points that converges into a nonoptimal solution. In order to solve this lack, some modifications have been proposed, such as an increase in the probability of sample points far from the present point [63] or the use of a set of points at a time instead of only one [64]. In order to increase the rate of convergence, Genovese et al. [65] proposed a two-level algorithm, including a "global annealing" in which all the design variables were disrupted simultaneously and a "local annealing" in which only one design variable was disrupted at a time. The local annealing was performed after each iteration of the global annealing in an attempt to improve the testing point locally. It was found that its speed of convergence was greater than the one-level simulated annealing and comparable with the optimization method based on gradient implemented by the Sequential Linear Programming (SLP) method.

SA is a good choice for the general case of optimal selection of laminate; however, it cannot be programmed to take benefit of the advantages of specific properties of a certain problem. GA is, in this respect, more flexible, but it frequently takes more computational time [56, 61, 66]. It is not easy to generalize the previous conclusion, since there are other researches, such as Rama Mohan Rao and Shyju [58], which shows that SA had better computational efficiency and was better at finding a solution to other combinatorial problems.

2.3. Tabu Search. Tabu Search (TS) was implemented by Pai et al. [14] for the optimization of piling sequence of laminates subjected to bulging and resistance requests as well as for matrix rupture. Results were compared with GA and they showed a comparative solution, but computational time depended on the case.

It has been used combined with other techniques, such as the SA. Rama Mohan Rao and Arvind [67] added a TS in the SA obtaining a method called tabu embedded in simulated annealing (TSA). Optimization of piling sequence of laminated composites was solved by the TSA, whilst restrictions are administrated through a correction strategy. TSA was faster than classic SA although more memory is required.

2.4. Variable Neighbourhood Search. Variable neighbourhood search (VNS) has been recently applied to laminated polymers design. Corz et al. [68] propose and compare an algorithm based on a variable environments search that allows the design of the geometry and composition of symmetric laminated composite materials. In this work, the implementation of the algorithm is shown in detail: objective function, encoding, fitness, neighbourhood structures, and local search, as well as the data used in the examples: volume fraction, efforts, and coefficients. The proposed model is compared with other techniques such as GA, SA, and TS, showing more efficient results.

3. A Memetic Algorithm-Based Model for the Designing of Composite Materials

In this section, the optimization problem, an encoding scheme for the representation of the problem, a fitness function to evaluate the feasibility of solutions, and the reproductive and local search operators used are presented.

3.1. Optimization Problem. The optimization problem can be formulated as follows: find the material (fiber F and matrix M), the volume fraction, VF, and the laminate stacking sequence, $\theta_1, \ldots, \theta_k, \ldots, \theta_n$, with the purpose of maximizing the utilization of the laminate, thus achieving the lowest number of laminas n. The set of design variables is expressed as the vector $\widetilde{s} = (F, M, \mathrm{VF}, \theta_1, \ldots, \theta_k, \ldots, \theta_n)$. The degree of utilization of the laminate material is defined as a positive real number that is obtained from the fitness function (FF),

Table 1: Encoding used for a laminate.

IU	Values										
Fiber	A set of eleven fibres are used, each represented by an integer number										
	1	2	3	4	5	6	7	8	9	10	11
	E-Glass	S-Glass	AS-1	AS-4	IM-7	P-100	T-40	T-300	Boron	Kevlar-49	Kevlar-149
Matrix	A set of five matrices are used, each represented by an integer number										
	1		2		3			4		5	
	Peek		PPS		Polyester			Epoxy		Polyamide	
Volume fraction	A real number from the set {0.3, 0.4, 0.5, 0.6, 0.7}										
Geometry of the laminate	A sequence of integers, one for each lamina, in the range [−80°, +90°], in 10° intervals										

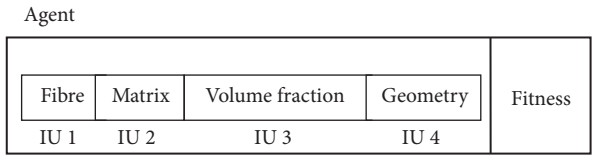


Figure 3: Structure of the laminate.

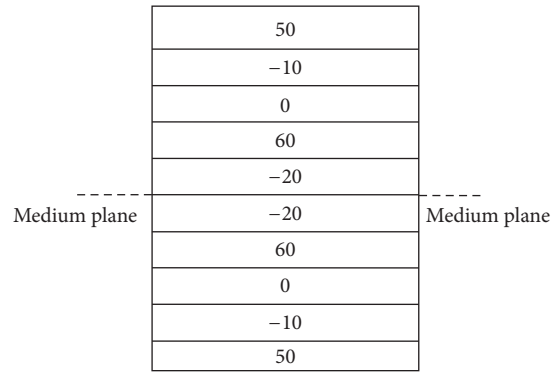


Figure 4: Symmetric laminate.

defined in the Section 3.2, to the $\tilde{s}$ vector. Therefore, the optimization problem and its corresponding constrains can be defined as follows:

$$\max \quad (\mathrm{FF}(\tilde{s})), \tag{1}$$

where

$n \geq 2$ (number of laminas);

VF $\in [0.3, 0.7]$ in 0.1 intervals (volume fraction);

$\theta_i \in [-80^\circ, 90^\circ]$ in 10° intervals (orientation of the laminas).

3.2. Encoding. The structure of the laminate will be represented by an agent formed by four information units (IU), which encode the fiber, the matrix, the volume fraction, and the laminate geometry (see Figure 3). Fitness, that shows how well adapted the solution is to its environment, is added to the structure.

Each of the four IU that define the laminate represents a different characteristic of its composition: fiber and matrix are represented by an integer number; volume fraction is represented by a real number; and geometry is represented by a sequence of integers that represent the direction of fibers. The values used for the different IU are shown in Table 1.

To represent the geometry of the laminate, we use the following notation: $[50/-10/0/60/-20]_s$. It is a symmetric laminate whose outer (first) lamina has an orientation of 50°, the second one (next) of -10°, and so on. A simplified representation of a symmetric laminate is shown in Figure 4.

Different failure criteria can be used to determine whether a lamina can withstand specific stress conditions without breaking [1]. In this article, the Tsai-Wu failure criteria [69] have been selected. In order to determine if a lamina breaks, for each lamina i of a laminate, its Tsai-Wu coefficient P_k^i is defined as

$$P_k^i = \left(\frac{1}{X} - \frac{1}{X'}\right) \cdot \sigma_1 + \left(\frac{1}{Y} - \frac{1}{Y'}\right) \cdot \sigma_2 + \frac{1}{X \cdot X'} \cdot \sigma_1^2 + \frac{1}{Y \cdot Y'} \cdot \sigma_2^2 - \sqrt{\frac{1}{X \cdot X' \cdot Y \cdot Y'}} \cdot \sigma_1 \cdot \sigma_2 + \frac{1}{S^2} \cdot \tau_{12}^2, \tag{2}$$

where X, X', Y, Y', and S are, respectively, the ultimate tensile and compressive strength in the fiber direction, the ultimate tensile and compressive transversal strength, and the shear strength. σ_1 and σ_2 are the stress resistance coefficients for the x- and y-axes, respectively, and τ_{12} is the stress coefficient of an orthotropic lamina under plane stress conditions. Applying the Tsai-Wu failure criteria [69], the lamina does not break if $P_k^i \leq 1, \forall i$, and breaks otherwise. Furthermore, if $P_k^i \cong 1$, the lamina i is working at full capacity. If a lamina is broken, then the entire laminate is discarded.

3.3. Fitness. The fitness function considers economic and safety criteria, so the alignment of stresses along the direction of the fibers is improved and the following are penalized: volume fraction and high laminate thickness; stacking more than four consecutive laminas with the same fiber orientation; the distribution of stress in directions other than that

Table 2: Fitness equation coefficients.

Coefficient	Description
P_1	Longitudinal coefficient along the direction of the fibres. It is a real value indicating the relationship between longitudinal stress in the direction of the fibres and the breakage stress in that direction. As the direction of the fibres is that of most resistance, it is of interest that this ratio is as high as possible. For this reason, it is in the numerator. β has been empirically determined as 2. $P_1 = \sum_i (\lvert\sigma_1^i / X^i\rvert \vee \lvert\sigma_1^i / X'^i\rvert)$ where the stress σ_1^i is either tensile or compressive and the sum applies to the laminas that do not break.
P_2	Longitudinal coefficient perpendicular to the direction of the fibres. It is a real value that indicates the relationship between the longitudinal stress perpendicular to the fibres and the breakage stress in that direction. As the direction which is perpendicular to the fibres is that of least resistance, it is of interest that this ratio is as small as possible. For this reason it is in the denominator. $P_2 = \sum_i (\lvert\sigma_2^i / Y^i\rvert \vee \lvert\sigma_2^i / Y'^i\rvert)$ where the stress σ_2^i is either tensile or compressive and the sum applies to all the laminas.
P_{12}	Shear coefficient. It is a real value that indicates the relationship between the shear stress in the plane and the maximum stress that can be tolerated in that plane. As the lamina has low resistance to these stresses, it is of interest that this ratio is as small as possible. For this reason it is in the denominator. The sum applies to all the laminas. $P_{12} = \sum_i \lvert\tau_{12}^i / S^i\rvert$
CLA	Coefficient that indicates whether the number of adjacent layers with the same orientation is less than or equal to four (CLA = 0) or greater than four (CLA = 10^{10}) It is in the denominator so as to heavily penalize this second possibility. Its values have been empirically determined.
VF	Fiber volume fraction of the laminate. As this value should tend towards small values that are more advantageous economically, it is placed in the denominator where small values will have a positive effect.
$n \cdot e$	Product of the number of laminas and its thickness (expressed in metres). γ has been empirically determined as 12.
R	It indicates the number of layers that break ($P_k^i > 1$), and is in the denominator in order to penalize such event. δ has been empirically determined as 4.

of the fibers; and lamina breakage. A higher value indicates a better solution. The fitness function, FF, is defined as

$$\mathrm{FF} = 10^{\alpha} \cdot \frac{P_1^{\beta}}{(\mathrm{CLA}+1)\cdot \mathrm{VF}\cdot (n\cdot e)^{\gamma}\cdot (R+1)^{\delta}\cdot (P_2+1)\cdot (P_{12}+1)}, \quad (3)$$

where the coefficients are described in Table 2. The multiplicative factor $10^{\propto}$ is to maintain fitness values between manageable values. α has been empirically determined as -28.

3.4. Reproductive Operator. Crossover operator is different for each IU In case of fiber, matrix, and volume fraction IU, a classic one-point crossover operator adapted to the different possible cases is used. However, for geometry IU, the classic crossover operator is not used because it is very static in changing the number of layers. To avoid this problem, a dynamic crossover operator with a different crossover point for each parent is proposed. Such crossover points are determined by generating two integer numbers from a continuous uniform distribution. The first number lies between 0 and the number of laminas of the first laminate, l_1, and the second between 1 and the number of laminas of the second laminate, l_2, plus 1. Then, the parent laminates are split into two parts, with the child joining together the left part of the first parent and the right part of the second parent being formed. Depending on the location of these crossover points, p_1 and p_2 can be obtained different cases shown in Table 3.

3.5. Local Search Operators. A set of local search operators are defined, following previous applications of a variable neighbourhood search-based model [68]. These operators allow a lamina to be added or removed, to change its orientation, type of matrix, and type of fiber or volume fraction. Two new operators have been defined in order to better exploit the region being explored in a given time. The operators proposed in the model are summarized in Table 4.

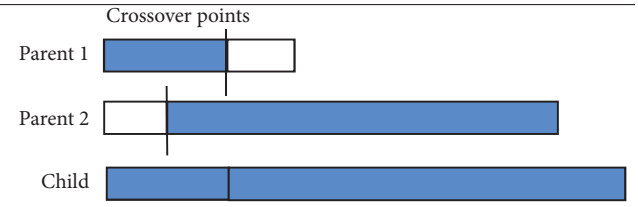
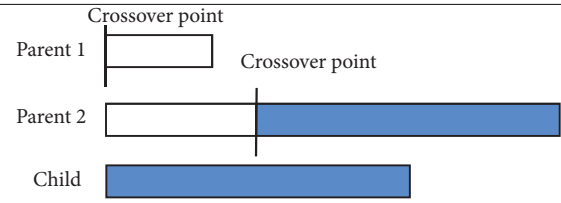
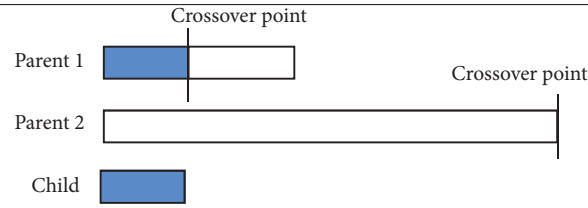
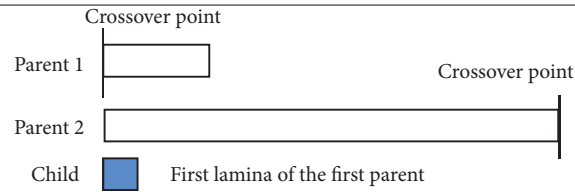
3.6. Implementation. The MA proposed model is compared with other heuristics in order to verify its advantages. These heuristics (detailed in Section 2) are GA, VNS, SA, and TS. All heuristics use the same representation of the problem proposed in this paper. Experiments were performed in the Picasso Computer at University of Malaga (512 IBM PowerPC 970 processors, 2994.04 GFlops).

In Tables 5 and 6, characteristics of the materials (fibers and matrices) used in the tests for the laminates design are shown.

4. Experiments and Results

The problem is to determine the composition and thickness of a laminated plate under a distributed N_x and N_y loading. Table 7 presents the statistics for 21 different loading cases ($n = 21$) and the 5 heuristics ($k = 5$) indicated in Section 3.6, with 500 simulations each. For each model, the results shown correspond to the smallest number of laminas of the material obtained, $\mathrm{NL}_{\min}$, the number of times this smallest number is

Table 3: Different cases for crossover operator.

Case	Scheme
(1) This is the general case. The child is formed by joining the $[1 \cdots p_1]$ laminas of the first parent, with the $[p_2 \cdots l_2]$ laminas of the second one. If the length of the child exceeds the maximum allowed, it is cut to achieve this ultimate size.	
(2) If the first crossover point, p_1, is zero, and the second crossover point, p_2, is not $l_2 + 1$, the child is formed with the $[p_2 \cdots l_2]$ laminas of the second parent.	
(3) If the first crossover point, p_1, is not zero, and the second crossover point, p_2, is $l_2 + 1$, the child is formed with the $[1 \cdots p_1]$ laminas of the first parent.	
(4) If the first crossover point, p_1, is zero, and the second crossover point, p_2, is $l_2 + 1$, the child is formed with the first lamina of the first parent.	

obtained, NL, and its average $\overline{\text{NL}}$. Also, the best (maximum) fitness obtained, FF_{max}, and its average $\overline{\text{FF}}$ are presented.

4.1. Statistical Significance Test. In order to explore the result of simulations a statistical significance test is applied. The choice of test depends upon what it is intended to study, as Demšar [70] showed that nonparametric tests are safer and more appropriate than parametric tests for comparisons between two or more algorithms on multiple data sets.

A null or no-effect hypothesis is to be formulated prior to the application of the test. It often supports the equality or absence of differences among the results of the algorithms and enables alternative hypotheses to be raised that support the opposite [71]. The null hypothesis can be represented by H_0 and the alternative hypothesis by H_a. The application of the tests leads to the computation of a statistic, which can be used to reject the null hypothesis at a given level of significance α. It is also possible to compute the smallest level of significance that results in the rejection of the null hypothesis. This level is the p value, which is the probability of obtaining a result at least as extreme as the one that was actually observed, assuming that the null hypothesis is true. The use of p values is often preferred over using only fixed α levels since they provide cleaner measures of how significant the result is (the smaller the p-value, the stronger the evidence against the null hypothesis is) [72].

The Friedman test [73, 74] is a nonparametric test with multiple comparisons that aims to detect significant differences between the behaviour of two or more algorithms. The null hypothesis for Friedman's test states equality of medians between the populations. The alternative hypothesis is defined as the negation of the null hypothesis, so it is nondirectional. The first step in calculating the test statistic is to convert the original results to ranks. They are computed using the following procedure (n is the number of problems considered where i is its associated index and k is the number of algorithms included in the comparison where j is its associated index):

(1) Gather observed results for each algorithm/problem pair.

(2) For each problem i, rank values from 1 (best result) to k (worst result). Denote these ranks as r_i^j $(1 \leq j \leq k)$.

(3) For each algorithm j, average the ranks obtained in all problems to obtain the final rank $R_j = (1/n) \sum_i r_i^j$.

Thus, it ranks the algorithms for each problem separately; the best performing algorithm should have the rank of 1, the second best rank 2, and so forth. Under the null hypothesis,

Table 4: Local search operators.

Operator	Description	Function
Number 1	It operates on the geometry IU of the laminate, varying the orientation of the laminas, moving each of them 10° closer to 0°. If the orientation of a lamina is 0°, it is left unchanged.	To obtain solutions with improved longitudinal stress tolerance along the x-axis better.
Number 2	It operates on the geometry IU of the laminate, varying the orientation of the laminas, moving each of them 10° closer to 90°. If the orientation of a lamina is 90°, it is left unchanged, and if it is −80° it is changed to 90°.	To obtain solutions with improved longitudinal stress tolerance along the y-axis better.
Number 3	It operates on the geometry IU of the laminate, varying the orientation of the laminas, moving each of them 10° closer to +45° or −45°, depending on whether it is positive or negative. If the orientation of a lamina is 40°, 50°, −40° or −50°, it is left unchanged, and if it is 0° it is changed to 10°.	To obtain solutions with improved shear stress tolerance.
Number 4	It operates on the geometry IU of the laminate, adding an individual lamina with random orientation and positioning. If the number of laminas is already the maximum possible, it is left unchanged.	To explore into the direction of the greatest number of laminas
Number 5	It operates on the geometry I.U. of the laminate, removing an individual lamina, chosen at random. If there is only one lamina, it is left unchanged.	To explore into the direction of the least number of laminas
Number 6	It operates on the geometry IU of the laminate, changing the orientation of a lamina, chosen at random and relocated in a new position, also chosen at random.	To explore into the direction of new orientations.
Number 7	It operates on the fiber IU, changing it by a distinct one, chosen at random, maintaining the matrix, the volume fraction, the number of laminas and their orientation constant.	To explore into the subspace solution of the fibres alone.
Number 8	It operates on the matrix I.U., changing it by a distinct one, chosen at random, maintaining the fiber, the volume fraction, the number of laminas and their orientation constant.	To explore into the subspace solution of the matrices alone.
Number 9	It operates on the volume fraction IU, changing its value for a distinct one, chosen at random, within the acceptable limits, maintaining the fiber, the matrix, the number of laminas and their orientation constant.	To explore into the subspace solution of the volume fraction alone.
Number 10	It operates on the geometry IU of the laminate, adding an individual lamina with orientation of –10° from the last positioned lamina. If the number of laminas is already the maximum possible, it is left unchanged.	To explore into the greatest number of laminas.
Number 11	It operates on the geometry IU of the laminate, removing an individual lamina, the closest to the last removed lamina, with different angular orientation. If there is only one lamina, it is left unchanged.	To explore into the smallest number of laminas.

which states that all the algorithms behave similarly (therefore their ranks R_j should be equal), the Friedman statistic F_f can be computed as

$$F_f = \frac{12n}{k(k+1)}\left[\left(\sum_j R_j^2\right) - \frac{k(k+1)^2}{4}\right] \quad (4)$$

which is distributed according to a chi-square distribution with $(k-1)$ degrees of freedom.

Iman and Davenport [75] derived a less conservative alternative with statistic distributed according to following F-distribution with $(k-1)$ and $(k-1)(n-1)$ degrees of freedom:

$$F_{\mathrm{ID}} = \frac{(n-1)\chi_F^2}{n(k-1) - \chi_F^2}. \quad (5)$$

Table 8 depicts the rank computed through the Friedman test for all heuristics considered in the exhaustive analysis (in our case of study $n = 21$ and $k = 5$) according to the average fitness, $\overline{F}$, of each distributed N_x and N_y loading. As can be deduced from Table 8, MA with a rank of 1 is the best

Table 5: Characteristics of fibers used in the design process.

Material	Module elastic E [GPa]	Resistance traction σ_f [MPa]	Coefficient Poisson υ
Glass fiber			
E-Glass	72.40	3450.00	0.20
S_Glass	86.90	4300.00	0.22
Carbon fiber			
AS-1	228.00	3100.00	0.20
AS-4	248.00	4070.00	0.20
IM-7	301.00	5310.00	0.20
P-100	758.00	2410.00	0.20
T-40	290.00	5650.00	0.20
T-300	231.00	3650.00	0.20
Boron fiber			
Boron	393.00	3100.00	0.20
Aramid fiber			
Kevlar 49	131.00	3620.00	0.35
Kevlar 149	179.00	3450.00	0.35

Table 6: Characteristics of matrices used in the design process.

Material	Module elastic E [GPa]	Resistance traction σ_f [MPa]	Coefficient Poisson υ
Thermoplastics			
Peek	3.24	100.00	0.40
PPS	3.30	82.70	0.33
Thermostables			
Polyester	3.00	50.00	0.30
Epoxy resin	4.60	58.60	0.36
Polyamide	3.50	103.00	0.35

performing algorithm, whereas SA with a rank of 4.948 is the worst.

The Friedman statistic computed by (4) (distributed according to chi-square with 4 degrees of freedom) is 82.552381 and its p value is $8.256673e-11$ and the Iman-Davenport extension computed by (5) (distributed according to F-distribution with 4 and 80 degrees of freedom) is 1140.526316 and its p value is $1.149264e-69$. In both cases, the p value is less than the significance level $\alpha = 0.05$; and H_0 for which there is no difference in rankings for these 5 heuristics is rejected.

Rejection of H_0 must be followed by a post hoc procedure to characterize the differences between algorithms. The aim of the application of post hoc procedures is to perform a comparison considering a control method and a set of algorithms. A family of hypotheses can be defined, all related to the control method. Then, the application of a post hoc test can lead to obtaining a p value which determines the degree of rejection of each hypothesis [76]. A family of hypotheses is a set of logically interrelated hypotheses of comparisons which, in $(1 \times n)$ comparisons, compares the $(k-1)$ algorithms of the study (excluding the control) with the control method, whereas, in $(n \times n)$ comparisons, it considers the $(k(k-1)/2)$ possible comparisons among algorithms. Therefore, the family will be composed of $(k-1)$ or $(k(k-1)/2)$ hypotheses, respectively, which can be ordered by its p value, from lowest to highest.

In our comparison, the appropriated post hoc procedure is the Nemenyi test [77] because multiple algorithms on multiple data sets are compared. In our case of study ($n = 21$ and $k = 5$) there are 10 hypotheses for $(5 \times 4)/2$ possible comparisons among algorithms. The p values for the Nemenyi test are shown in Table 9 ordered by p values, where in each row the first heuristic has better (lower) average ranking than the second. If the corresponding p value is smaller than $\alpha = 0.05$, then the first heuristic is significantly better than the second (H_0 hypothesis rejected).

As can be deduced from Table 9, the MA proposed model significantly performs better than the others because the H_0 hypothesis is rejected in all comparative pairs.

4.2. Analysis of Results. As indicated in Section 4.1, the MA proposed model is significantly better than the others heuristics respect to the fitness function. Moreover, in many cases, the proposed model obtains a material with the lowest number of layers and such occurrences are greater than the

Table 7: Statistics corresponding to the series of simulations. Biaxial loading $N_x : N_y$.

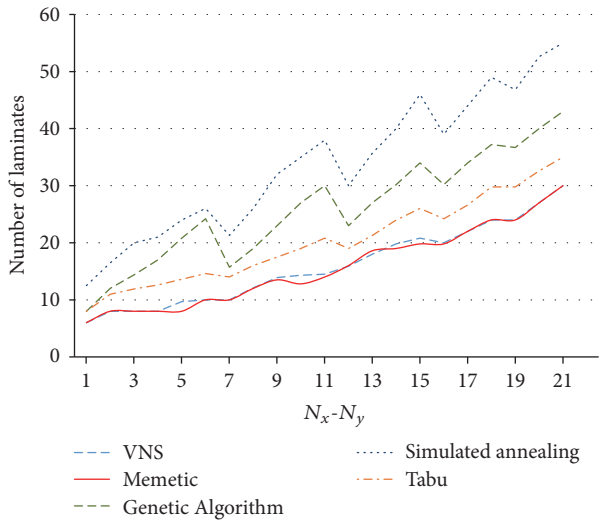
Loads	Algorithm	NL_{min}	NL	$\overline{NL}$	FF_{max}	$\overline{FF}$
N_x: 250.000 N/m N_y: 250.000 N/m	VNS	6	500	6	$2.8594 \cdot 10^8$	$2.7272 \cdot 10^8$
	MA	6	500	6	$2.8594 \cdot 10^8$	$2.8594 \cdot 10^8$
	AG	6	89	8.28	$2.8594 \cdot 10^8$	$6.969 \cdot 10^7$
	SA	8	76	12.456	$8.978 \cdot 10^6$	$1.1579 \cdot 10^5$
	TS	6	32	8.7120	$2.7920 \cdot 10^8$	$7.4150 \cdot 10^6$
N_x: 250.000 N/m N_y: 500.000 N/m	VNS	8	500	8	$1.4029 \cdot 10^7$	$1.2698 \cdot 10^7$
	MA	8	500	8	$1.4029 \cdot 10^7$	$1.3822 \cdot 10^7$
	AG	8	35	11.8	$1.3145 \cdot 10^7$	$1.2643 \cdot 10^6$
	SA	8	10	16.544	$5.6258 \cdot 10^6$	$2.0485 \cdot 10^4$
	TS	8	13	10.5680	$1.0056 \cdot 10^7$	$1.5439 \cdot 10^5$
N_x: 250.000 N/m N_y: 750.000 N/m	VNS	8	498	8.008	$6.6161 \cdot 10^6$	$6.3187 \cdot 10^6$
	MA	8	500	8	$6.6161 \cdot 10^6$	$6.6044 \cdot 10^6$
	AG	8	3	14.4733	$6.5042 \cdot 10^6$	$1.9654 \cdot 10^5$
	SA	10	26	19.532	$4.3045 \cdot 10^5$	$3.4465 \cdot 10^3$
	TS	8	3	11.5360	$5.2699 \cdot 10^6$	$3.5922 \cdot 10^4$
N_x: 250.000 N/m N_y: 1.000.000 N/m	VNS	8	471	8.116	$8.2968 \cdot 10^6$	$5.4898 \cdot 10^6$
	MA	8	500	8	$8.2968 \cdot 10^6$	$8.2815 \cdot 10^6$
	AG	10	18	17.1267	$7.6349 \cdot 10^5$	$6.3974 \cdot 10^4$
	SA	10	24	21.12	$6.7663 \cdot 10^5$	$2.5123 \cdot 10^3$
	TS	10	3	12.4680	$5.0663 \cdot 10^5$	$5.1883 \cdot 10^3$
N_x: 250.000 N/m N_y: 1.250.000 N/m	VNS	8	118	9.528	$6.6895 \cdot 10^6$	$1.866 \cdot 10^6$
	MA	8	500	8	$6.6895 \cdot 10^6$	$6.0696 \cdot 10^6$
	AG	10	2	20.52	$1.0673 \cdot 10^6$	$1.9896 \cdot 10^4$
	SA	12	3	23.672	$4.5062 \cdot 10^4$	$3.2251 \cdot 10^2$
	TS	10	1	13.6400	$5.9026 \cdot 10^5$	$3.0134 \cdot 10^3$
N_x: 250.000 N/m N_y: 1.500.000 N/m	VNS	10	496	10.016	$9.0479 \cdot 10^5$	$6.8493 \cdot 10^5$
	MA	10	500	10	$9.0479 \cdot 10^5$	$8.9861 \cdot 10^5$
	AG	12	14	24.26	$2.8056 \cdot 10^5$	$9.3622 \cdot 10^3$
	SA	10	22	26.24	$5.6722 \cdot 10^5$	$2.4689 \cdot 10^3$
	TS	12	4	14.7000	$8.7860 \cdot 10^4$	$1.0494 \cdot 10^3$
N_x: 500.000 N/m N_y: 500.000 N/m	VNS	10	400	10.4	$6.37 \cdot 10^5$	$5.0705 \cdot 10^5$
	MA	10	481	10.08	$6.37 \cdot 10^5$	$6.1494 \cdot 10^5$
	AG	10	3	15.6467	$5.6117 \cdot 10^5$	$6.2101 \cdot 10^4$
	SA	12	4	21.292	$1.5327 \cdot 10^5$	$1.1971 \cdot 10^3$
	TS	10	3	13.9240	$5.6117 \cdot 10^5$	$1.0223 \cdot 10^4$
N_x: 500.000 N/m N_y: 750.000 N/m	VNS	12	260	12.27	$8.9005 \cdot 10^4$	$7.3328 \cdot 10^4$
	MA	12	496	12.02	$9.0984 \cdot 10^4$	$8.7607 \cdot 10^4$
	AG	12	4	18.9467	$8.5403 \cdot 10^4$	$8.4764 \cdot 10^3$
	SA	16	3	26.456	$4.0543 \cdot 10^3$	30.7819
	TS	12	1	15.9200	$1.7488 \cdot 10^4$	$6.1199 \cdot 10^2$
N_x: 500.000 N/m N_y: 1.000.000 N/m	VNS	12	15	13.96	$3.6897 \cdot 10^4$	$1.3575 \cdot 10^4$
	MA	12	98	13.61	$3.6897 \cdot 10^4$	$1.8147 \cdot 10^4$
	AG	14	1	23.1267	$1.1361 \cdot 10^4$	$1.146 \cdot 10^3$
	SA	16	3	31.672	$5.6763 \cdot 10^3$	22.3771
	TS	14	3	17.5640	$5.2902 \cdot 10^3$	81.8767

Table 7: Continued.

Loads	Algorithm	NL_{min}	NL	$\overline{NL}$	FF_{max}	$\overline{FF}$
N_x: 500.000 N/m N_y: 1.250.000 N/m	VNS	12	72	14.28	$2.9117 \cdot 10^4$	$8.6122 \cdot 10^3$
	MA	12	339	12.81	$2.9117 \cdot 10^4$	$2.173 \cdot 10^4$
	AG	16	2	27.4133	$2.6399 \cdot 10^3$	$2.2388 \cdot 10^2$
	SA	18	2	35.388	$1.23 \cdot 10^3$	4.4746
	TS	16	1	19.2560	$9.2541 \cdot 10^2$	23.4648
N_x: 500.000 N/m N_y: 1.500.000 N/m	VNS	14	363	14.636	$1.1324 \cdot 10^4$	$5.3271 \cdot 10^3$
	MA	14	494	14.02	$1.1324 \cdot 10^4$	$1.0644 \cdot 10^4$
	AG	18	10	30.1067	$8.7844 \cdot 10^2$	85.4087
	SA	16	2	38.476	$9.3088 \cdot 10^2$	4.1987
	TS	16	1	20.3560	$8.7751 \cdot 10^2$	18.3449
N_x: 750.000 N/m N_y: 750.000 N/m	VNS	16	500	16	$3.522 \cdot 10^3$	$3.3743 \cdot 10^3$
	MA	16	500	16	$3.522 \cdot 10^3$	$3.522 \cdot 10^3$
	AG	16	15	22.7667	$3.522 \cdot 10^3$	$7.5575 \cdot 10^2$
	SA	16	2	29.65	$3.1351 \cdot 10^3$	18.3958
	TS	16	16	18.9880	$3.5220 \cdot 10^3$	$1.4030 \cdot 10^2$
N_x: 750.000 N/m N_y: 1.000.000 N/m	VNS	18	500	18	$1.0035 \cdot 10^3$	$9.2486 \cdot 10^2$
	MA	18	500	18	$1.0073 \cdot 10^3$	$9.8167 \cdot 10^2$
	AG	18	16	27.4133	$9.7849 \cdot 10^2$	$1.8292 \cdot 10^2$
	SA	20	4	35.736	$3.3870 \cdot 10^2$	2.003
	TS	18	9	21.4320	$8.8434 \cdot 10^2$	24.9275
N_x: 750.000 N/m N_y: 1.250.000 N/m	VNS	18	38	19.848	$5.2267 \cdot 10^2$	$3.1132 \cdot 10^2$
	MA	16	7	18.49	$5.2345 \cdot 10^2$	$4.3804 \cdot 10^2$
	AG	20	10	30.52	$3.2510 \cdot 10^2$	54.4953
	SA	20	2	40.38	$1.8789 \cdot 10^2$	1.3868
	TS	20	2	23.7400	$1.9273 \cdot 10^2$	2.8100
N_x: 750.000 N/m N_y: 1.500.000 N/m	VNS	18	11	20.708	$3.3956 \cdot 10^2$	$1.4176 \cdot 10^2$
	MA	18	63	19.776	$3.3956 \cdot 10^2$	$1.9003 \cdot 10^2$
	AG	22	16	34.3	$1.0962 \cdot 10^2$	16.2374
	SA	26	1	45.928	5.8144	0.032
	TS	22	3	25.8480	43.3493	0.7962
N_x: 1.000.000 N/m N_y: 1.000.000 N/m	VNS	20	500	20	$3.2888 \cdot 10^2$	$3.1014 \cdot 10^2$
	MA	20	500	20	$3.2888 \cdot 10^2$	$3.2887 \cdot 10^2$
	AG	20	5	30.44	$2.9623 \cdot 10^2$	44.4987
	SA	20	2	38.96	$2.2611 \cdot 10^2$	1.8568
	TS	20	1	24.3960	$1.9686 \cdot 10^2$	5.8882
N_x: 1.000.000 N/m N_y: 1.250.000 N/m	VNS	22	500	22	$1.1748 \cdot 10^2$	$1.0838 \cdot 10^2$
	MA	22	500	22	$1.1785 \cdot 10^2$	$1.1536 \cdot 10^2$
	AG	22	9	34.2467	$1.1539 \cdot 10^2$	12.4818
	SA	26	3	44.54	22.4217	0.1669
	TS	22	3	26.7880	$1.0333 \cdot 10^2$	1.7983
N_x: 1.000.000 N/m N_y: 1.500.000 N/m	VNS	24	500	24	46.2995	42.4606
	MA	24	500	24	46.4240	45.1922
	AG	24	8	37.9467	44.3415	4.5575
	SA	28	7	49.4	13.0737	0.165
	TS	24	1	29.6360	19.4564	0.2810

Table 7: Continued.

Loads	Algorithm	NL_{min}	NL	$\overline{NL}$	FF_{max}	$\overline{FF}$
N_x: 1.250.000 N/m N_y: 1.250.000 N/m	VNS	24	469	24.124	46.643	41.4749
	MA	24	500	24	46.643	46.6245
	AG	24	3	37.0533	42.1974	4.8935
	SA	26	1	46.908	13.6039	0.0983
	TS	26	10	29.3760	15.1788	0.7773
N_x: 1.250.000 N/m N_y: 1.500.000 N/m	VNS	26	130	27.496	9.8711	7.8726
	MA	26	231	27.144	10.1457	8.3843
	AG	28	16	40.2333	7.8922	1.6101
	SA	30	2	52.46	4.6966	0.0335
	TS	28	7	32.5520	7.0430	0.2389
N_x: 1.500.000 N/m N_y: 1.500.000 N/m	VNS	30	500	30	3.8763	3.5545
	MA	30	500	30	3.8763	3.8245
	AG	30	10	42.86	3.7687	0.6741
	SA	30	1	54.748	3.1197	0.015
	TS	38	8	35.3000	2.3280	0.1268



Abscissa	N_x (N/m)	N_y (N/m)
1	250.000	250.000
2	250.000	500.000
3	250.000	750.000
4	250.000	1.000.000
5	250.000	1.250.000
6	250.000	1.500.000
7	500.000	500.000
8	500.000	750.000
9	500.000	1.000.000
10	500.000	1.250.000
11	500.000	1.500.000
12	750.000	750.000
13	750.000	1.000.000
14	750.000	1.250.000
15	750.000	1.500.000
16	1.000.000	1.000.000
17	1.000.000	1.250.000
18	1.000.000	1.500.000
19	1.250.000	1.250.000
20	1.250.000	1.500.000
21	1.500.000	1.500.000

Figure 5: Average number of layers for the different algorithms.

Table 8: Average rankings of algorithms achieved by Friedman test.

Algorithm	Average rankings
MA	1
VNS	2
GA	3
TS	4.095
SA	4.948

solutions obtained by the others heuristics (see Table 7). Figure 5 shows the average number of laminas corresponding to the laminates obtained with all heuristics.

As an example of results, in Table 10 are shown the specifications of the best laminate obtained with the 5 heuristics in the loading case N_x = 1.000.000 N/m, N_y = 1.500.000 N/m.

Finally, Figure 6 shows graphically the different coefficients (P_k^i or Tsai-Wu, P_1, P_2, and P_{12}) corresponding to the laminas of the best laminate obtained with the MA-based model in the loading case shown in Table 10. In this case, a large degree of uniformity of the coefficients P_k^i around 1 can be observed. It indicates an excellent exploitation of all the laminas.

5. Application to Specific Cases

Previously, the proposed model has been applied to general cases. In this section, the proposed model is applied to two

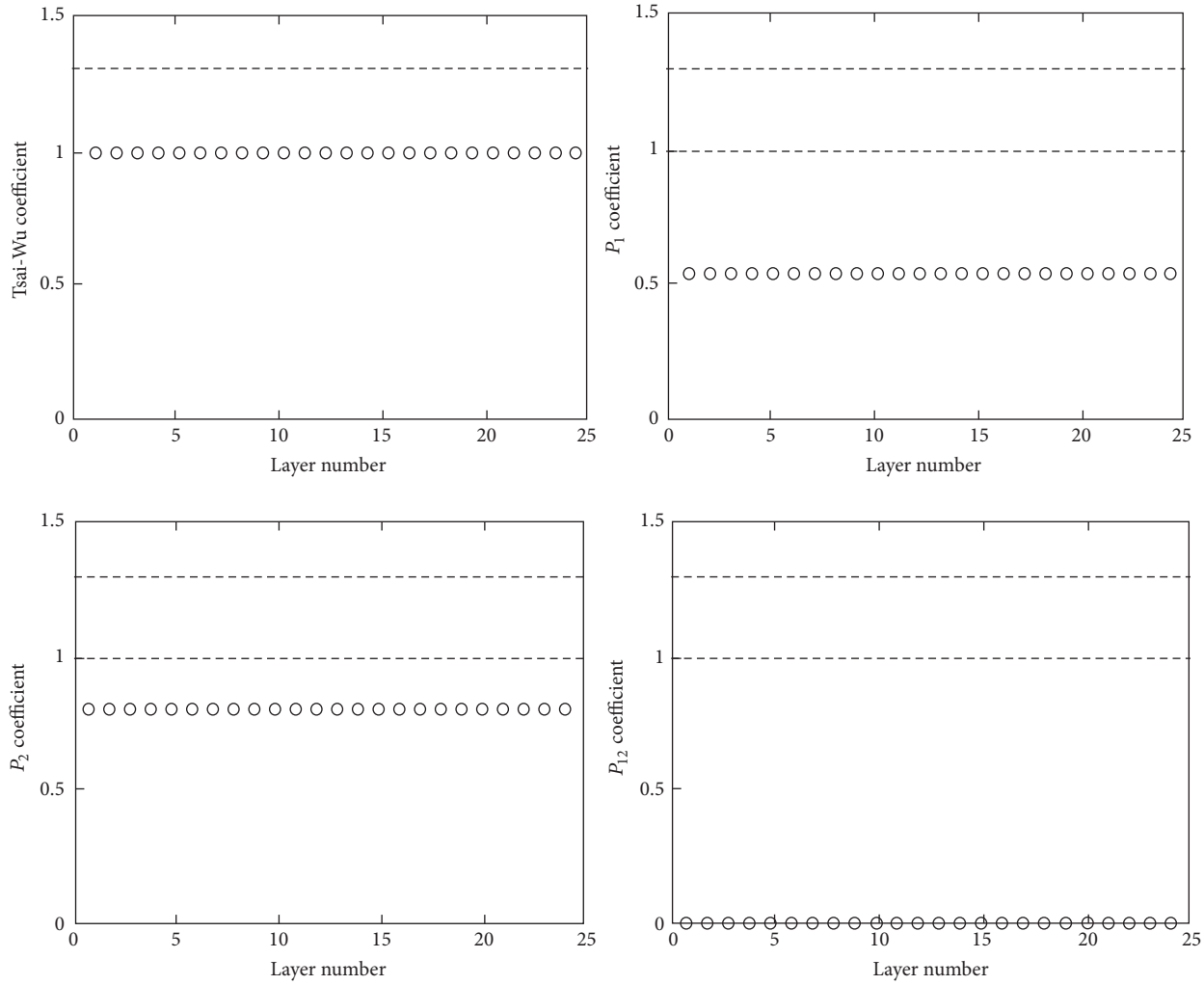


Figure 6: Coefficients P_k^i (Tsai-Wu), P_1, P_2, and P_{12} of each laminate in the best laminate obtained by the MA-based model.

Table 9: p values for Nemenyi test.

Hypothesis	p_{Nemenyi}
MA vs. SA	0
MA vs. TS	0
VNS vs. SA	0
VNS vs. TS	0.000018
MA vs. GA	0.000042
GA vs. SA	0.000095
GA vs. TS	0.024796
MA vs. VNS	0.040424
VNS vs. GA	0.040424
SA vs. TS	0.097110

significant specific cases comparing the results with another author's solution.

5.1. Minimum Thickness Design. Le Riche and Haftka [26] developed a GA for minimum thickness composite laminate design. They consider a graphite-epoxy plate with the load case with $N_x = 12.500\,\text{lb/in} = 2.189.956\,\text{N/m}$ and $N_y = 3.125\,\text{lb/in} = 547.890\,\text{N/m}$. The characteristics of the best laminated obtained by authors are shown in Table 11.

The laminate provided by the MA proposed model, under the same load case, is shown in Table 12.

5.2. Stacking Sequence Design. Liu et al. [54] developed a permutation GA for stacking sequence design of composite laminates. They consider a graphite-epoxy laminate with the load case with $N_x = -10.000\,\text{lb/in} = -1.751.965\,\text{N/m}$ and $N_y = -2.000\,\text{lb/in} = -350.393\,\text{N/m}$; $N_{xy} = 1.000\,\text{lb/in} = 175.197\,\text{N/m}$. The characteristics of the best laminated obtained by authors are shown in Table 13.

The laminate provided by the MA proposed model, under the same load case, is shown in Table 14.

The laminate represented in Tables 12 and 14 has been validated using a finite elements simulation software. To do so we employed the ANSYS 17.0 software package; additionally, the SHELL281 was used as the element of validation code, maintaining the same conditions and loadings when possible, and its characteristics can be consulted in [78].

Table 10: Characteristics of the best laminate obtained with each algorithm.

Algorithm	Fiber	Matrix	V_f	Fitness	Stacking sequence	Thickness (mm)
MA	P-100	Peek	0.40	46.424	$[0^\circ\ 70^\circ\ -40^\circ\ 40^\circ\ 0^\circ\ 80^\circ\ -80^\circ\ -70^\circ\ -80^\circ\ 40^\circ\ -40^\circ\ 80^\circ]_S$	4.32
VNS	P-100	Peek	0.40	46.229	$[-40^\circ\ -60^\circ\ 70^\circ\ 0^\circ\ 80^\circ\ 0^\circ\ 60^\circ\ -80^\circ\ -80^\circ\ -40^\circ\ 50^\circ\ 40^\circ]_S$	4.32
GA	P-100	Peek	0.40	44.341	$[-50^\circ\ -60^\circ\ 70^\circ\ -10^\circ\ 80^\circ\ 0^\circ\ 60^\circ\ -80^\circ\ -80^\circ\ -40^\circ\ 40^\circ\ 50^\circ]_S$	4.32
SA	P-100	Peek	0.30	13.073	$[-50^\circ\ 40^\circ\ 40^\circ\ 60^\circ\ -40^\circ\ 50^\circ\ -40^\circ\ -40^\circ\ -40^\circ\ 60^\circ\ 50^\circ\ 80^\circ\ 80^\circ\ -70^\circ]_S$	5.04
TA	P-100	Peek	0.40	19.456	$[50^\circ\ -40^\circ\ -50^\circ\ -40^\circ\ 50^\circ\ 50^\circ\ 60^\circ\ -60^\circ\ -60^\circ\ 50^\circ\ 50^\circ\ -50^\circ]_S$	4.32

Table 11: Characteristics of the best laminate obtained by Le Riche and Haftka's GA.

Characteristic	Value
Stacking sequence	$[\pm 45_2/90_2/\pm 45_3/0_2/\pm 45/0_4/\pm 45/0_2]_s$
N° of Laminas	48
Laminate thickness	0.240 in = 6.096 mm
FF_{max}	38.002

Table 12: Characteristics of the best laminate obtained by the MA proposed model.

Characteristic	Value
Stacking sequence	$[30_4/-20_2/-30_4/20_2/30_2/-30_2]_s$
Number of laminas	32
Laminate thickness	0.234 in = 5.94 mm
FF_{max}	42.001

Table 13: Characteristics of the best laminate obtained by Liu et al. permutation GA.

Characteristic	Value
Stacking Sequence	$[0_7/45_{15}/90_7]_s$
N° of Laminas	58
Laminate Thickness	0.290 in = 7.366 mm
FF_{max}	29.002

Table 14: Characteristics of the best laminate obtained with the MA proposed model.

Characteristic	Value
Stacking Sequence	$[0_4/-40_3/10_3/0_4/-40_3/0]_s$
N° of Laminas	42
Laminate Thickness	0.275 in = 6.985 mm
FF_{max}	40.010

6. Conclusions

In this paper, we have shown that the use of techniques based on systematic exploitation of knowledge of the problem, together with the combination of metaheuristics based on population as well as in local search, offers more effective and dynamic methods for the solution of laminated polymers design. To this end, a study of the main techniques used in the design of laminated polymers was firstly performed, in order to carry out a comparative analysis.

A Memetic Computing-based model has been presented for the design of Symmetric Laminated Composites and Structures. This model implements a general encoding for the design of composites and a fitness function that has taken into account economic and safety criteria in design. Also, a dynamic reproductive operator is presented, in which the classic crossover operator is modified in order to improve the dynamics in the change of the number of layers.

Finally, a set of local search operators are implemented. These allow a lamina to be added or removed, to change its orientation, type of matrix, type of fiber, or volume fraction. Two new operators have been defined and added to the set to improve the exploitation of the solutions in the region of the search space.

The proposed Memetic Computing model has been subjected to a broad analysis and applied in two specific cases. Firstly, the model has been applied to the design of a plate under distributed N_x and N_y loading, and the results have been compared with those obtained by other well-known design methods. In most cases, the minimum number and average number of laminas are lesser. Also, the proposed model has been compared with two significant models found in the literature, obtaining comparable results. These results show that the proposed model is a general and flexible design method for Symmetric Laminated Composites and Structures. We consider that these results can be applied to real-life scenarios with a high reliability. Finally, all solutions obtained by the MA proposed model in Sections 5.1 and 5.2 have been validated by the ANSYS software package using the same conditions and load system.

Conflicts of Interest

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

Acknowledgments

This work was partially supported by the Department of Research, Area of Publications, Critical Mass and Patents of the University of Guayaquil, Ecuador.

References

[1] M. Kashtalyan, *Introduction to Composite Materials Design*, vol. 115, CRC Press, Boca Raton, Fla, USA, 2nd edition, 2010.

[2] Z. Gürdal, R. T. Haftka, P. Hajela, and Z. Gürdal, *Design and Optimization of Laminated Composite Materials*, John Wiley and Sons, 1999.

[3] L. Grosset, R. LeRiche, and R. T. Haftka, "A double-distribution statistical algorithm for composite laminate optimization," *Structural and Multidisciplinary Optimization*, vol. 31, no. 1, pp. 49–59, 2006.

[4] J. Huang and R. T. Haftka, "Optimization of fiber orientations near a hole for increased load-carrying capacity of composite laminates," *Structural and Multidisciplinary Optimization*, vol. 30, no. 5, pp. 335–341, 2005.

[5] F. L. Matthews, G. A. O. Davies, D. Hitchings, and C. Soutis, *Finite Element Modelling of Composite Materials and Structures*, Woodhead Publishing Limited, 2000.

[6] T. J. Lenz, *Designing Composite Material Systems Using Generic Tasks and Case-based Reasoning*, Michigan State University, 1997.

[7] D. B. Adams, L. T. Watson, and Z. Gürdal, "Optimization and blending of composite laminates using genetic algorithms with migration," *Mechanics of Advanced Materials and Structures*, vol. 10, no. 3, pp. 183–203, 2003.

[8] E. Falkenauer, *Genetic Algorithms and Grouping Problems*, John Wiley and Sons, New York, NY, USA, 1998.

[9] G. Soremekun, Z. Gürdal, R. T. Haftka, and L. T. Watson, "Composite laminate design optimization by genetic algorithm with generalized elitist selection," *Computers and Structures*, vol. 79, no. 2, pp. 131–143, 2001.

[10] N. Kogiso, L. T. Watson, Z. Gürdal, and R. T. Haftka, "Genetic algorithms with local improvement for composite laminate design," *Structural Optimization*, vol. 7, no. 4, pp. 207–218, 1994.

[11] M. T. Mcmahon and L. T. Watson, "A distributed genetic algorithm with migration for the design of composite laminate structures," *Parallel Algorithms and Applications*, vol. 14, no. 4, pp. 329–362, 2000.

[12] S. Rajeev and C. S. Krishnamoorthy, "Discrete optimization of structures using genetic algorithms," *Journal of Structural Engineering*, vol. 118, no. 5, pp. 1233–1250, 1992.

[13] K. Yamazaki, "Two-level optimization technique of composite laminate panels by genetic algorithms," in *Proceedings of the 37th AIAA/ASME/ASCE/AHS/ASC Structure, Structural Dynamics and Materials Conference, 1996*, pp. 1882–1887, usa, April 1996.

[14] N. Pai, A. Kaw, and M. Weng, "Optimization of laminate stacking sequence for failure load maximization using Tabu search," *Composites Part B: Engineering*, vol. 34, no. 4, pp. 405–413, 2003.

[15] Y. Hirano and A. Todoroki, "Stacking sequence optimizations for composite laminates using fractal branch and bound method: Application for supersonic panel flutter problem with buckling load condition," *Advanced Composite Materials: The Official Journal of the Japan Society of Composite Materials*, vol. 13, no. 2, pp. 89–106, 2004.

[16] T.-U. Kim and H.-C. Sin, "Optimal design of composite laminated plates with the discreteness in ply angles and uncertainty in material properties considered," *Computers and Structures*, vol. 79, no. 29-30, pp. 2501–2509, 2001.

[17] R. Dawkins, *The Selfish Gene: 30th Anniversary Edition*, OUP Oxford, Oxford, USA, 2006.

[18] P. Moscato and C. Cotta Porras, "An introduction to memetic algorithms," *Inteligencia Artificial: Revista Iberoamericana de Inteligencia Artificial*, vol. 7, no. 19, pp. 131–148, 2003.

[19] C. Cotta, "Memetic algorithms with partial Lamarckism for the shortest common supersequence problem," *Lecture Notes in Computer Science*, vol. 3562, pp. 84–91, 2005.

[20] M. Gen, "Parallel machine scheduling problems using memetic algorithms," *Computers and Industrial Engineering*, vol. 33, pp. 761–764, 1997.

[21] Buriol, L. S., M. G. C. Resende, C. C. Ribeiro and M. Thorup. 2002. "A Memetic Algorithm for OSPF Routing." Proceedings of the sixth INFORMS Conference on Telecommunications, Boca Raton, Florida, 187-188.

[22] M. Pastorino, S. Caorsi, A. Massa, and A. Randazzo, "Reconstruction algorithms for electromagnetic imaging," *IEEE Transactions on Instrumentation and Measurement*, vol. 53, no. 3, pp. 692–699, 2004.

[23] Venkataraman, S. and R. T. Haftka. 1995. "Optimization of composite panels – a review." Proceedings of the American society of composites-14th annual technical conference. Fairborn. OH. 479–488.

[24] K. J. Callahan and G. E. Weeks, "Optimum design of composite laminates using genetic algorithms," *Composites Engineering*, vol. 2, no. 3, pp. 149–160, 1992.

[25] S. Nagendra, R. T. Haftka, and Z. Gürdal, "Stacking sequence optimization of simply supported laminates with stability and strain constraints," *AIAA Journal*, vol. 30, no. 8, pp. 2132–2137, 1992.

[26] R. Le Riche and R. T. Haftka, "Improved genetic algorithm for minimum thickness composite laminate design," *Composites Engineering*, vol. 5, no. 2, pp. 143–161, 1995.

[27] N. R. Ball, P. M. Sargent, and D. O. Ige, "Genetic algorithm representations for laminate layups," *Artificial Intelligence in Engineering*, vol. 8, no. 2, pp. 99–108, 1993.

[28] R. Le Riche and R. T. Haftka, "Optimization of laminate stacking sequence for buckling load maximization by genetic algorithm," *AIAA journal*, vol. 31, no. 5, pp. 951–956, 1993.

[29] J. H. Park, J. H. Hwang, C. S. Lee, and W. Hwang, "Stacking sequence design of composite laminates for maximum strength using genetic algorithms," *Composite Structures*, vol. 52, no. 2, pp. 217–231, 2001.

[30] N. Kogiso, L. T. Watson, Z. GÜrdal, R. T. Haftka, and S. Nagendra, "Design of composite laminates by a genetic algorithm with memory," *Mechanics of Composite Materials and Structures*, vol. 1, no. 1, pp. 95–117, 1994.

[31] A. Todoroki, "Stacking sequence matching by two-stage genetic algorithm with consanguineous initial population," in *Proceedings of the 38th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference*, pp. 1297–1302, Kissimmee, FL, USA, April 1997.

[32] T. Messager, M. Pyrz, B. Gineste, and P. Chauchot, "Optimal laminations of thin underwater composite cylindrical vessels," *Composite Structures*, vol. 58, no. 4, pp. 529–537, 2002.

[33] A. Todoroki and T. Ishikawa, "Design of experiments for stacking sequence optimizations with genetic algorithm using response surface approximation," *Composite Structures*, vol. 64, no. 3-4, pp. 349–357, 2004.

[34] M. Abouhamze and M. Shakeri, "Multi-objective stacking sequence optimization of laminated cylindrical panels using a genetic algorithm and neural networks," *Composite Structures*, vol. 81, no. 2, pp. 253–263, 2007.

[35] R. Le Riche and J. Gaudin, "Design of dimensionally stable composites by evolutionary optimization," *Composite Structures*, vol. 41, no. 2, pp. 97–111, 1998.

[36] E. Potgieter and N. Stander, "The genetic algorithm applied to stiffness maximization of laminated plates: Review and comparison," *Structural Optimization*, vol. 15, no. 3-4, pp. 221–229, 1998.

[37] I. Rajendran and S. Vijayarangan, "Optimal design of a composite leaf spring using genetic algorithms," *Computers and Structures*, vol. 79, no. 11, pp. 1121–1129, 2001.

[38] G. Narayana Naik, S. Gopalakrishnan, and R. Ganguli, "Design optimization of composites using genetic algorithms and failure mechanism based failure criterion," *Composite Structures*, vol. 83, no. 4, pp. 354–367, 2008.

[39] A. Todoroki, K. Watanabe, and H. Kobayashi, "Application of genetic algorithms to stiffness optimization of laminated composite plates with stress-concentrated open holes," *JSME International Journal, Series A: Mechanics and Material Engineering*, vol. 38, no. 4, pp. 458–464, 1995.

[40] K. Sivakumar, N. G. R. Iyengar, and K. Deb, "Optimum design of laminated composite plates with cutouts using a genetic algorithm," *Composite Structures*, vol. 42, no. 3, pp. 265–279, 1998.

[41] R. T. Haftka and Z. Gürdal, *Elements of Structural Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1992.

[42] M. Kemal Apalak, M. Yildirim, and R. Ekici, "Layer optimisation for maximum fundamental frequency of laminated composite plates for different edge conditions," *Composites Science and Technology*, vol. 68, no. 2, pp. 537–550, 2008.

[43] M. Walker and R. E. Smith, "A technique for the multiobjective optimisation of laminated composite structures using genetic algorithms and finite element analysis," *Composite Structures*, vol. 62, no. 1, pp. 123–128, 2003.

[44] A. Todoroki and R. T. Haftka, "Stacking sequence optimization by a genetic algorithm with a new recessive gene like repair strategy," *Composites Part B: Engineering*, vol. 29, no. 3, pp. 277–285, 1998.

[45] C.-C. Lin and Y.-J. Lee, "Stacking sequence optimization of laminated composite structures using genetic algorithm with local improvement," *Composite Structures*, vol. 63, no. 3-4, pp. 339–345, 2004.

[46] S. Nagendra, D. Jestin, Z. Gürdal, R. T. Haftka, and L. T. Watson, "Improved genetic algorithm for the design of stiffened composite panels," *Computers and Structures*, vol. 58, no. 3, pp. 543–555, 1996.

[47] V. Kradinov, E. Madenci, and D. R. Ambur, "Application of genetic algorithm for optimum design of bolted composite lap joints," *Composite Structures*, vol. 77, no. 2, pp. 148–159, 2007.

[48] A. Muc and W. Gurba, "Genetic algorithms and finite element analysis in optimization of composite structures," *Composite Structures*, vol. 54, no. 2-3, pp. 275–281, 2001.

[49] D. J. Deka, G. Sandeep, D. Chakraborty, and A. Dutta, "Multiobjective optimization of laminated composites using finite element method and genetic algorithm," *Journal of Reinforced Plastics and Composites*, vol. 24, no. 3, pp. 273–285, 2005.

[50] C. H. Park, W. I. Lee, W. S. Han, and A. Vautrin, "Improved genetic algorithm for multidisciplinary optimization of composite laminates," *Computers and Structures*, vol. 86, no. 19-20, pp. 1894–1903, 2008.

[51] W. F. Punch III, R. C. Averill, E. D. Goodman, S.-C. Lin, Y. Ding, and Y. C. Yip, "Optimal design of laminated composite structures using coarse-grain parallel genetic algorithms," *Computing Systems in Engineering*, vol. 5, no. 4-6, pp. 415–423, 1994.

[52] P. Kere and J. Lento, "Design optimization of laminated composite structures using distributed grid resources," *Composite Structures*, vol. 71, no. 3-4, pp. 435–438, 2005.

[53] V. B. Gantovnik, Z. Gürdal, and L. T. Watson, "A genetic algorithm with memory for optimal design of laminated sandwich composite panels," *Composite Structures*, vol. 58, no. 4, pp. 513–520, 2002.

[54] B. Liu, R. T. Haftka, and M. A. Akgün, "Composite wing structural optimization using genetic algorithms and response surfaces," in *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 1998*, pp. 1–12, MO, USA, September 1998.

[55] C. A. Conceicao Antonio, "A hierarchical genetic algorithm with age structure for multimodal optimal design of hybrid composites," *Structural and Multidisciplinary Optimization*, vol. 31, no. 4, pp. 280–294, 2006.

[56] P. M. Sargent, D. O. Ige, and N. R. Ball, "Design of laminate composite layups using genetic algorithms," *Engineering with Computers*, vol. 11, no. 2, pp. 59–69, 1995.

[57] K. Sivakumar, N. G. R. Iyengar, and D. Kalyanmoy, "Optimization of composite laminates with cutouts using genetic algorithm, variable metric and complex search methods," *Engineering Optimization*, vol. 32, no. 5, pp. 635–657, 2000.

[58] A. Rama Mohan Rao and P. P. Shyju, "Development of a hybrid meta-heuristic algorithm for combinatorial optimisation and its application for optimal design of laminated composite cylindrical skirt," *Computers and Structures*, vol. 86, no. 7-8, pp. 796–815, 2008.

[59] V. Savic, M. E. Tuttle, and Z. B. Zabinsky, "Optimization of composite I-sections using fiber angles as design variables," *Composite Structures*, vol. 53, no. 3, pp. 265–277, 2001.

[60] M. Lombardi, R. Haftka, and C. Cinquini, "Optimization of composite plates for buckling by simulated annealing," in *Proceedings of 33rd AIAA/ASME/AHS structures, structural dynamics and materials conference*, AIAA Paper 92-2314-CP, Dallas, TX, USA.

[61] D. Sadagopan and R. Pitchumani, "Application of genetic algorithms to optimal tailoring of composite materials," *Composites Science and Technology*, vol. 58, no. 3-4, pp. 571–589, 1998.

[62] A. C. Arvin and C. E. Bakis, "Optimal design of press-fitted filament wound composite flywheel rotors," *Composite Structures*, vol. 72, no. 1, pp. 47–57, 2006.

[63] H. E. Romeijn, Z. B. Zabinsky, D. L. Graesser, and S. Neogi, "New reflection generator for simulated annealing in mixed-integer/continuous global optimization," *Journal of Optimization Theory and Applications*, vol. 101, no. 2, pp. 403–427, 1999.

[64] O. Erdal and F. O. Sonmez, "Optimum design of composite laminates for maximum buckling load capacity using simulated annealing," *Composite Structures*, vol. 71, no. 1, pp. 45–52, 2005.

[65] K. Genovese, L. Lamberti, and C. Pappalettere, "Improved global-local simulated annealing formulation for solving non-smooth engineering optimization problems," *International Journal of Solids and Structures*, vol. 42, no. 1, pp. 203–237, 2005.

[66] M. Di Sciuva, M. Gherlone, and D. Lomario, "Multiconstrained optimization of laminated and sandwich plates using evolutionary algorithms and higher-order plate theories," *Composite Structures*, vol. 59, no. 1, pp. 149–154, 2003.

[67] A. Rama Mohan Rao and N. Arvind, "Optimal stacking sequence design of laminate composite structures using tabu embedded simulated annealing," *Structural Engineering and Mechanics*, vol. 25, no. 2, pp. 239–268, 2007.

[68] A. Corz, J. A. Gomez-Ruiz, J. I. Pelaez, E. Tenorio, and J. Veintimilla, "Design and optimization of symmetric laminated composites using a variable neighbourhood search-based

model," *Engineering Optimization*, vol. 44, no. 4, pp. 505–520, 2012.

[69] S. W. Tsai and E. M. Wu, "A general theory of strength for anisotropic materials," *Journal of Composite Materials*, vol. 5, pp. 58–80, 1971.

[70] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[71] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman and Hall/CRC, Boca Raton, Fla, USA, 5th edition, 2000.

[72] J. H. Zar, *Biostatistical Analysis*, Prentice Hall, 5th edition, 2009.

[73] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.

[74] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *The Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86–92, 1940.

[75] R. Iman and J. Davenport, "Approximations of the critical region of the Friedman statistic," *Communications in Statistics*, vol. 9, pp. 571–595, 1980.

[76] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[77] P. B. Nemenyi, *Distribution-Free Multiple Comparisons*, Princeton University, 1963.

[78] Ansys Inc., 2016. Ansys Release 17.0.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 1406292, 14 pages
https://doi.org/10.1155/2017/1406292

Research Article

A GODFIP Control Algorithm for an IRC Grain Dryer

Aini Dai,[1,2] Xiaoguang Zhou,[1] and Xiangdong Liu[3]

[1]*School of Automation, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*Science and Information College, Qingdao Agriculture University, Qingdao 266109, China*
[3]*College of Engineering, China Agricultural University, Beijing 100083, China*

Correspondence should be addressed to Xiaoguang Zhou; zxg_bupt@126.com

Received 25 March 2017; Revised 18 July 2017; Accepted 9 August 2017; Published 20 September 2017

Academic Editor: Domenico Quagliarella

Drying is an energy intensive and complex nonlinear process and it is difficult to control and make the traditional control meet the challenges. In order to effectively control the output grain moisture content of a combined infrared radiation and convection (IRC) grain dryer, taking into account the superiority of the fuzzy control method in dealing with complex systems, in this article, a genetic optimization dual fuzzy immune PID (Proportional-Integral-Derivative) (GODFIP) controller was proposed from the aspects of energy savings, stability, accuracy, and rapidity. The structure of the GODFIP controller consists of two fuzzy controllers, a PID controller, an immune algorithm, and a genetic optimization algorithm. In addition, a NARX model which can give relatively good predictive output information of the IRC dryer was established and used to represent the actual drying process to verify the control performance in the control simulation and anti-interference tests. The effectiveness of this controller was demonstrated by computer simulations, and the anti-interference performance comparative study with the other controllers further confirmed the superiority of the proposed grain drying controller which has the least value of performance objective function, the shortest rising time, and the best anti-interference ability compared to the other three compared controllers.

1. Introduction

Grain drying control is very important because it can decrease the grain loss by controlling the grain moisture and temperature to the desired level and maintain the quality, freshness, and longer life storage of grain. There are parameter uncertainties or variations in the grain dying process, and the establishment of a good control system for a grain dryer is difficult. Drying is an energy intensive process, one of the main control objectives is to dry the moisture content to desired level with efficient energy consumption and good quality, besides that stability of the system and robustness of the controller towards any disturbances are also the fundamental requirements of a dryer controller [1, 2].

In this research, the control object is a combined infrared radiation and convection (IRC) grain dryer designed by our research group which combines the direct and indirect heating drying technology. Infrared radiation drying is a new drying technology developed in recent 30 years and now has become more popular because of its advantages, such as its low drying time, the reasonable quality of the final dried product, and its greater energy savings capability, in addition to its lower price compared to microwave and vacuum drying methods [3]. So far, the research scope of the infrared radiation drying technology mainly focuses on the study of experiments and simulation of drying, but there are no studies in the aspect of infrared radiation grain drying control [4–11]. In order to meet the drying control requirements of the IRC dryer, effective control strategies for the grain dryer should be further researched on. Usually, in the whole grain drying process, the speed of the discharging grain motor is controlled (i.e., the adjustment of the drying time of grain in the dryer) to realize the drying target by detecting the error and its rate of change between the desired and the actual grain output moisture content according to the corresponding control algorithm.

Traditional control methods have encountered a lot of obstacles because grain drying is a complicated heat and mass transfer process which is characterized by long delay process, highly nonlinearity, multidisturbance, strong coupling, and

so on. It is difficult to make an accurate mathematical model of grain drying, so the control of grain drying is a challenging job [2, 12]. Liu and Bakker-Arkema have also reviewed some traditional control limitations, such as the controllers designed by Marchant, Whitfied, McFarlane, and Courtois et al. Indeed, the classical feedback control is necessary but is inadequate for controlling grain dryer [2]. The control effect of the feed-forward control is much better than that of the feedback control only, but the accuracy of the feed-forward control is affected by whether or not the system disturbance is measurable [13]. In addition, Proportional-Integral-Derivative (PID) controller is successfully applied in the classic automatic control and still used in the control of grain drying now, but it relies on the mathematical model and has some limitations of which the values of the control parameters (kp', ki', kd') are not changed in the whole control process, and when the controlled object and the environment are uncertain, the PID controller will be difficult to achieve satisfactory control effect and difficult to reach the control requirements with more strict restriction on overshoot. Model-predictive control (MPC) is a compound optimization algorithm which is based on model, rolling optimization, and feedback correction; it will control output changes by tracking the change of the set valve, so it is effective for the nonlinear and large lag process control [2]. In a series of Liu and Bakker-Arkema's papers, a model-predictive controller (MPC) was especially designed for grain drying. The simulation and field tests both showed that the controller performed well over a wide range of drying conditions. The distributed-parameter process model that the MPC employs is more comprehensive than the lumped parameter model and provides more detailed information on the process, but it requires more computation time and still relies on the accuracy of the system mathematical model [2].

In all, a drawback of the above-mentioned studies is that the authors generally made several simplifications in developing the dryer mathematical model based on some assumptions and observations. These simplifications are expected to affect the performance of these models and consequently their reliability in representing the real process when using these models for control purposes. Moreover, the mathematical models generally consist of sets of highly complex and nonlinear partial differential equations (PDES) with several auxiliary algebraic equations that involve transfer coefficients and thermophysical properties that require highly complicated numerical techniques to solve, rendering them undesirable options in control systems [14].

Since the 1970s, the research development of computer control technology and artificial intelligence has provided new ways for advanced control of the grain drying; thereafter drying control comes into the intelligent control period, of which the fuzzy logic controller (FLC) is a typical intelligent controller which imitates humans' decision-making and common sense [15, 16]. The FLC does not need to know the mathematical model of the controlled object and only needs to accumulate the experience data of a skilled human operator, which is the biggest difference from the conventional control. In essence, the FLC provides an algorithm which can convert the linguistic control strategy based on expert knowledge into an automatic control strategy. The FLC is not only widely used in industry but also a hot research topic in the control of grain drying now [17–25]. Combining FLC algorithm with traditional control algorithms, the problem of grain drying control can be effectively solved.

Fuzzy immune PID controller based on the immune feedback mechanism combines the intelligent FLC with the traditional PID controller and has the advantages of simple, good robustness and independence on the system model, which uses the characteristics of fuzzy control to learn the biological immune feedback mechanism under the complex disturbance and uncertain environment [26–29]. However, the algorithm also has some limitations, although the PID control parameters in the control process can be changed, the change rates of parameters are the same, affecting the control performance to a certain extent. Aiming at this limitation, in this paper, an improved fuzzy immune PID controller is designed to solve the limitation of the general fuzzy immune PID control algorithm.

In all, based on the idea of artificial intelligence, this paper proposes an improved fuzzy immune PID controller combined with two kinds of evolutionary algorithms: the immune feedback algorithm and the genetic optimization algorithm, which has improved the limitation of the traditional PID controller and the general fuzzy immune PID controller. Because the algorithm adopts two kinds of fuzzy controller and uses the genetic algorithm to optimize the initial controller parameters of the model, the proposed controller in this paper is called the genetic optimization dual fuzzy Immune PID (GODFIP) controller. Based on the GODFIP, the speed of discharging grain motor can be automatically adjusted to achieve the precise control of the output grain moisture of the IRC grain dryer according to the difference and its change rate between the output grain moisture content and the target moisture content. Finally, the NARX (Nonlinear Autoregressive models with Exogenous Inputs) model is used to represent the actual drying process to test the effectiveness of the proposed controller, and the comparative study with the other related controllers is also made, and the simulation results show that the control effect of GODFIP is better than that of other compared controllers.

2. The Mathematical Model of Grain Drying and Its Control Algorithm Structure

2.1. The Mechanical Structure of the New Grain Dryer. The IRC grain drying system has been put into use in Harbin Development Zone, Binxi town, China, Dongyu Machinery Co. Ltd. Fresh, mature corns were purchased from a local farm (an agricultural area in north of China).

The IRC grain dryer mechanism system is shown in Figure 1. It can be seen that the system mainly consists of the following parts: a wet grain barn, a grain dryer, and dried grain barn, and 3 elevators used to raise grains to the barns and 5 belt machines used to transport, of which the new radiation-convection grain dryer is rectangularity in shape and the overall dimensions are 4.75 m in height, 2.06 m in length, and 1.3 m in depth as shown in Figure 2.

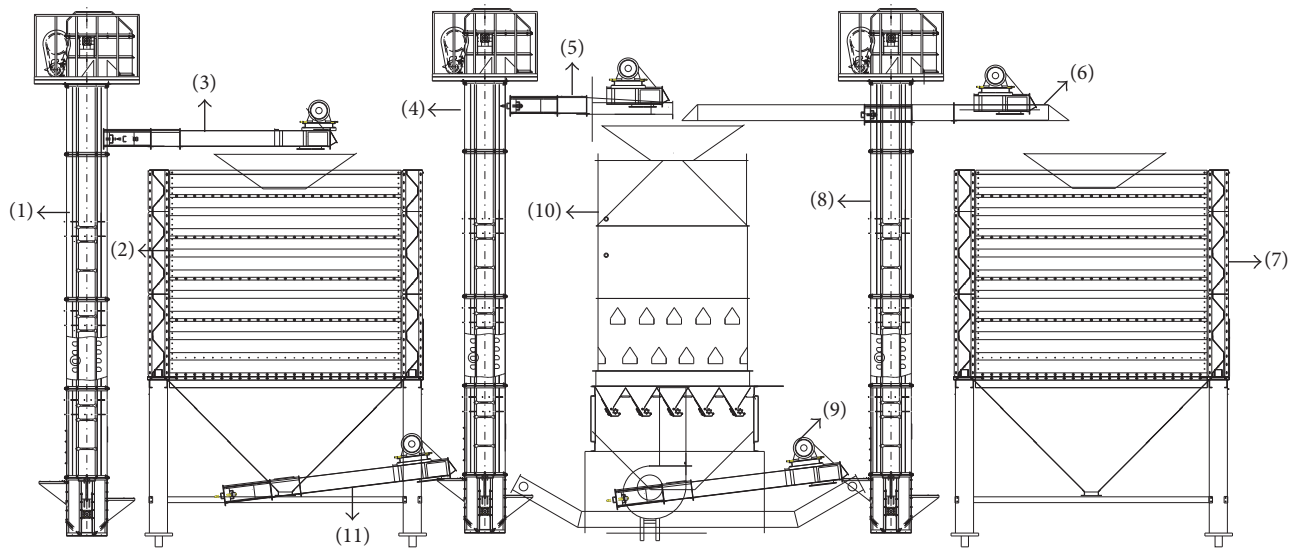


Figure 1: Mechanic structure diagram of the IRC grain dryer. (1) Bucket elevator T1, (2) wet grain barn, (3) belt conveyor P1, (4) bucket elevator T2, (5) belt conveyor P3, (6) belt conveyor P5, (7) dried grain barn, (8) bucket elevator T3, (9) belt conveyor P4, (10) dryer, and (11) belt conveyor P2.

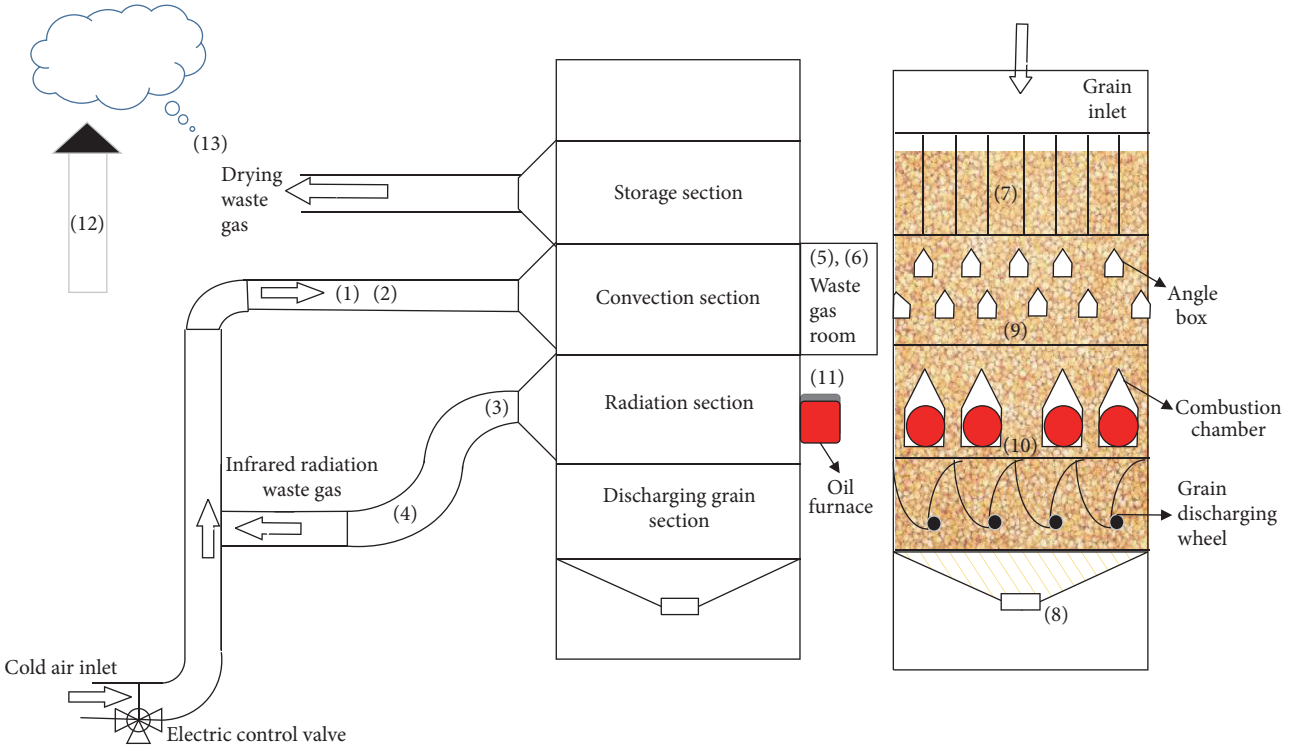


Figure 2: Scheme of the IRC grain dryer. (1) Main hot air speed. (2) Hot air temperature. (3) Infrared exhaust gas temperature. (4) Infrared exhaust gas velocity. (5) Exhaust gas temperature and humidity. (6) Drying waste gas. (7) Inlet grain temperature and moisture. (8) Outlet grain temperature and moisture. (9) Postdrying grain temperature. (10) Infrared grain temperature. (11) Combustion tube temperature. (12) Flue-gas temperature. (13) Ambient temperature and humidity.

Figure 2 is the schematic diagram of the IRC grain dryer which consists of four sections: the storage grain section (0.8 m in height), the convection section (1.1 m in height), the radiation section (0.8 m in height), and the discharging grain section (1.2 m in height), of which the convection section is a combination design and there are three kinds of drying process to choose: counter flow drying; concurrent and counter flow drying; mix flow drying, and the combustion

Figure 3: Controlling situation and equipment for the IRC grain dryer.

furnace of radiation section is a side heat radiation type equipped with some corresponding safety devices and can be automatically ignited. Capacitive moisture sensor is installed in the inlet and outlet of the grain dryer for the need of control and some other online sensors detecting the real drying parameters are also installed as shown in Figure 2 (numbers (1)–(13)).

2.2. The Control System of the IRC Dryer. The controlling situation and some equipment for the grain dryer are shown in Figure 3. Programmable controller technology (PLC) + frequency converter are adopted in the control system and the drying parameters can be real-time detected, such as the air temperature and humidity, the grain moisture, and temperature. The grain discharging speed is controlled by the frequency converter (VFD-007M43B) installed in the control cabinet which worked in two operating modes (manual mode or automatic mode); when working in manual mode, long-term work experiences are generally needed to manually adjust the discharging grain speed to achieve the drying target. In automatic mode, a reliable control algorithm is also needed to automatically control the dryer.

The control system of the IRC dryer is also equipped with a computer, which is connected with PLC (S7-300) through Ethernet. The experimental data can be analyzed and processed in the computer, and different drying algorithms can be designed and tested.

2.3. The Process of Radiation-Convection Grain Drying. As seen from Figures 1 and 2, the drying process is as follows. (1) First, raise the wet grain into the dryer from the wet grain barn by the bucket elevator T2 and belt conveyors P2 and P3; when the upper limit grain level sensor is installed in the storage section of the grain dryer alarms, the wet grain feeding is stopped and ready for drying. (2) Secondly, let oil furnace heat the radiator and begin to carry out radiation drying in the radiation section by use of the high temperature of the radiator. (3) Reuse the radiation exhaust gas of infrared radiation section and mix it with appropriate amount of cold air to achieve the regulation of the mixed air temperature by controlling the opening of solenoid valve, and blow the mixed air into the convection section through pipe to carry out the convection drying in the convection section based on the heat and mass exchange principle. (4) Finally, start the discharging grain wheel and other relevant devices, then discharge the dried grain into the dried grain barn by the bucket elevator T3 and belt conveyor P5, and end the drying.

In the whole drying, the speed of discharging grain motor can be adjusted automatically by an intelligent control algorithm every time interval or adjusted manually by an experience worker according to the detected drying parameters.

2.4. The Identification Equation of Drying Process. For the complex IRC grain dryer, the description of the dynamic process of grain drying is more difficult. It is an effective way to learn the characteristics of the drying process by modeling the input and output data [30].

2.4.1. Autoregressive Exogenous (ARX). Autoregressive Exogenous (ARX) model has been widely applied in the prediction control. It does not need to know the physical mechanism inside the complex process, so it is regarded as a "black box" model. It provides a fast and efficient solution to the actual system output by means of a least squares approach, and it has the advantages of simple structure and strong robustness. It is an autoregressive model which has exogenous inputs, and it relates the current output value of a time series to past output values of the same series and current and past values of the driving (exogenous) series. More details about ARX can be found in [31, 32]. The basic structure of ARX model identification is shown in

$$\begin{aligned} A(z)\, y(k) &= B(z)\, u(k) + e(k), \\ A(z) &= 1 + \sum_{i=1}^{n_a} a_1 z^{-i}, \\ B(z) &= z^{-t_d} \sum_{j=0}^{n_b - 1} b_j z^{-j}, \end{aligned} \tag{1}$$

where the ARX order is determined as $[n_a \;\; n_b \;\; t_d]$, n_a is the model order of $A(Z)$, n_b is the model order of $B(Z)$, and t_d is the estimated pure time delay between the exogenous input signal $u(k)$ and the output signal $y(k)$; $e(k)$ is a white noise term, generally assumed to be Gaussian and White; k represents the discrete time step.

The input and output data for identification model of grain drying are from the drying experiment of the IRC grain dryer (corn mixed flow and radiation) in December 4, 2015, a total of 384 sets of data, and the sampling frequency is 60 HZ. The input data of the identified drying model is the current and past drying time of grain being experienced in the dryer and the past output grain moisture content of the

grain dryer; the output data of the model is the current output grain moisture of the grain dryer real detected by the output grain moisture sensor which has been calibrated using the 105°C standard oven method (GB 5497-1985).

The corn of the drying experiment purchased from the local farmers is a natural harvest species: number 1 XingXing (a breed name of corn), and its initial grain moisture content is about 26%. The ambient temperature is about minus 10°C, relative humidity 60–70%. The hot air temperature is between 80 and 120°C, and the hot wind speed is 12 m/s.

By using the identification toolbox in Matlab, ident, the identified transfer function of the model is a two-order lag system as shown in (2), of which the identification model order is [2 2 1] (the input and output order are, resp., equal to 2 and the time delay t_d is equal to 1), and $A(Z)$ and $B(Z)$ are as shown in (3) and (4).

$$G(z) = \frac{B(Z)}{A(Z)} = \frac{0.3799z^{-1} - 0.3804z^{-2}}{1 - 0.7471z^{-1} - 0.2277z^{-2}} \times Z^{-1}, \quad (2)$$

$$A(z) = 1 - 0.7471z^{-1} - 0.2277z^{-2}, \quad (3)$$

$$B(z) = 0.3799z^{-1} - 0.3804z^{-2}. \quad (4)$$

In this study, we use mean squared error (MSE) and squared correlation coefficient (R) to evaluate the prediction performance of the designed drying model shown in

$$\begin{aligned} &\mathrm{MSE} = \frac{1}{m}\sum_{i=1}^{m}\left(y_i - \widehat{y}_i\right)^2, \\ &R \\ &= \sqrt{\frac{\left(m\sum_{i=1}^{m}\widehat{y}_i y_i - \sum_{i=1}^{m}\widehat{y}_i\sum_{i=1}^{m}y_i\right)^2}{\left(m\sum_{i=1}^{m}\widehat{y}_i^2 - \left(\sum_{i=1}^{m}\widehat{y}_i\right)^2\right)\left(m\sum_{i=1}^{m}y_i^2 - \left(\sum_{i=1}^{m}y_i\right)^2\right)}} \\ &\in [0 \quad 1], \end{aligned} \quad (5)$$

where m is the actual total number of the dataset and y_i and $\widehat{y}_i$ are actual and predicted values, respectively; the closer MSE is to zero, the better the prediction performance of model is, and the closer R (range from 0 to 1) is to 1, the better the model fits.

Experimental model identification results are shown in Figure 4. The squared correlation coefficient R between the model and the measured data is equal to 95.5%, and the MSE error is equal to $2.29 * 10^{-4}$, showing that the resulting linear model has a good approximation of the actual drying process.

2.4.2. Nonlinear Autoregressive with Exogenous Input (NARX) Model. The theory of linear systems identification is a relatively matured field [33]; however, as seen from Figure 4, the ARX identified linear models can not adequately capture all the magnitudes of the real output response due to the high degree of nonlinearity of the system, so in this paper a nonlinear model known as the Nonlinear Autoregressive with Exogenous input (NARX) model is developed.

The NARX model has been proven to have a superior performance and has been successfully employed in solving various types of complex nonlinear model problems in recent years [34–38]. The NARX model uses a nonlinear function to predict the output grain moisture content of drying by regressing from the past values of system output y and the past values of exogenous input u. The form of the NARX model is shown in

$$\begin{aligned} \widehat{y}(k) = f\big[&y(k-1), y(k-2), \ldots, y(k-n_a), x(k-t_d), \\ &x(k-t_d-1), \ldots, x(k-t_d-n_b+1)\big], \end{aligned} \quad (6)$$

where $f[\bullet]$ is a nonlinear mapping function that estimates the output $\widehat{y}(k)$ at time sample k. In this study the artificial neural network is used to regress the nonlinear function $f[\bullet]$. The NARX network model structure consists of an input layer, a hidden layer, and an output layer of which the number of hidden neurons is 10 and the time lags order for the input and output series is 2 (i.e., $n_a = n_b = 2$). The architecture of the NARX model adopted in this study to predict the output moisture content is shown in Figure 5, where $x(t)$ is the system input (drying time); $y(t)$ is the system output (output grain moisture content); w is the weights value; b is the bias value.

In the case of modeling, the input-output data of 384 samples are randomly divided into three parts to develop a NARX model: 268 training datasets, 58 validation datasets, and 58 testing datasets. During the training phase, the past input and output data of training are presented to train and adjust the neural network by using the Levenberg-Marquardt algorithm; during validation, the validation data are used to measure network generalization and to halt training when generalization stops improving; during testing, the testing data have no effect on training data, so an independent measure of network performance can be provided during and after training.

Table 1 and Figure 6 show the regression prediction results of the NARX model, and it shows that the NARX model has high prediction accuracy of which the MSE and R on testing data are equal to $1.062 * 10^{-5}$ and 99.78%, respectively, and the predicted errors (the difference values between the predicted

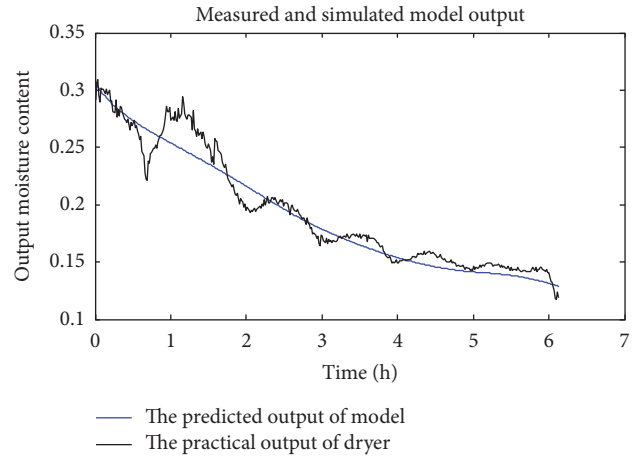


Figure 4: Fitting effect curve of the ARX model with the actual curve.

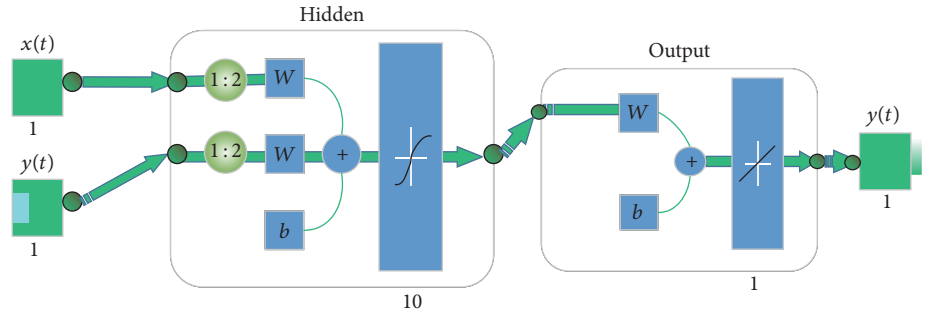


Figure 5: The architecture of the NARX model.

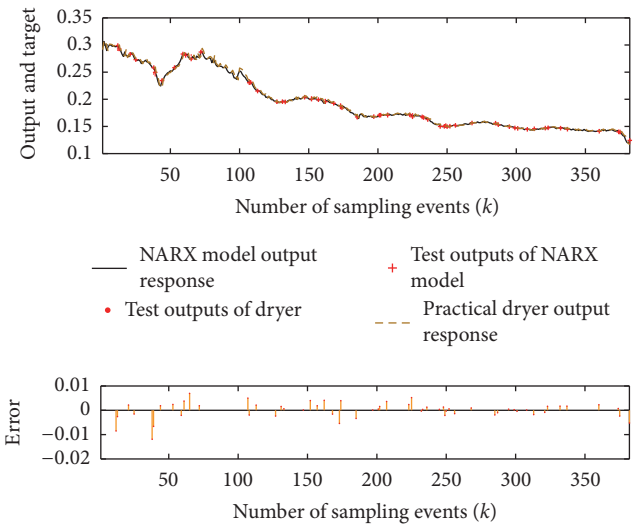


Figure 6: The prediction results of the NARX model on the testing data.

Table 1: The simulation prediction results of the NARX model.

Data	Size	Model prediction results	
		MSE	R
Training data	268	$1.70 * 10^{-5}$	99.68%
Validation data	58	$1.08 * 10^{-5}$	99.78%
Testing data	58	$1.06 * 10^{-5}$	99.78%

output values of model and the practical output values of dryer) are basically within in ±1%. These results indicate that the developed NARX model is considered as a much better representative for the nonlinear grain drying process than the ARX model (MSE: $2.29 * 10^{-4}$; R: 95.5%).

2.5. The Control Algorithm Structure of Grain Dryer. There are many factors that affect the control performance of grain drying as shown in Figure 7, the main factors of which are the grain initial temperature and inlet moisture, the ambient environment temperature and moisture, the hot air and cooling air flow, the hot air temperature, and the speed of discharging grain. In addition, the corn used in the experiment is purchased from different local farmers and each batch of grain moisture is different, so it will easily lead to the variations of the grain moisture content in the dryer. Furthermore, the ambient temperature is below zero, the ice inside the grain will affect the detection accuracy of the grain moisture sensor. Therefore, grain drying is such a complex process that satisfactory control effect is difficult to achieve.

Under a certain period of time and environmental conditions, some variables can be thought to be unchanged for a certain batch of grain drying, such as the grain initial temperature and initial moisture content, the ambient environment temperature and moisture, the hot air temperature, humidity, and the hot air flow rate. Usually, in the engineering practice of grain drying, the drying time is often taken as the control variable and the output grain moisture content as the important controlled variable; the other affection factors are taken as the disturbance signals.

The designed control scheme of this paper is shown in Figure 8, where the intelligent controller in this paper refers to the designed dual fuzzy immune PID controller, and the immune algorithm refers to the feedback control law, $e(t)$ is the error between the output and the target value, de/dt is the change rate of $e(t)$, and the interference signal can be the inlet grain moisture, the ambient temperature, and so on. The drying time of grain in the dryer can be tuned by controlling the speed of the discharging grain motor in the whole drying process according to the corresponding intelligent control algorithm to achieve the target value; in addition, the controller's initial parameters are optimized by a genetic algorithm.

3. Design of the Genetic Optimization Dual Fuzzy Immune PID (GODFIP) Controller

3.1. The General Fuzzy Immune PID Controller Design

3.1.1. Biological Immune Principle. Biological immune system can produce antibodies against a foreign invasion of the antigen, which plays the defense role. The most important cells in the immune system are the lymphocytes which are mainly two kinds: B and T cells; B cells are responsible for antibody production and carry out the immunity function and T cells regulate the whole immune process. T cells are composed of inhibit T cells (T_S) and helper cells (T_H), which, respectively, inhibit and help B cells respond to a stimulus. When the antigen is increasing, there are more T_H cells and

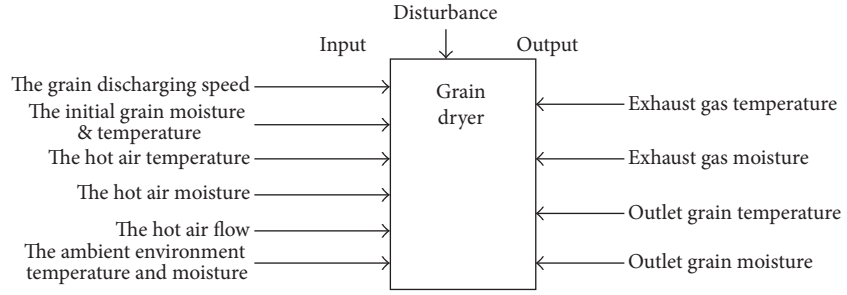


Figure 7: Factors affecting grain drying.

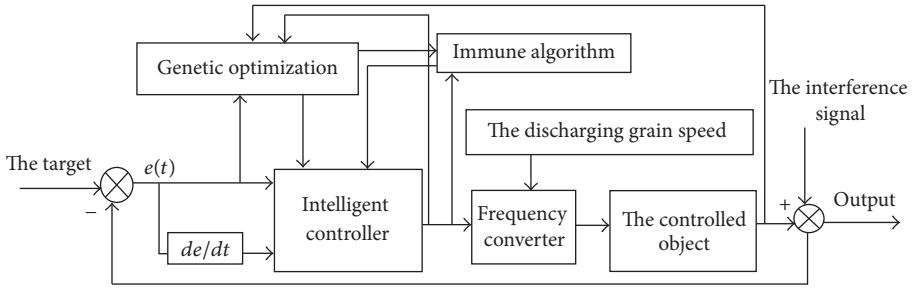


Figure 8: Block diagram of the control model scheme for the IRC grain dryer.

less T_S cells in the body which will produce more B cells, that is, more antibodies; when the antigen is gradually reduced, T_S cells will increase and inhibit the generation of T_H cells; thereby reducing the B cells, antibodies decreased. When the antigen is eliminated, the immune response is completed.

Assuming that the k generation amount of antigen is $\varepsilon(k)$, the change rate of the number of antigens is $\Delta\varepsilon(k)$; the output of T_H cells is $T_H(k)$; the output of T_S cell is $T_S(k)$; according to the immune feedback mechanism, the total stimulation received by B cells is $S(k)$, as shown in

$$S(k) = T_H(k) - T_S(k). \tag{7}$$

Among them $T_H(k) = k_1\varepsilon(k)$; $T_S(k) = k_2 f[\varepsilon(k), \Delta\varepsilon(k)]\varepsilon(k)$; $f(\cdot)$ is a nonlinear control function that represents the ability to suppress external stimuli. Its feedback control law is shown in (8) and thus is also called the nonlinear p-type controller.

$$\begin{aligned} S(k) &= T_H(k) - T_S(k) \\ &= \{k_1 - k_2 f[\varepsilon(k), \Delta\varepsilon(k)]\varepsilon(k)\} \\ &= k\{1 - \eta f[\varepsilon(k), \Delta\varepsilon(k)]\}\varepsilon(k) = K_p\varepsilon(k), \\ K_p &= k\{1 - \eta f[\varepsilon(k), \Delta\varepsilon(k)]\}. \end{aligned} \tag{8}$$

Among them, $k = k_1$ is the speed of control response, $\eta = k_1/k_2$ is related to the stability of the response, and K_p will change with the amount of the antigen. Moreover, a reasonable adjustment of k and η is also crucial, and it will enable the control system to have smaller overshoot and faster response rate.

3.1.2. Design of General Fuzzy Immune PID Feedback Controller. Imitating the above immune feedback mechanism, a p-type fuzzy immune PID feedback controller is designed, which combines the fuzzy immune controller with the general PID controller.

The discrete form of the ordinary PID controller is as shown in

$$u'(k) = \left(kp' + \frac{ki'}{z-1} + kd'\frac{z-1}{z}\right)e(k). \tag{9}$$

The discrete output of the fuzzy immune PID is shown in (10) by replacing the antigen $\varepsilon(k)$ and its change rate $\Delta\varepsilon(k)$ of the fuzzy immune system with the output of general PID controller $u'(k)$ and its change rate $\Delta u'(k)$, respectively.

$$\begin{aligned} u(k) &= K_p u'(k) = k\{1 - \eta f[u'(k), \Delta u'(k)]\} \\ &\cdot \left(kp' + \frac{ki'}{z-1} + kd'\frac{z-1}{z}\right)e(k) \\ &= \left(kp + \frac{ki}{z-1} + kd\frac{z-1}{z}\right)e(k), \end{aligned} \tag{10}$$

where

$$\begin{aligned} kp &= kkp'\{1 - \eta f(\bullet)\} = K_p kp', \\ ki &= kki'\{1 - \eta f(\bullet)\} = K_p ki', \\ kd &= kkd'\{1 - \eta f(\bullet)\} = K_p kd'. \end{aligned} \tag{11}$$

Its controller structure is shown in Figure 9, and the controlled object is the IRC dryer; the controller variable is

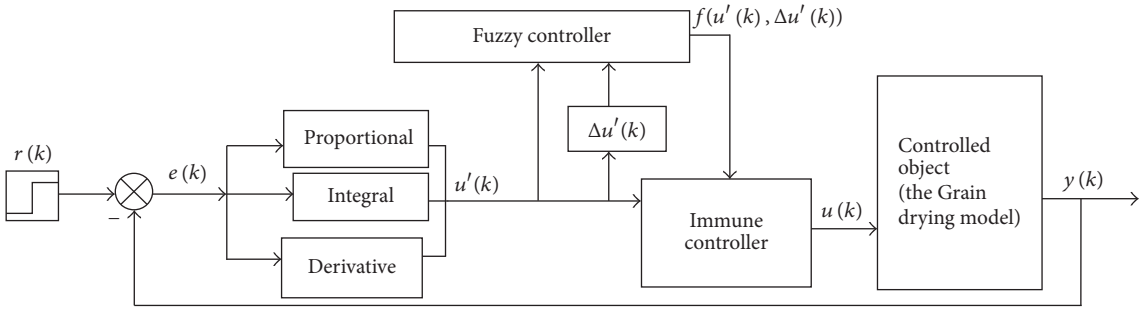


Figure 9: Structure of the fuzzy immune PID feedback controller.

the output of the model: $y(k)$ is the outlet grain moisture; $r(k)$ is the input of the control model which is a step response function (the target moisture content value); $e(k)$ is the error between the output and the target value; de/dt is the change rate of e; $u'(k)$ is the output of the general PID controller; $u(k)$ is the immune controller's output which can control the drying time of grain in the dryer (i.e., $u(k)$ can be used to control the speed of discharging grain motor to obtain the tuning of the grain drying time).

The Mamdani fuzzy controller designed in the control algorithm is used to approximate the nonlinear function ($f[u'(k), \Delta u'(k)]$), and it has two inputs and one output. The two input variables of the fuzzy controller are the output of the PID algorithm ($u'(k)$) and its change rate of error ($\Delta u'(k)$), and the output variable of the fuzzy controller is a nonlinear function ($f[u'(k), \Delta u'(k)]$). As seen in (11), the parameters of the controllers kp, ki, and kd can be adjusted adaptively with the change of $u'(k)$ and $\Delta u'(k)$, which overcomes the limitations that the general PID controller's parameters can not be dynamically adjusted during the control process.

Firstly, the fuzzy method of the input variables should be used to transform from the basic domain to the corresponding fuzzy set domain and define the quantification factor of input variables ($K_{u'}$; $K_{\Delta u'}$) to get the fuzzy input ($U' = K_{u'} * u'$; $\Delta U' = K_{\Delta u'} * \Delta u'$), so the output needs to transform from the fuzzy set to the basic domain. In this paper, the fuzzy set domain of the input and output is $[-1, 1]$.

Secondly, the membership functions of input and output lingual variables should be formed to determine the distribution of different variables. There are commonly three types of membership function to be used: (1) normal distribution; (2) triangle; (3) trapezoidal. The numbers of the fuzzy sets for the input variable and output variable are used to meet the requirements of accuracy. As the numbers of the fuzzy sets increase, the numbers of fuzzy control rules increase accordingly, which will improve the accuracy of control, but meantime the complexity of control is increased; on the premise of satisfying the requirement of control precision, the least numbers of the fuzzy sets can be equal to 3 based on the principle of determining the minimum inference rules numbers; in addition, the numbers of the fuzzy sets for the input and the output can be unequal; in order to improve the accuracy of control, the fuzzy sets numbers for the output variable can be increased [39]. Based on the above analysis, in this paper each of the input variables has three fuzzy sets: {positive, zero, negative} or {P, Z, N}; the output adopts five fuzzy sets: {positive big, positive small, zero, negative small, negative big} or {PB, PS, Z, NS, NB}. The Mamdani fuzzy controller can use the following 9 knowledge rules shown in the Table 2 (e.g., if $u'(k)$ is P and $\Delta u'(k)$ is P then $f[u'(k), \Delta u'(k)]$ is NB), which imitating the control principle of biological immune feedback. The input and output membership functions of the fuzzy controller are shown in Figures 10(a)–10(c).

Table 2: Knowledge rules of the fuzzy controller.

$f[u'(k), \Delta u'(k)]$		$\Delta u'(k)$		
		P	Z	N
$u'(k)$	P	NB	NS	PS
	Z	NS	Z	PS
	N	NS	PS	PB

And then, the fuzzy output can be obtained by the fuzzy inference synthesis algorithm according to the rules of Table 2.

Finally, according to the fuzzy rules, defuzzification will transform the output $f(\bullet)$ of the fuzzy controller from a fuzzy set to a crisp number by using the gravity method (centroid) and the Zadeh fuzzy logic operation (AND). The relationship surface figure of the output and input in the universe of discourse is shown in Figure 10(d).

3.2. Genetic Optimization Dual Fuzzy Immune PID (GODFIP) Controller

3.2.1. The Structure of GODFIP. It can also be seen from (11) that the PID parameters (kp, ki, kd) have the same change rate (K_p) which should be different in the actual control process, thus affecting the control effect to some extent. In this paper, by adding a fuzzy PID parameter adjusting controller on the general fuzzy immune PID controller to adjust the increment values of the PID parameters in (11), making kp', ki', kd' variable in the process of control shown in (12), the change

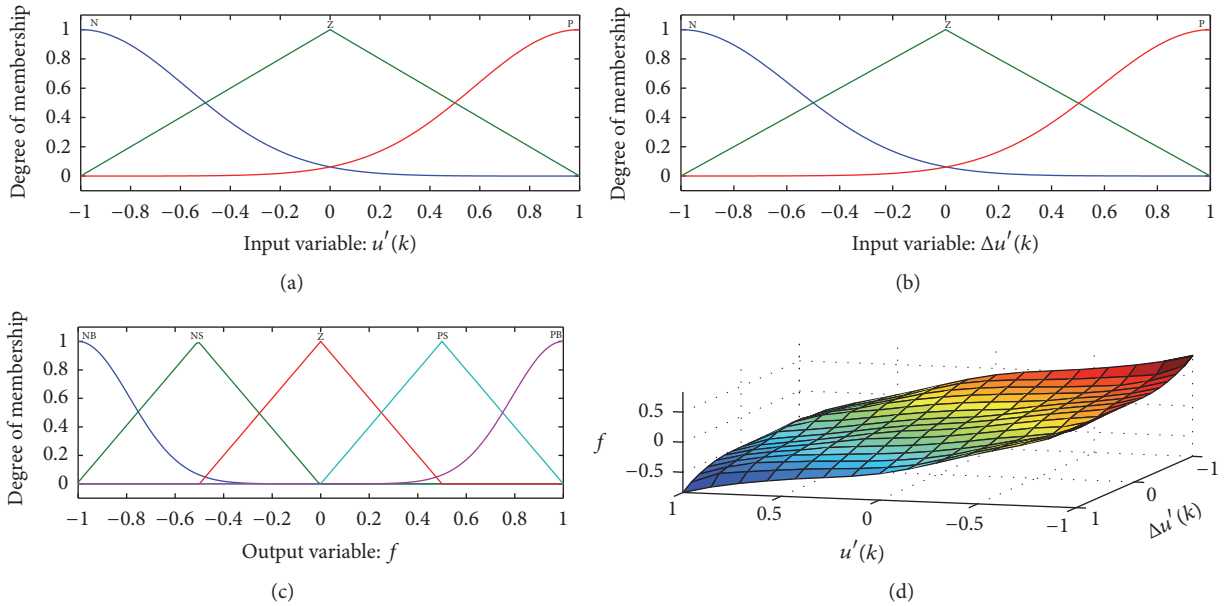


Figure 10: Degree of membership function plots (a) $u'(k)$, (b) $\Delta u'(k)$, (c) f, and (d) relationship surface figure of the output and input.

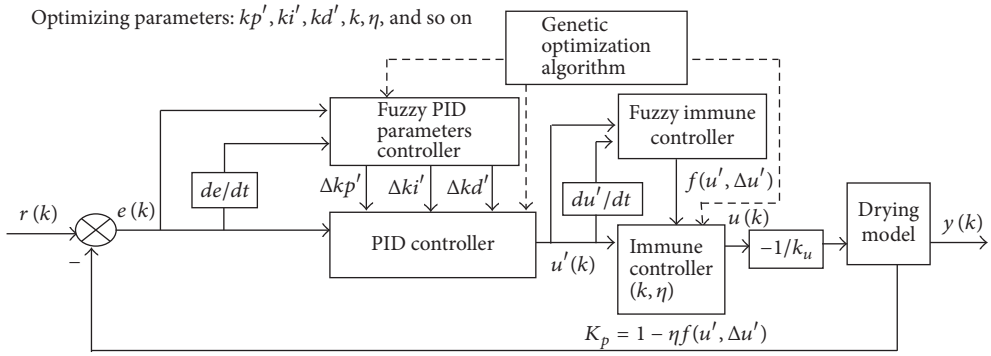


Figure 11: Structure of the genetic optimization dual fuzzy immune PID (GODFIP) controller.

rates of the PID parameters (kp, ki, kd) in (11) are not the same.

$$kp'(k) = kp' + \Delta kp'(k),$$
$$ki'(k) = ki' + \Delta ki'(k), \quad (12)$$
$$kd'(k) = kd' + \Delta kd'(k).$$

In addition, in the process of control, there is some difficulty in selecting the parameters (kp', ki', kd', k, η), which is often time consuming and affects the control effect to some extent. Moreover, a reasonable adjustment of k and η is also crucial, and it will enable the control system to have smaller overshoot and faster response. So genetic algorithm is adopted to optimize the parameters of the GODFIP controller, and the optimal control is realized.

Therefore, in this paper, a genetic optimization dual fuzzy immune PID (GODFIP) controller is designed, which not only can improve the limitation of the general fuzzy immune PID control but also can find the optimal parameters (kp', ki', kd', k, and η) according to the control situation. The structure of GODFIP controller is as shown in Figure 11.

In Figure 11, $r(k)$ is the reference target input; $e(k)$ is the error between the output and the target value; de/dt is the change rate of e; $u'(k)$ is the output of the PID controller after parameter adjustment (i.e., the input of fuzzy immune controller); $f(\bullet)$ is the nonlinear function; $u(k)$ is the immune controller's output which can control the drying time; the drying model is the controlled object; $y(k)$ is the output of model (i.e., the output grain moisture).

Taking the output grain moisture $y(k)$ as the controlled variable and the output of the fuzzy immune controller $u(k)$ as the control variable, the drying time of grain $T_{\text{dry}}(k)$ in the dryer can be adjusted to realize the control target.

In the drying experiment, by the experiment of calculating the grain weight of being discharged from the dryer within an hour (20 HZ: 3.126 t/h; 10 HZ: 1.753 t/h), the linear inverse relationship between the speed of discharging motor and the drying time is obtained ($(u = -k_u * T_{\text{dry}}(k) + c)$).

Table 3: PID parameters tuning rules of fuzzy PID parameters controller.

		de/dt																			
		$\Delta kp'$							$\Delta ki'$							$\Delta kd'$					
		NB	NM	NS	ZE	PS	PM	PB	NB	NM	NS	ZE	PS	PM	PB	NB	NM	NS	ZE	PS	PM	PB
e	NB	PB	PB	PM	PM	PS	ZE	ZE	NB	NB	NM	NM	NS	ZE	ZE	PS	NS	NB	NB	NB	NM	PS
	NM	PB	PB	PM	PS	PS	ZE	ZE	NB	NB	NM	NS	NS	ZE	ZE	PS	NS	NB	NB	NB	NM	PS
	NS	PM	PM	PM	PM	ZE	NS	NS	NB	NM	NS	NS	ZE	PS	PS	ZE	NS	NM	NM	NS	NS	ZE
	ZE	PM	PM	PS	ZE	NS	NM	NM	NM	NM	NS	ZE	PS	PM	PM	ZE	NS	NS	NS	NS	NS	ZE
	PS	PS	PS	ZE	NS	NS	NM	NM	NM	NS	ZE	PS	PS	PM	PB	ZE	ZE	ZE	ZE	ZE	ZE	ZE
	PM	PS	ZE	NS	NM	NM	NM	NB	ZE	ZE	PS	PS	PM	PB	PB	PB	NS	PS	PS	PS	PS	PB
	PB	ZE	ZE	NM	NM	NM	NB	NB	ZE	ZE	PS	PM	PM	PB	PB	PB	PM	PM	PM	PS	PS	PB

A brief introduction to the design principle of fuzzy PID parameter controller and genetic algorithm of the control structure of GODFIP are as follows (Sections 3.2.2 and 3.2.3).

3.2.2. Fuzzy PID Parameter Controller. In Figure 11, there are two inputs for the fuzzy PID parameter controller, e and de/dt; three outputs, $\Delta kp'$, $\Delta ki'$, and $\Delta kd'$ (i.e., three PID parameters increment values); the universe of discourse of the input and output is $[-3, 3]$; $\Delta kp'$, $\Delta ki'$, and $\Delta kd'$ can be deduced from the fuzzy rules shown in Table 3. The fuzzy PID parameter controller can change the increment values of the PID parameters dynamically according to the fuzzy rules by tracking the error signal and its change rate.

3.2.3. Genetic Optimization Algorithm. Genetic algorithm is a stochastic global optimization method that mimics the metaphor of the natural biological evolution. According to the fitness function value, the global optimal solution can be obtained by the genetic evolution. The fitness value of each individual in the population is calculated by the fitness function and provided to the operator for selection, crossover, and mutation and screening individuals to find the best by retaining the best fitness value and eliminating the poor fitness values. If the termination condition is satisfied, then the optimal individual is used to be assigned to the parameters of the controller; otherwise continue to calculate the new species until the global optimal value is found.

So we can use genetic algorithm to optimize the control parameters (kp', ki', kd', k, η). The basic genetic processes are selection, crossover, and mutation. The process of genetic algorithm is as follows:

(1) Parameter coding: real code is adopted, and for a given parameter range [min, max], the real number coding is equal to min + (max − min) ∗ rand.

(2) Population initialization: the individual coding length is 8 (kp, ki, kd, k, η, and 3 proportional coefficients used in the control structure), the population size is 20, the generations of evolution are 100, and the termination error is $1e-6$. In order to avoid the blindness of searching for the best, the initial values of parameters are firstly obtained by the first run of the genetic algorithm and then take this as the center to both sides to find the best.

(3) Determining fitness function: the minimum objective function of parameter selection is obtained from the aspects of reducing energy loss, stability, accuracy, and rapidity and to be provided for the operator selection and judgment. To prevent the control input of the controlled object too large and save the energy consumption, the output of the fuzzy immune controller is also added to the objective function, and the optimal function of the controller is shown in

$$J = \int_0^\infty \left[w_1 |e(t)| + w_2 u(t)^2\right] dt + w_3 t_r, \tag{13}$$

where $e(t)$ is the system error; $u(t)$ is the output of the fuzzy immune controller; t_r is the rising time; w_1, w_2, and w_3 are the weights.

In order to avoid overshoot, the penalty function is adopted as shown in (14) ($w_4 \gg w_1$), $\text{err}\, y(t) = y(t) - y(t-1)$, where $y(t)$ is the output of the controlled object; δ is the size of the overshoot to be controlled; once the overshoot is generated, it will be used as the optimal indicator.

$$\begin{aligned} &\text{if} \quad |e(t)| > \delta \\ &\text{then} \quad J \\ &\qquad = \int_0^\infty \left[w_1 |e(t)| + w_2 u^2(t) + w_4 \left|\text{err}\, y(t)\right|\right] dt \\ &\qquad\quad + w_3 t_r. \end{aligned} \tag{14}$$

4. Simulation and Discussion

4.1. Control Simulation Based on NARX Model. In order to verify the effectiveness of the proposed GODFIP controller in the grain drying control, the following controllers, the general PID controller, the fuzzy PID controller, the fuzzy immune PID controller, and the GODFIP controller, are, respectively, designed and simulated to be compared with the GODFIP controller by programming in the Matlab. As can be seen from Section 2.5, the NARX drying model has a better modeling accuracy and performs much better than the ARX drying model, so it can be considered to be a reliable representative of the IRC dryer. In this control simulation, the NARX model is selected to represent the actual drying process to verify the control performance.

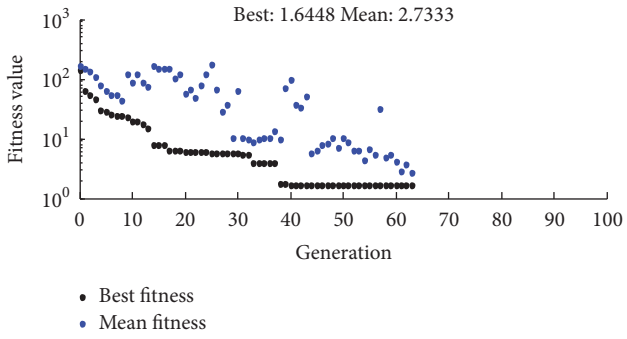


FIGURE 12: The fitness curve of optimization simulation for the GODFIP controller.

During the simulation, the controlled variable is the output grain moisture content $y(k)$ of which the initial value is set to 26%; the control variable is the speed of the discharging grain motor; the input of each model is the step response function (final value = 15%) mimicking the target grain moisture; the step response curves of different controllers are compared within 100 sampling events.

The optimal solutions of control parameters of the GODFIP controller are evaluated by a real-coded genetic algorithm. The fitness curve of the optimization simulation is shown in Figure 12, and the best objective function value J is equal to 1.6448 after 63 generations, and the optimal PID parameters are, respectively, as follows:

(1) $kp' = 2.601$; (2) $ki' = 1.551$; (3) $kd' = 1.137$; (4) $k = 2.53$; (5) $\eta = 8.328$; (6)–(8) other three proportional coefficients: −8.426, 5.062, 1.939.

In order to achieve a fair comparison, the genetic optimization algorithm is also used to optimize the other three controllers. Moreover, three runs of the genetic optimization algorithm program are made to avoid the stochastic error for each controller; finally, the average value of the objective function value J is considered as the performance measure of each controller.

Figure 13 compares the performances of different controllers in controlling the NARX drying model and the control simulation results are as shown in Table 4, where the definition of t_r, $\delta\%$, $y(t_p)$, $y(\infty)$, t_{s1}, and t_{s2} is as follows.

The rising time to the target value t_r is the time to reach 95% of the steady state for the first time in the transient process.

Maximum overshoot $\delta\%$ is as shown in

$$\delta\% = \frac{\left(y\left(t_p\right) - y\left(\infty\right)\right)}{y\left(\infty\right)}, \tag{15}$$

where $y(t_p)$ is the first peak value of the system response and $y(\infty)$ is the steady value of the system response.

Adjusting time t_{s1} is the time required from the first peak value $y(t_p)$ to the steady value $y(\infty)$ that falls between the deviations allowed (±5%) and is maintained within the allowable range.

The adjusting time t_{s2} is the time to the steady value after the interference disappears.

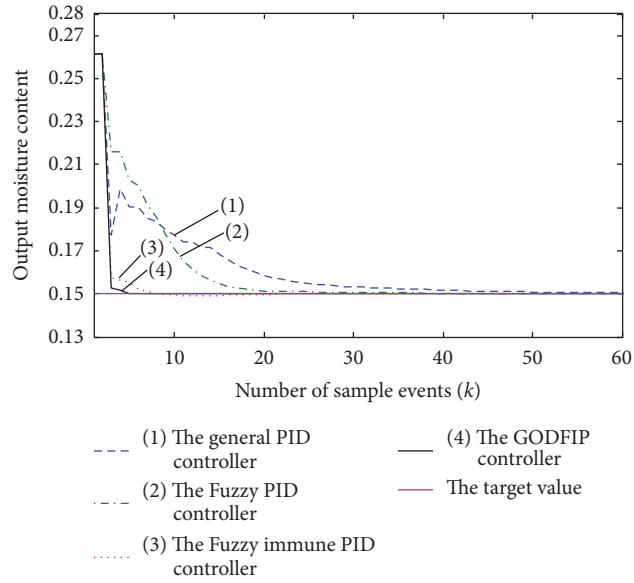


FIGURE 13: Simulation results comparison of different controllers for the grain drying process of IRC dryer.

4.2. The Anti-Interference Test Simulation Based on NARX Model. The dynamic influence factors of the grain drying process are a lot, and the influence of various disturbance factors in the control process easily leads to the variations of the output grain moisture. In order to verify the anti-interference performances of the controllers, the interference signal with the amplitude of 0.02 at sampling number 105 is added to represent a possible increase in the initial moisture of grain that enters the IRC dryer which is shown in Figure 14(a), and this anti-interference simulation test is within 200 sampling events. The anti-interference performance comparison results are shown in Figure 14(b).

4.3. Simulation Result Discussion

(1) The Optimization Results. The genetic optimization algorithm is used to achieve the optimal control of the GODFIP controller, the fuzzy immune PID controller, the fuzzy PID controller, and the general PID controller based on the performance objective function from the aspects of energy savings, stability, accuracy, and rapidity. As can be seen from the comparison results shown in Table 4, the J value of the GODFIP is equal to 1.6450 and is obviously less than the compared controllers which is about 3.53% decrease compared to the fuzzy immune PID controller, about 91.9% decrease compared to the fuzzy PID controller, and about 93.68% decrease compared to the general PID controller. Hence, the GODFIP controller has achieved the best control performances in terms of accomplishing the least value of J compared to the other three controllers, showing the effectiveness of the GODFIP controller.

(2) The Overshoot, the Adjusting Time t_{s1}, and the Rising Time t_r. Overshoot, adjusting time, and rising time are important indexes to judge a controller's performance reflecting the

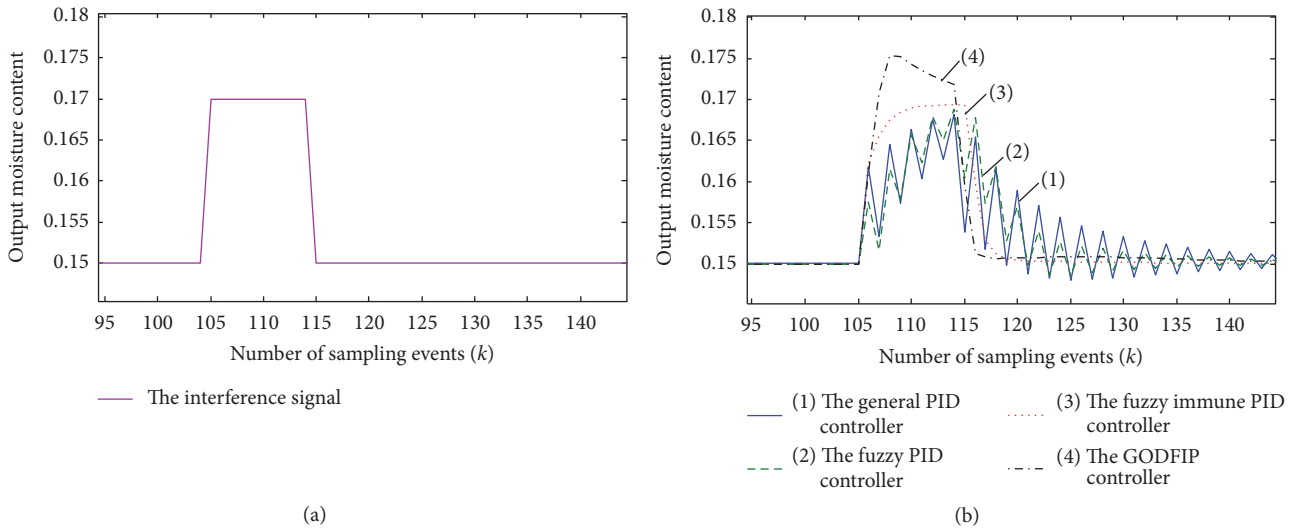


Figure 14: The anti-interference effect comparisons between the GODFIP controller and the other controllers: (a) the interference signal and (b) the anti-interference effect comparison results.

Table 4: Control performance comparisons of different controllers.

Controllers	J	δ	$y(t_p)$	t_r (number of samples)	t_{s1} (number of samples)	t_{s2} (number of samples)
General PID	26.046	0.13%	0.1502	57	0	39
Fuzzy PID	20.348	0.07%	0.1501	20	0	30
General fuzzy immune PID	1.7052	0.00%	0.1500	8	0	13
GODFIP	1.6450	0.00%	0.1500	5	0	10

control stability of the system, not oscillating wildly (in the drying control reflecting the output moisture content fluctuations), and the rapidity of the system response-rapidly tracking control of the target signal. By the simulation comparisons of Figure 13 and the control performance values of Table 4, owing to the adopted genetic optimization algorithm on the controllers, the overshoot and the adjusting time t_{s1} of all compared controllers are almost zero, showing the excellent optimal ability of genetic algorithm. It can also be seen that the rising time to the target value of the GODFIP controller is 5 samplings which is about 37.5% decrease compared to the general fuzzy immune PID controller (8 samples), about 75% decrease compared to the fuzzy PID controller (20 samples), and about 91.2% decrease compared to the general PID controller (57 samples), showing the effectiveness of the GODFIP control algorithm.

Under the same optimization conditions, the PID controller and the fuzzy PID controller need more samples to the target value than the other two immune controllers (the GODFIP and the fuzzy immune PID) in order to accomplish the optimal control target; in fact, it is impractical for the grain drying process because it will cause an inefficient energy consumption and bad dried grain quality. The GODFIP controller and the fuzzy immune PID controller are both superior to the PID controller and the fuzzy PID controller, which not only can the stability of the control system be achieved, but also the system output can be rapidly adjusted to the target value, showing the advantage of the immune algorithm. In addition, the GODFIP controller has performed better compared to the fuzzy immune PID controller, so the GODFIP controller is more suitable for the IRC dryer than the other compared controllers.

(3) The Anti-Interference Test. As can be seen from Figure 14, the GODFIP controller has the best anti-interference performances compared to the other three controllers that it can quickly respond to and track the interference signal and can adjust the output value to the target value rapidly and steadily after the interference disappears; moreover, the fluctuations affected by the interference are less. However, for the same test, the general PID controller and the fuzzy PID controller have showed oscillatory behavior when an interference exits during the drying process, of which the general PID controller has taken nearly 39 samples and the fuzzy PID controller has taken nearly 30 samples to reach the desired moisture content level, showing the inefficiency of handling the effect of interference by the two controllers and inferior to the anti-interference ability of the GODFIP controller. Moreover, the GODFIP controller has improved the anti-interference performance of the fuzzy immune PID controller of which 3 samples to the target value are reduced compared to the fuzzy immune PID controller. The anti-interference performance test further verifies the robustness of the proposed GODFIP controller over an uncertainty

range of condition and demonstrates that it is a suitable controller for the IRC grain dryer.

5. Conclusion

In this paper, a genetic optimization dual fuzzy immune PID (GODFIP) controller based on the immune feedback mechanism is designed and simulated to control an IRC grain dryer represented by an identified Autoregressive with Exogenous input (NARX) model. The NARX model has a higher model approximation accuracy (MSE: $1.062 * 10^{-5}$; R: 99.78%) than the identified linear ARX model (MSE: $2.29 * 10^{-4}$; R: 95.5%), so the NARX model is a better candidate in representing the nonlinear dynamics of the IRC grain dryer to verify the control performances of the GODFIP controller. In order to achieve a fair comparison, the genetic optimization algorithm is utilized to optimize the proposed GODFIP controller and the other three compared controllers based on the performance objective function J from the aspects of energy savings, stability, accuracy, and rapidity. Finally, the control simulation comparison and the anti-interference simulation test are made. As can be seen from the simulation results of Figures 13 and 14 and Table 4, the GODFIP controller has the least value of J compared to the other three controllers which is equal to 1.645 about 3.53% decrease compared to the fuzzy immune PID, about 91.9% decrease compared to the fuzzy PID controller, and about 93.68% decrease compared to the PID and the shortest rising time about 37.5% decrease compared to the fuzzy immune PID, about 75% decrease compared to the fuzzy PID, and about 91.2% decrease compared to the PID and has the best anti-interference ability which can adjust the output to the target value rapidly and steadily. It is tested from the simulation results that the GODFIP controller has improved the control performance of the fuzzy immune PID controller and is obviously superior to the PID controller and the fuzzy PID controller, so the GODFIP controller can track the target value rapidly and steadily and can handle the uncertainty conditions of complex systems to some extent which is more suitable for such a complex system as grain drying. The big difference between this control method and the traditional control method is that it is not dependent on the transfer function of the controlled object. The proposed GODFIP controller can provide an effective reference for the actual control strategy of the grain drying process and may be applied to control the real IRC dryer in future works.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Aini Dai mainly engaged in the research of grain drying control and intelligent control. Xiaoguang Zhou mainly engaged in the research of control theory and its application in engineering. Xiangdong Liu mainly engaged in the research of drying technology and theory of agricultural products.

Acknowledgments

The authors of this paper would like to acknowledge the China National Common Weal Industrial Special Scientific Research Funds for Grain Industry (no. 201413006) and the 111 Project (B08004) for funding of this study, the cooperation of the research group, and the experimental help of the engineers in Dongyu Machinery Co. Ltd., China.

References

[1] Q. Liu and F. W. Bakker-Arkema, "Automatic control of cross-flow grain dryers, part 2: design of a model-predictive controller," *Journal of Agricultural Engineering Research*, vol. 80, no. 2, pp. 173–181, 2001.

[2] Q. Liu and F. W. Bakker-Arkema, "A model-predictive controller for grain drying," *Journal of Food Engineering*, vol. 49, no. 4, pp. 321–326, 2001.

[3] Y. K. Pan, X. Z. Wang, and X. D. Liu, *Modern Drying Technology*, Chemical Industry Publisher, Beijing, China, 2007.

[4] M. H. Riadh, S. A. B. Ahmad, M. H. Marhaban, and A. C. Soh, "Infrared heating in food drying: an overview," *Drying Technology*, vol. 33, no. 3, pp. 322–335, 2015.

[5] X. Wang, Q. Hu, B. Xiao, D. Yang, and X. Liu, "Modeling simulation of combined convective and infrared radiation in rice drying process," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 44, no. 9, pp. 145–151, 2013.

[6] P. Dufour, D. Blanc, Y. Touré, and P. Laurent, "Infrared drying process of an experimental water painting: Model predictive control," *Drying Technology*, vol. 22, no. 1-2, pp. 269–284, 2004.

[7] L. Luo, "A mathematical model for vacuum far-infrared drying of potato slices," *Drying Technology*, vol. 32, no. 2, pp. 180–189, 2014.

[8] T. M. Afzal and T. Abe, "Some fundamental attributes of far infrared radiation drying of potato," *Drying Technology*, vol. 17, no. 1-2, pp. 137–155, 1999.

[9] R. Dhib, "Infrared drying: from process modeling to advanced process control," *Drying Technology*, vol. 25, no. 1, pp. 97–105, 2007.

[10] J. Wang, "A single-layer model for far-infrared radiation dying of onion slices," *Drying Technology*, vol. 20, no. 10, pp. 1941–1953, 2002.

[11] C. Likitrattanaporn and A. Noomhorm, "Effects of simultaneous parboiling and drying by infrared radiation heating on parboiled rice quality," *Drying Technology*, vol. 29, no. 9, pp. 1066–1075, 2011.

[12] A. S. Mujumdar, "Editorial: the Making of the Handbook of Industrial Drying," *Drying Technology*, vol. 32, no. 6, pp. 627-628, 2014.

[13] D. M. Bruce and N. J. B. McFarlane, "Control of mixed-flow grain driers; testing of a feedback-plus-feedforward algorithm," *Journal of Agricultural Engineering Research*, vol. 52, no. C, pp. 11–23, 1992.

[14] O. F. Lutfy, S. B. Mohd Noor, M. H. Marhaban, and K. A. Abbas, "Non-linear modelling and control of a conveyor-belt grain dryer utilizing neuro-fuzzy systems," *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, vol. 225, no. 5, pp. 611–622, 2011.

[15] M. Negnevitsky, *Artificial intelligence: a guide to intelligent systems*, Addison-Wesley publisher, New Jersey, NJ, USA, 2005.

[16] F. Han, W. F. Wu, and H. Zhu, "Control status and developing trend of grain drying process," *Journal of The Chinese Cereals and Oils Association*, vol. 24, no. 5, pp. 150–153, 2009.

[17] S. M. Lewis, G. Fitts, M. Kelly, and L. Dale, "A fuzzy logic-based spatial suitability model for drought-tolerant switchgrass in the United States," *Computers and Electronics in Agriculture*, vol. 103, pp. 39–47, 2014.

[18] S. El Ferik, M. Tariq Nasir, and U. Baroudi, "A behavioral adaptive fuzzy controller of multi robots in a cluster space," *Applied Soft Computing*, vol. 44, pp. 117–127, 2016.

[19] A. V. Taprantzis, C. I. Siettos, and G. V. Bafas, "Fuzzy control of a fluidized bed dryer," *Drying Technology*, vol. 15, no. 2, pp. 511–537, 1997.

[20] H. Mansor, S. B. Mohd Noor, R. K. Raja Ahmad, F. S. Taip, and O. F. Lutfy, "Intelligent control of grain drying process using fuzzy logic controller," *Journal of Food, Agriculture and Environment*, vol. 8, no. 2, pp. 145–149, 2010.

[21] N. Perrot, C. Bonazzi, and G. Trystram, "Application of fuzzy rules-based models to prediction of quality degradation of rice and maize during hot air drying," *Drying Technology*, vol. 16, no. 8, pp. 1533–1565, 1998.

[22] Q. Zhang and J. B. Litchfield, "Knowledge representation in a grain drier fuzzy logic controller," *Journal of Agricultural Engineering Research*, vol. 57, no. 4, pp. 269–278, 1994.

[23] O. F. Lutfy, H. Selamat, and S. B. Mohd Noor, "Intelligent modeling and control of a conveyor belt grain dryer using a simplified type 2 neuro-fuzzy controller," *Drying Technology*, vol. 33, no. 10, pp. 1210–1222, 2015.

[24] M. S. Liu, C. W. Wu, and H. Wang, "Simulation of grain drying process by a fuzzy expert system," *Transactions of the Chinese Society of Agricultural Machinery*, vol. 32, no. 4, pp. 54–56, 2001.

[25] L. Sun and Z. Zhu, "Fuzzy Aeration Control for In-Bin Stored Grain," in *Proceedings of the International Conference on Digital Manufacturing and Automation, ICDMA '13*, pp. 626–629, June 2013.

[26] S. K. Tiwari and G. Kaur, "Analysis of fuzzy pid and immune pid controller for three tank liquid level control," *International Journal of Soft Computing and Engineering*, vol. 1, no. 4, 2011.

[27] T. Zheng, X. Xiao, and Y. Ren, "Fuzzy immune PID speed control system and its simulation in industrial sewing machine," *Mechanical and Electrical Engineering Magazine*, vol. 25, no. 12, pp. 19–22, 2008.

[28] F. Hu and W. N. Zhou, "Application of fuzzy immune PID controller based on particle swarm optimization in power plant steam temperature control system," *Advanced Materials Research*, vol. 950, pp. 257–262, 2014.

[29] G. D. Li and M. X. Wang, "Fuzzy immune-PID controller and its application in caustic process," *Computer Engineering*, vol. 35, no. 1, pp. 218–220, 2009.

[30] T. Zhou, *Introduction to the System Identification of Control*, Tsinghua University Publisher, Beijing, China, 2004.

[31] F. Soderstrom, "Least squares parameter estimation of continuous-time ARX models from discrete-time data," *IEEE Transactions on Automatic Control*, vol. 42, no. 5, pp. 659–673, 1997.

[32] D. Liberati, S. Cerutti, E. Di Ponzio, V. Ventimiglia, and L. Zaninelli, "The implementation of an autoregressive model with exogenous input in a single sweep visual evoked potential analysis," *Journal of Biomedical Engineering*, vol. 11, no. 4, pp. 285–292, 1989.

[33] W.-X. Zhao and W. X. Zheng, "Nonparametric algorithms for identification of nonlinear autoregressive systems with exogenous inputs," *IFAC Proceedings Volumes*, vol. 42, no. 10, pp. 1609–1614, 2009.

[34] N. Mohd and N. Aziz, "Performance and robustness evaluation of nonlinear autoregressive with exogenous input model predictive control in controlling industrial fermentation process," *Journal of Cleaner Production*, vol. 136, pp. 42–50, 2016.

[35] B. C. Ng, I. Z. M. Darus, H. Jamaluddin, and H. M. Kamar, "Dynamic modelling of an automotive variable speed air conditioning system using nonlinear autoregressive exogenous neural networks," *Applied Thermal Engineering*, vol. 73, no. 1, pp. 1253–1267, 2014.

[36] R. W. K. Chan, J. K. K. Yuen, E. W. M. Lee, and M. Arashpour, "Application of Nonlinear-Autoregressive-Exogenous model to predict the hysteretic behaviour of passive control systems," *Engineering Structures*, vol. 85, pp. 1–10, 2015.

[37] C. V. Mai, M. D. Spiridonakos, E. N. Chatzi, and B. Sudret, "Surrogate modeling for stochastic dynamical systems by combining nonlinear autoregressive with exogenous input models and polynomial chaos expansions," *International Journal for Uncertainty Quantification*, vol. 6, no. 4, pp. 313–339, 2016.

[38] A. Andalib and F. Atry, "Multi-step ahead forecasts for electricity prices using NARX: a new approach, a critical analysis of one-step ahead forecasts," *Energy Conversion and Management*, vol. 50, no. 3, pp. 739–747, 2009.

[39] L. T. Koczy, "Fuzzy if. then rule models and their transformation into one another," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 26, no. 5, pp. 621–637, 1996.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 1670709, 11 pages
https://doi.org/10.1155/2017/1670709

Research Article

A Parallel Biased Random-Key Genetic Algorithm with Multiple Populations Applied to Irregular Strip Packing Problems

Bonfim Amaro Júnior, Plácido Rogério Pinheiro, and Pedro Veras Coelho

Graduate Program in Applied Informatics, University of Fortaleza (UNIFOR), Av. Washington Soares, 1321 Bl J Sl 30, 60.811-905 Fortaleza, CE, Brazil

Correspondence should be addressed to Plácido Rogério Pinheiro; placido@unifor.br

Received 2 April 2017; Revised 6 July 2017; Accepted 1 August 2017; Published 17 September 2017

Academic Editor: Jorge Magalhaes-Mendes

The irregular strip packing problem (ISPP) is a class of cutting and packing problem (C&P) in which a set of items with arbitrary formats must be placed in a container with a variable length. The aim of this work is to minimize the area needed to accommodate the given demand. ISPP is present in various types of industries from manufacturers to exporters (e.g., shipbuilding, clothes, and glass). In this paper, we propose a parallel Biased Random-Key Genetic Algorithm (μ-BRKGA) with multiple populations for the ISPP by applying a collision-free region (CFR) concept as the positioning method, in order to obtain an efficient and fast layout solution. The layout problem for the proposed algorithm is represented by the placement order into the container and the corresponding orientation. In order to evaluate the proposed (μ-BRKGA) algorithm, computational tests using benchmark problems were applied, analyzed, and compared with different approaches.

1. Introduction

The large market dispute between manufacturing and exporters, coupled with the scarcity of some items that make up the raw material for a product manufacturing, has motivated research around the world to find answers that reproduce efficient solutions at a low cost.

Cutting and packing (C&P) problems belong to NP-hard [1]. A dataset of items must be packed in a two-dimensional stage and an objective function can be analyzed through the minimum area needed to place pieces or the maximum number of items allowed in a current layout. Our research focus is on a particular C&P problem commonly referred to as the irregular strip packing problem (ISPP).

Furthermore, according to the typology of [1], the ISPP has an open arbitrary dimension classification. A variation of C&P can be described as a container (C) with a constant width (W) and a variable length (L) and a dataset of irregular polygons, where the objective is to minimize L and place (with no overlaps) the entire demand into C. In some cases all angles to rotate polygons are accepted; in other cases only some angles are allowed to rotate the polygons. Figure 1 illustrates a classical instance of this problem known as Trousers [2].

Various studies have been devoted to obtain solutions (layouts) that minimize surface area and computational time. However, despite the many proposals that have been presented, the analysis of further strategies to obtain improved layouts is still pertinent. Some examples of strategies are as follows:

(i) A positioning rule must be applied to the list of parts

(ii) A precise layout is accepted as the initial solution, and overlapping of items is valid, but a penalty is applied to the objective function

(iii) A set of clusters between parts is performed in the preprocessing phase to reduce the complexity of the fit among the parts

(iv) Other mathematical programming models.

The approach proposed here was a parallel computing implementation of the Biased Random-Key Genetic Algorithm (BRKGA) [3] using multiple populations applied to an ISPP. Each solution develops a sequence in which the

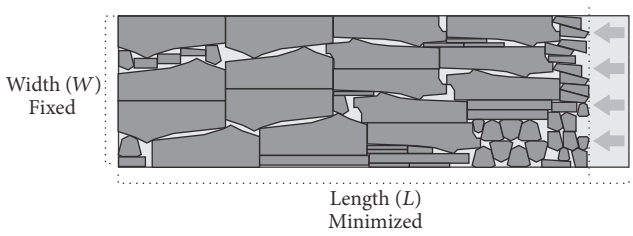


Figure 1: An example of a solution to an irregular strip packing problem.

pieces are positioned in the container. The current angle and the method of placement applied was based on a collision-free region [4]. Computational tests carried out on dataset examples of the target problem are discussed, and the method presented here is shown to be compatible with or better than some other important studies.

This paper is structured as follows. Section 2 presents some studies about ISPPs and their different approaches to solve them. Section 3 introduces important concepts associated with the suggested methodology, which is presented in Section 4. Documentation concerning computational tests on well-known literature datasets is given in Section 5. Finally, in Section 6 the conclusions concerning the proposed methodology are presented along with some suggestions to develop this work further.

2. Approaches to Tackle Irregular Strip Packing Problems

ISPPs are a challenge since they present obstacles for the construction of feasible and optimized solutions. Among these are the geometric aspects of the polygons and the combinatorial characteristics for quality layouts. These two obstacles must be managed concomitantly. In addition, an estimate of the computational complexity of each approach must be measured to serve as a criterion to select the methods for the planning phase in algorithmic projects.

Various mathematical programming models have been developed to provide feasible solutions for ISPPs [5, 6]. Although there are still constraints for industrial scale applications, due to combinatorial characteristic optimization, among other factors, the entire computational concept arising from this type of strategy introduces a theoretical foundation of great importance to research in this area.

For instance, in [7], the authors presented a model based on constraint logic programming (CLP) adapted to ISPPs. Also, this strategy was applied to convex and nonconvex polygons. The concepts of NFP and IFP were used to satisfy the geometric constraint of overlap and to show the domain of the positioning of the first item. The composition of the layouts, using the fit patterns, adopts a CLP model with variables within a finite domain. The width, the upper boundary of the stage place, and the geometry of the pieces from the dataset compose the initial variables of the proposed model. Once the initial settings are defined, the approach has a layer that manages the constant changes in the variable domains. Moreover, it handles the consistency of the process.

On the other hand, the authors in [8] presented a mixed integer linear programming (MILP) approach to ISPPs based on the work of [9]. The aim was to maximize efficiency while minimizing the length needed for the layout. Another aim of the objective function was that all the pieces must still be as near as possible to their origin position, as if compacted. This process is carried out by minimizing the coordinates of the reference points of each polygon. The objective function calculation does not consider the wastage and holes between the items. All tests were made for cases in the garment industry, and the model runtime was just a few seconds.

Likewise, in [10], the authors also proposed an MILP; however, the decision variables, in this case, are binary and are associated with each discrete point present on board, a nomenclature for the container. An outstanding advantage of this model is its flexibility to tackle the geometry of the items and the container and therefore allowing an extension to more complex problems. For instance, this approach can be applied to nonconvex boards and items with holes. On the other hand, there is a limitation on the number of parts to run in a feasible time.

Another model based on CLP was presented by [11]. The researchers reported constraints aiming to ensure all feasibility requirements to achieve a solution. By changing the definition of the variables, which have a binary domain, this approach diverges from the models referred to by [7, 12] which can be applied to the problem with fixed containers.

Nevertheless, methods from various fields of computational science, such as the concepts of metaheuristics and dynamic programming, have been applied to produce feasible solutions for ISPPs and to reduce their computational times. Several approaches in the literature have considered these concepts in order to obtain estimates for their use in real cases [13–15]. However, this type of strategy leads us to a thorough investigation of the calibration parameters.

In the field of metaheuristics, Jakobs [16] exposed an approach that combines a genetic algorithm and a Bottom-Left (BL) strategy as a constructive heuristic for the positioning of items. The chromosome coding was mapped as an ordered sequence of parts, also applied by [17, 18]. On the other hand, in [19], the authors used the same idea, but a raster representation for the pieces.

Incidentally, in [20], the authors proposed a probabilistic heuristic known as 2-exchange as a parameter Δ to control the change between neighbors items. This feature coordinates the size of the search between the neighborhoods by assigning the maximum amount of permutations between items in a complete sequence. Also, the authors used an NFP-based mechanism to position pieces in spaces left empty by the solutions.

Furthermore, in [21] these authors presented a new constructive heuristic based on the BL with the characteristic to fill empty spaces in the layouts, innovating one of the biggest limitations of the traditional method. The translation of the items begins on the Bottom-Left side, and the horizontal sliding is done discretely, from a point inside a grid, while vertical movement is performed in continuous space. So, it determines the valid positions; the NFP algorithm was applied. This technique is combined with hill climbing and a tabu search.

Another strategy was developed by [4]. In this paper, the approach was divided into two steps, aiming to obtain efficient solutions for ISPPs. The initial stage consists of the Simulated Annealing algorithm, considering the dimensions of a container as being fixed. The initial solution is randomly generated, the sequence of positioning items is represented by integer values as well as the permissible rotations for each case, and the location point is an ordered pair of real numbers within the interval $[0, 1]$.

At the last level, the control over two attributes of the approach is performed: the value of the size of the container dimensions and the initial temperature applied Simulated Annealing (SA). As soon as a layout is found, at the end of the first stage, this level reduces the area and restarts the SA process. If no solution is found, the container length is increased and the initial stage restarted. Also, two parameters are used to manage compaction and expansion, respectively, of the container. The authors of [22–24] applied the same idea as this approach but used a method of relaxed positioning method with the aim of minimizing the amount of overlap between the pieces.

Techniques that combine models in mathematical programming and artificial intelligence strategies are also possible. These approaches are known as hybrids and, similarly, several studies have investigated their results and limitations.

For instance, in [7], the authors proposed an approach that differs from others by not requiring a positioning rule. A discrete container and a CLP model was adopted, and the objective function was to find the best point of placing, besides the current fixed solution.

In [25], a Simulated Annealing (SA) was combined with a linear programming model. First an initial layout is obtained by applying the Bottom-Left method, and each piece is selected according to a length-based criteria. The SA guides the search over the solution space where each neighborhood structure handles linear programming models, which is composed of a compaction and a separation algorithm.

Another combination between local and guided search applied to two- and three-dimensional items was proposed in [24]. The Bottom-Left method finds an initial layout length. By reducing this value, there are overlapping situations, which are removed by a local search through one of the following changes: horizontal and vertical translations, rotation, or flipping. The guided local search is adopted to try escape of solutions in local minima. Another outstanding aspect is the computational complexity. Given a simple polygon with m vertices and a set of n edges, the algorithm can find the translation with the minimum number of overlays in $O(mn \log(mn))$.

Also, an extended local search algorithm based on nonlinear programming is designed in [23]. The procedure starts from a feasible layout, and its length is set as best. Then a new layout is obtained by random exchanges between two polygons at the current solution. Through a time limit, the length is reduced, and a local search manages the overlap subproblem.

The approaches presented in this section were divided into three distinct categories as [26] mathematical programming, metaheuristics, and mixed strategies. For our research, we adopted a Biased Random-Key Genetic Algorithm (BRKGA) applied with multiple parallel populations to tackle ISPP.

3. Supporting Concepts

Before discussing the proposed algorithm, we need to describe some supporting concepts. These concepts are geometric representations; no-fit polygon (NFP), and Inner-Fit Polygon (IFP); collision-free region (CFR); and Biased Random-Key Genetic Algorithm (BRKGA).

3.1. Geometric Representations. The geometric representation of nesting problems is an important choice to obtain feasible solutions and will impact directly on the availability of geometric tools that can be used to perform the overlap detection in the layout. The problem aspect depends on the adopted representation. The impact of choice depends on the accuracy of the overlap detection, on the memory requirements needed, and on the computational performance to execute the approach. There are two basic types of representation for polygons: raster or geometric (Figure 2).

Raster representation uses a matrix to describe the geometry of an item. This matrix is obtained from the definition of a default size grid. A grid position that has some internal region of the item is marked with a different value. The rasterized representation has the advantage of being able to represent generic geometry items, and there are simple geometric tools to deal with overlaps. However, if the grid resolution is low, the shape of the object may be different and change a solution. But if the resolution of the grid is high, the computational cost to find overlaps is expensive.

Moreover, polygonal representation uses polygons to represent the items. A list of bidimensional vectors containing the coordinates of vertices is applied to compose the shape of the item. Sometimes, the items cannot be represented in their exact form; in these cases, an approximation of vertices is applied. The geometric calculations to overlap detection are complex. Due to their precision, several approaches in literature use polygonal representation.

There are other representations. However, there has been little research with them. To typify the pieces that need better points precision, linear or nonlinear functions can be applied (phi-functions). This representation strategy is the mathematical expression obtained from the distance equation between two objects. They were first presented and applied by [27]. The combinations of primary objects serve to compose complex ones.

The ISPP demands a suitable geometrical representation when continuous rotations are necessary for a set of pieces. In these cases, pieces can be represented by circles (Circle Covering) [28]. Due to the irregular pattern of pieces, the circle representation, of the problem in case, needs to agree with an approximation of polygons.

3.2. No-Fit Polygon (NFP) and Inner-Fit Polygon (IFP). This trigonometric technique was presented in [29] as a shape envelop and changed to NFP by [30]. The idea behind this

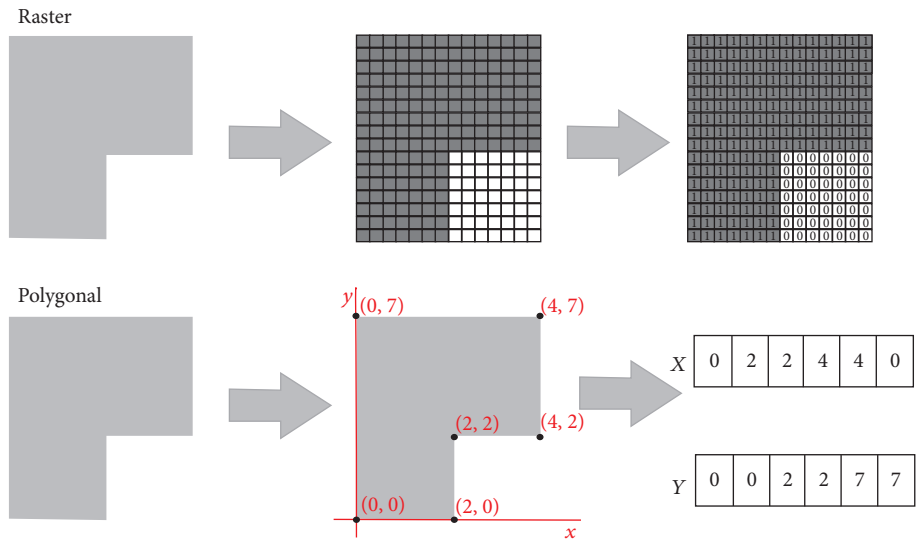


Figure 2: A raster and polygonal representation.

trigonometric technique is as follows: given two polygons, P_i (the fixed piece) and P_j (the orbital piece), and a reference point on P_j called r_j, the no-fit polygon of P_i in relation to P_j denoted as $\text{NFP}_{(i,j)}$ is the locus of points traced by r_j when P_j slides around the external contour of P_i as shown in Figure 3.

From this definition, three situations may arise:

(i) If r_j stays inside $\text{NFP}_{(i,j)}$ *(grey area)* then P_j intersects P_i.

(ii) If r_j strays over the boundary of $\text{NFP}_{(i,j)}$ then P_j touches P_i.

(iii) If r_j stays outside $\text{NFP}_{(i,j)}$ then P_j does not intersect or touch P_i.

In our implementation, all possible NFPs are computed in a preprocessing level applying the Minkowski sum [31] method and polygon decomposition is based on [32]. Given two arbitrary point sets, A and B, the Minkowski sum of A and B is represented as $A \oplus B = \{a + b: a \in A, b \in B\}$. A simple vector algebra can be applied to show that $A \oplus (-B)$, known as the Minkowski difference of A and B, is equivalent to $\text{NFP}_{(A,B)}$ since a stipulation that polygons stay in counter-clockwise orientation, $-i$, is simply B in a clockwise direction.

The concept of NFP as described is applied only for convex polygons. However, the application of decomposition in nonconvex polygons normally constructs several subpolygons, and consequently many no-fit polygons must be created. In computational times, a unique no-fit polygon intersection is fast, but an additional calculation must be considered to recombine all decomposed parts. Therefore, if a large number of parts are needed or the polygon contains holes, the time to complete this operation is expensive.

In [33], the authors analyzed different decomposition and applied recombination operations to construct the Minkowski sums of nonconvex polygons. The first inference

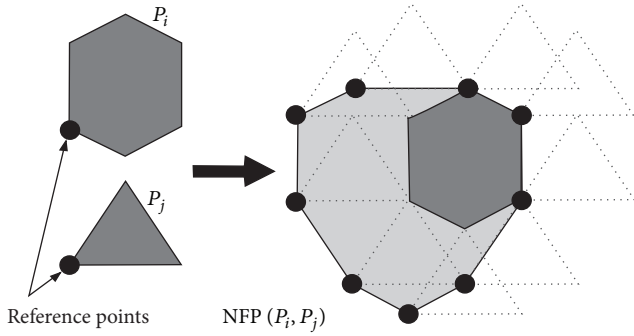


Figure 3: The no-fit polygon $\text{NFP}_{(i,j)}$ of two convex polygons P_i and P_j.

is that the computation times to find optimal decomposition does not compensate the possible benefits that may occur in the recombination process.

In [34] the author presented an approximation algorithm which takes $O(n \log n)$ time and $O(n)$ space, where n is the vertices number of the input polygon and outputs a decomposition whose size is guaranteed to be no more than four times the size of the optimal decomposition.

As observed in [32], the authors proposed an efficient heuristic for decomposing a polygon into convex subpolygons. The algorithm works this way. Initially, it tries to subdivide the polygon by connecting two reflex vertices with an edge. When this is not possible anymore, it eliminates all reflex vertices by connecting them to other convex vertices, such that the new edge best approximates the angle bisector of the reflex vertex. The algorithm operates in $O(n^2)$ time and takes $O(n)$ space in the worst case, where n is the number of vertices of the input polygon.

The Inner-Fit Polygon (IFP) [20] is similar to the NFP except that the orbiting polygon, in this case, slides around

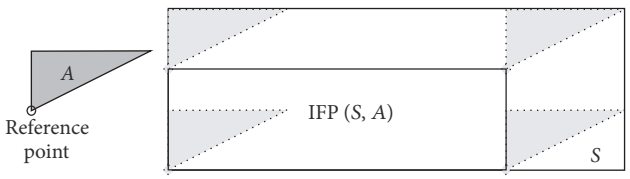


Figure 4: The Inner-Fit Polygon $IFP_{(S,A)}$ of stage S and piece A.

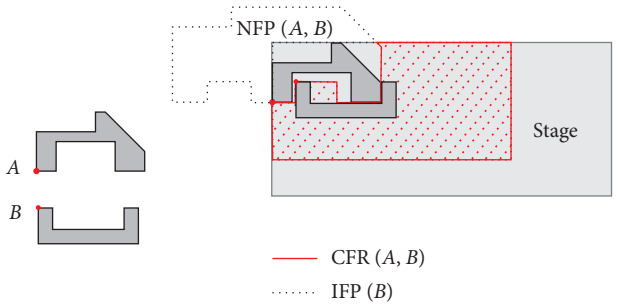


Figure 5: The collision-free region $CFR_{(S,B)}$ of stage S and piece B.

inside the stationary polygon. The IFP represents a set of points that allow the placement of a polygon inside a hole of another polygon, usually the container. Approaches to tackle NFP can be adjusted to handle IFP. If the outline of the IFP is a rectangle, the IFP is called Inner-Fit-Rectangle (IFR). This happens when generating the IFP from the container, since it is, most of the time, a rectangle. An IFR can be seen in Figure 4.

3.3. Collision-Free Region (CFR). The collision-free region (CFR) is a geometric procedure that allows the feasible position points of a piece inside the strip to be found, efficiently. This technique was developed to plan robot motion in order to present the shortest path without hitting any obstacle. In [4], the CFR was adapted to tackle the irregular strip packing problem using a polygonal Boolean operation.

Consider a stage S and a next piece in sequence B. The $CRF_{(S,B)}$ is composed of the polygon difference (Figure 5) between (1) and (2) described next:

$$\text{(i)}\ \ IFP_{(S,B)} \tag{1}$$

$$\text{(ii)}\ \bigcup_{i=1}^{P} NFP\left(P_i, P_B\right). \tag{2}$$

In (1), the Inner-Fit Polygon between S and piece B is computed to find a polygon that represents all available points inside S. In the other equation (2), the resultant polygon is composed of a union between the NFPs of B and each piece already placed P_i. The basic case occurs when the stage is empty; then $CFR_{(S,B)} = IFP_{(S,B)}$.

3.4. Biased Random-Key Genetic Algorithm (BRKGA). The Biased Random-Key Genetic Algorithm (BRKGA) is an evolutionary metaheuristic for discrete and global optimization

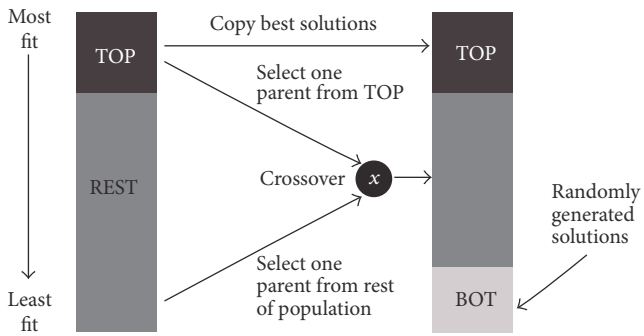


Figure 6: Population evolution process to next generations of a BRKGA.

problems based on [35]. Each solution is considered as a list of float numbers in the interval [0, 1], called random keys.

The initial population is generated with p vectors of random keys. This procedure is the starting point of the BRKGA method [3]. In all iterations, two subvectors are created, the set containing high-valued solutions (elite) and the residual individuals (nonelite). The best elements (elite) are transcribed, similarly, to the next population. A small number of random-key vectors (the mutants) are added to the population of the next iteration. The remaining elements of the population of the next iteration are generated by combining with the parameterized uniform crossover of [36], pairs of solutions, where one is elite and the other not.

The main difference between BRKGA and RKGA is the way that the operator of choice and crossover are implemented. In the RKGA procedure both parents are chosen from the entire population and in BRKGA one parent is always selected from the elite set, while the other is chosen from the nonelite (or, in some cases, the rest of the population). Both algorithms combine parents with uniform crossover based on [36], but in BRKGA, one parent belongs to the elite while the other is in the nonelite group. Figure 6 illustrates the evolutionary process for each new population with TOP, REST, and BOT parts of found individuals.

In [37], the authors proposed a RKGA for ISPP and for the positioning method applied three constructive procedures to the polygonal representation of the pieces. A solution is coded as a vector $X = (x_1, x_2, \ldots, x_n, x_{n+1}, \ldots, x_{2n}, \ldots, x_{3n})$, where n represents the number of items of instance. All chromosomes have $3n$ random keys in a real interval $[0, 1)$. The n's first decide the positioning sequence of items, then $x_{n+1}, \ldots, x_{2n}$ shows the angles of rotations and, finally, $x_{2n+1}, \ldots, x_{3n}$ stands for the placing rules BL, BL-fill [21], and some heuristics developed in [38].

Furthermore, a BRKGA to ISPP was developed by [39], but in this work, the researches applied a BL heuristic to the positioning items and presented an NFP-raster at a grid model introduced by [10]. Each individual is mapped by a chromosome with n alleles, containing keys with real values in the interval $[0, 1)$. The rotation of the pieces is verified by its corresponding key in the initial vector. Considering that each polygon has k allowable angles, the domain gap of the variables is divided into the same k subintervals.

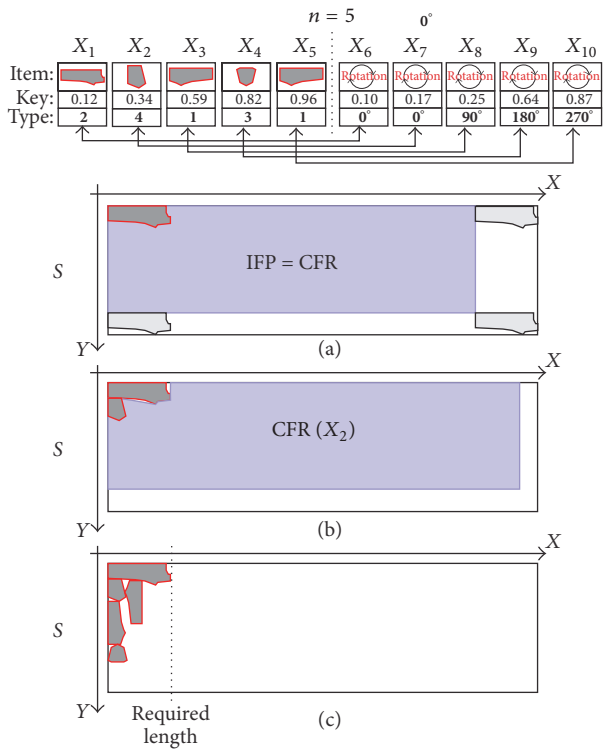


Figure 7: Collision-free region positioning method.

4. Proposed Methodology

In the μ-BRKGA heuristic, a chromosome is mapped as a list $X = \{X_1, X_2, \ldots, X_{2n}\}$, where n is the total number of pieces to be placed representing an initial solution. A dataset piece number and orientation stand for each gene. Any chromosome contains n genes, which are the total number of items to be packed. The variables $X_1, X_2, \ldots, X_n$ determine the Item Packing Sequence (IPS) and $X_{n+1}, X_2, \ldots, X_{2n}$ correspond to item packing angle (IPA). All alleles are set in ascending order.

The positioning rule applied to μ-BRKGA is based on the concept of the collision-free region [4] and, therefore, the evaluation considers the minimization of the length for the layout produced for each individual, as well as the maximization of the utilization. For instance, Figure 7 illustrates the procedure applied in a dataset with 5 items.

The positioning rule process receives a decoded vector as input in an ascending order according to the allele key. The CFR polygon is computed for all pieces at IPS rotated by their corresponding IPA. In Figure 7(a) the first piece, stage (S) is empty, so CFR(X_1) equals IFP(S, X_1). Also, in Figure 7(b) the remaining items are placed at the leftmost coordinate, choosing smaller x (left) and minor y (bottom), from the CRF polygon. Then for the last step of the process, the layout length can be assigned and the corresponding chromosome is evaluated (Figure 7(c)).

A BRKGA procedure is applied to a single population. However, this algorithm can be changed to accept more than one population. The method here uses μ populations, where μ depends on many of processor kernels on a target test computer. All initial populations are assigned some random solutions in an independent way. The information exchange is carried out according to the attribute P_{change} that defines the periodicity of this process. If $P_{\text{change}} = 1$, in every new generation, the population evolution process will be modified using the following rule:

(i) The "TOP" solutions P^e for each population P_i in Generation$_{(i)}$ are changed for worst layouts of "REST" solutions space of population P_{i+1} in Generation$_{(i+1)}$, considering P as a circular list, where the last element P_μ sends information to P_1; see Figure 8.

The main goal of this strategy is to improve the variability of good solutions between all populations, left in charge of the crossover process (REST), mutation solutions (BOT), elitism (TOP), and, at CFR positioning method, the work to construct new solutions to be evaluated and translated to next generations until stop criteria are verified. Figure 9 shows a flowchart of the proposed approach, discussing all levels: preprocessing, algorithm, and final solution.

In the preprocessing stage, all control parameters are defined (see Section 5) and assigned. For each instance selected, the no-fit polygons are computed for every pair of polygons type. At the next level, we considered our modified BRKGA (μ-BRKGA) with the information exchange rules and μ populations. In the last step, the best solution is picked as the one with the shortest length between all the layouts of each population adopted.

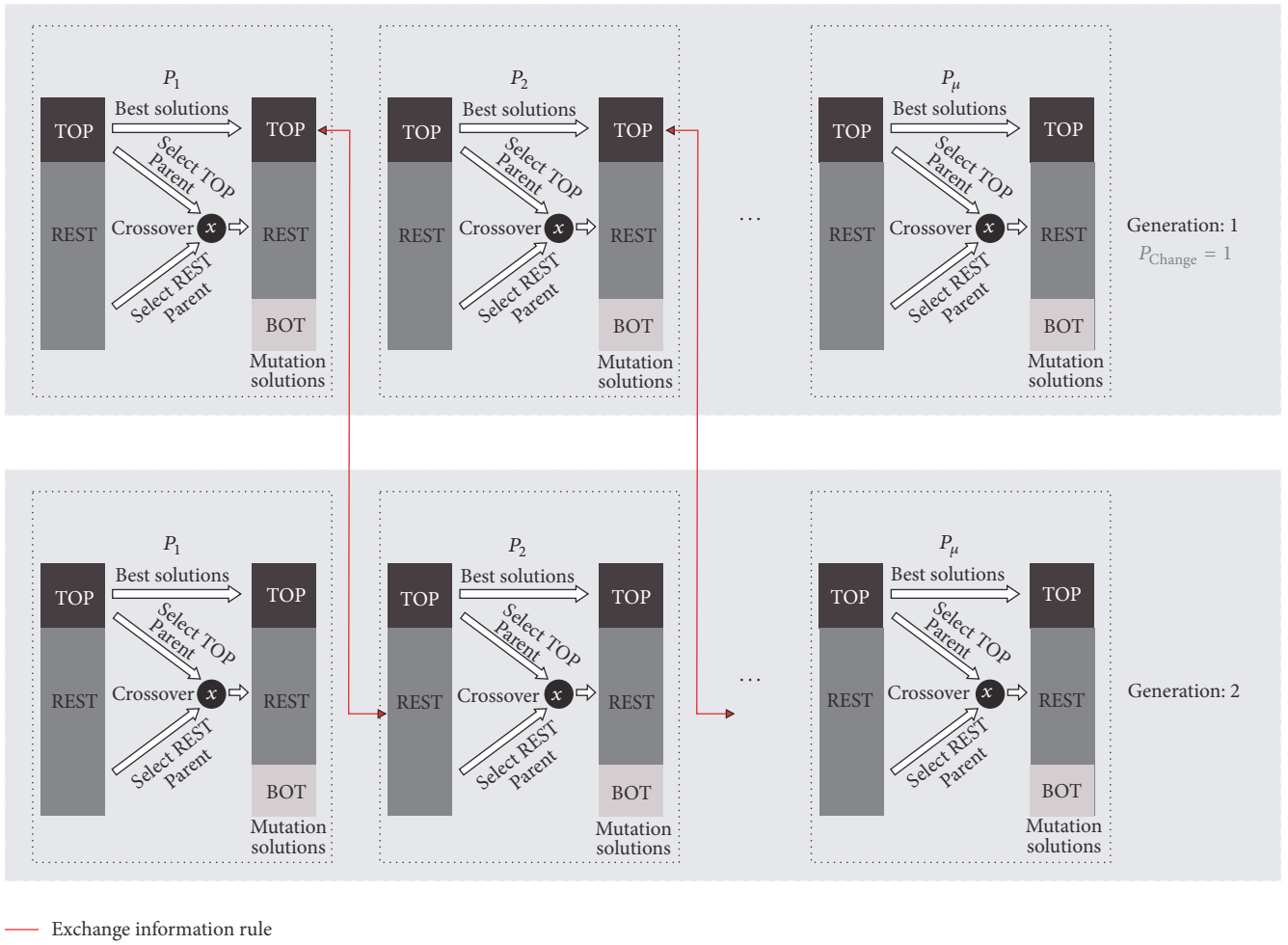


Figure 8: Information exchange rule for the next generations of a BRKGA with multiple populations.

5. Computational Experiments

The algorithm was written in Java Language. Computations tests were carried out on a machine with a 3.2 GHz Intel i7-960 CPU with 4 kernels ($\mu = 4$) and 8 GB of RAM memory. To measure the results of the μ-BRKGA, a set of tests was carried out on well-known problems extracted from the EURO Special Interest Group on Cutting and Packing (ESICUP) [40]. Also jigsaw puzzles (Dighe 1 and Dighe 2), clothes examples (Marques, Albano, Trousers, and Swim), and datasets created artificially (Jakobs 1, Jakobs 2, Shapes 0, and Shapes 1) were available. Table 1 presents some information of these examples. The following parameters for the parallel genetic algorithm with multiple populations were chosen: total number of individuals $P = 200 * \mu$ (200 per population), maximum number of generations of 200, $P^e = 0.3 * P$ (Elite parameter), $P^m = 0.2 * P$ (mutants parameter), $P^a = 0.85$ [36], $P_{change} = 5$ (information exchange processes), and the stop criteria were defined by P_{stop}. If the best solution has the same fitness as the P_{stop} generations, the algorithm ends. The value set for this parameter is 30.

For each instance, the μ-BRKGA algorithm was executed 10 times and the best results are presented in Table 2 with the focus on utilization percentage found in best solution and average of tests. Also, the same table shows a comparative analysis of the best use among the algorithms SAHA [25], BLF [21], 2DNest [21], BS [31], CFREFP [4], and GCSPC [41]. All these approaches have a prominent importance in the literature for problems of positioning irregular figures.

By analyzing Table 2, the μ-BRKGA application presents significant results in several datasets. In [37], a RKGA procedure for ISPP was implemented and the same tests were executed. Table 3 demonstrates that μ-BRKGA had a faster and better solution than RKGA-NP. The dataset named FU, which presented a prominent result in μ-BRKGA, was not applied in RKGA-NP.

In the RKGA-NP a different process of coding to represent a chromosome was applied and a group of random keys is responsible for choosing the respective positioning rule: Bottom-Left, Bottom-Left-Fill [21], and the heuristics developed in [38]. These characteristics contributed to the longer time required for the results shown in Table 3. However, the positioning method applied in μ-BRKGA reached better solutions in some cases. Figure 10 illustrates the final best layout for Albano, FU, Jakobs_1, and Trousers.

Table 1: Problem instances characteristics.

Instance	Quantity of different pieces	Quantity of pieces	Width	Orientations admitted
Albano	8	24	4900	{0°, 180°}
Dighe 1	16	16	100	{0°}
Dighe 2	16	16	100	{0°}
Jakobs 1	25	25	40	{0°, 90°, 180°, 270°}
Jakobs 2	25	25	70	{0°, 90°, 180°, 270°}
FU	12	12	38	{0°}
Marques	8	24	104	{0°, 90°, 180°, 270°}
Shapes 0	4	43	40	{0°}
Shapes 1	4	43	40	{0°, 180°}
Swim	10	48	5752	{0°, 180°}
Trousers	17	64	79	{0°, 180°}

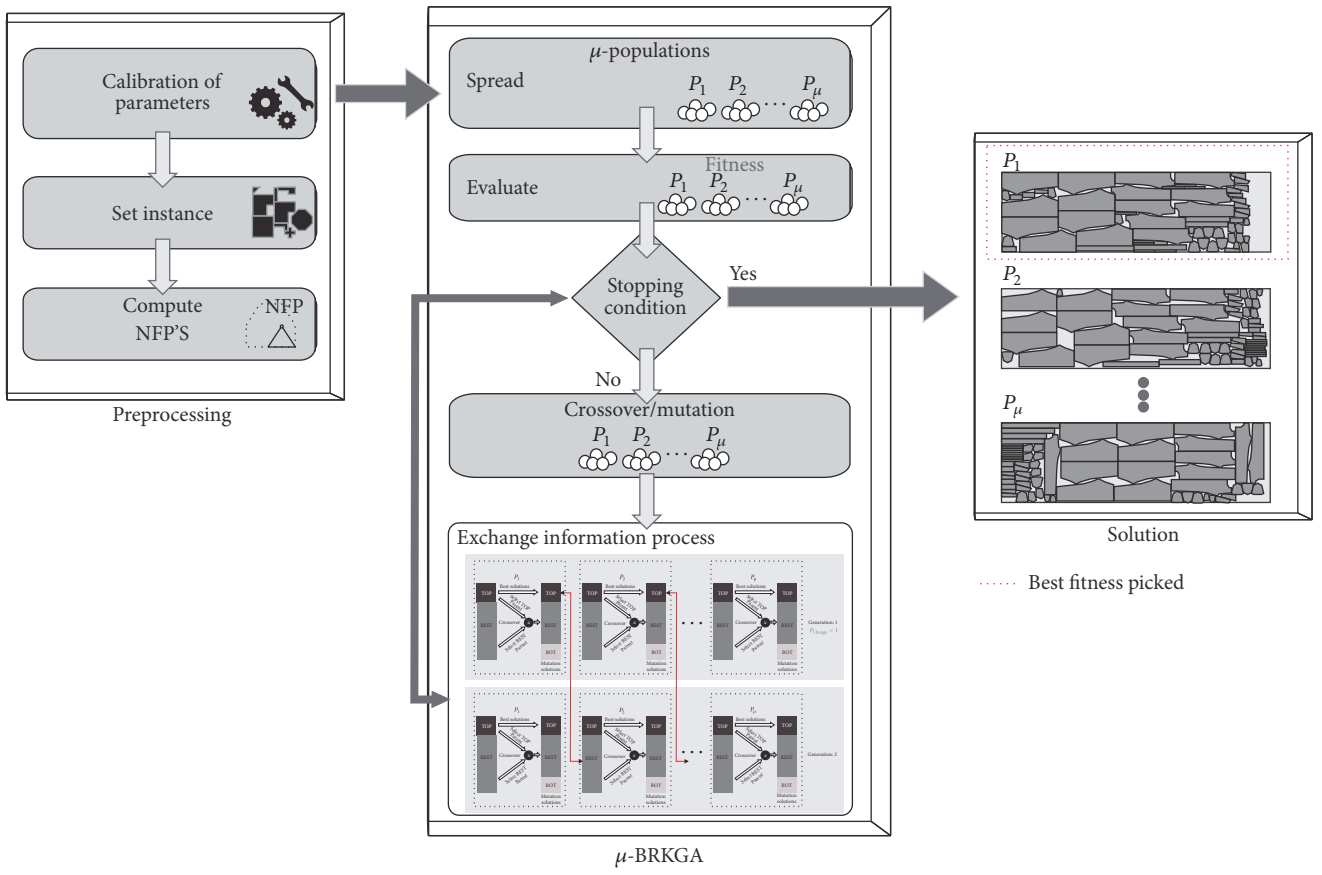


Figure 9: Flowchart of entire approach.

6. Conclusions and Future Work

An irregular strip packing problem with a rectangular stage, a fixed width, and an unlimited length was discussed in this paper. To tackle it, we combined a collision-free region placement procedure with a parallel Biased Random-Key Genetic Algorithm with multiple populations, which were also used to develop the order in which the polygons were placed within the stage and their corresponding orientations in order to find the most effective layouts.

ISPPs are present in many production processes in factories, and the approaches to overcome these problems need to find good solutions in a relative short time. In the method proposed here, we chose an effective metaheuristic combined with a convincing placement algorithm to carry this out. The parallel aspect helps to find the best solutions in the space available.

Computational results on datasets of ISPP show that the approach outperformed and/or matched well-known algorithms. As a future work, we aim to modify and implement

Table 2: μ-BRKGA × reference approaches.

Instance	SAHA	BLF	2DNest	BS	CFREFP	GCSPC	μ-BRKGA		
							Best (%)	Avg. (%)	Time (sec.)
Albano	87.43	84.60	87.88	87.88	89.21	89.58	**89.11**	**84.69**	**4200**
Dighe 1	100.00	77.40	99.84	100.00	100.00	100.00	**100.00**	**100.00**	**1920**
Dighe 2	100.00	79.40	93.02	100.00	100.00	100.00	**100.00**	**100.00**	**2050**
Jakobs 1	75.89	82.60	89.03	85.96	89.07	89.10	**89.09**	**78.79**	**6300**
Jakobs 2	77.28	74.80	81.07	80.40	84.83	87.73	**79.75**	**77.90**	**7150**
FU	90.96	86.9	92.03	88.48	91.96	91.96	**91.96**	**88.34**	**1815**
Marques	88.14	86.50	89.82	88.92	90.01	90.59	**89.14**	**84.31**	**3240**
Shapes 0	66.50	60.50	66.42	65.41	67.59	68.79	**66.50**	**63.50**	**1200**
Shapes 1	71.25	66.50	73.23	72.55	72.52	76.73	**72.55**	**65.98**	**1860**
Swim	74.37	68.40	72.49	75.04	71.78	75.94	**73.59**	**70.78**	**7400**
Trousers	89.96	88.50	90.46	90.38	90.07	91.00	**90.52**	**88.67**	**7250**

Table 3: Results of RKGA-NP [37] versus μ-BRKGA.

Instance	RKGA-NP			μ-BRKGA		
	Best (%)	Avg. (%)	Time (sec.)	Best (%)	Avg. (%)	Time (sec.)
Albano	84.74	84.69	10230	**89.11**	**84.69**	**4200**
Dighe 1	100.00	99.96	2053	**100.00**	**100.00**	**1920**
Dighe 2	100.00	99.99	2291	**100.00**	**100.00**	**2050**
Jakobs 1	81.67	78.79	10268	**89.09**	**78.79**	**6300**
Jakobs 2	79.75	77.90	12205	**79.75**	**77.90**	**7150**
FU	—	—	—	**91.96**	**88.34**	**1815**
Marques	84.47	84.31	6871	**89.14**	**84.31**	**3240**
Shapes 0	63.74	63.50	1826	**66.50**	**63.50**	**1200**
Shapes 1	66.37	65.98	3025	**72.55**	**65.98**	**1860**
Swim	73.59	70.78	14352	**73.59**	**70.78**	**7400**
Trousers	90.07	88.67	16025	**90.52**	**88.67**	**7250**

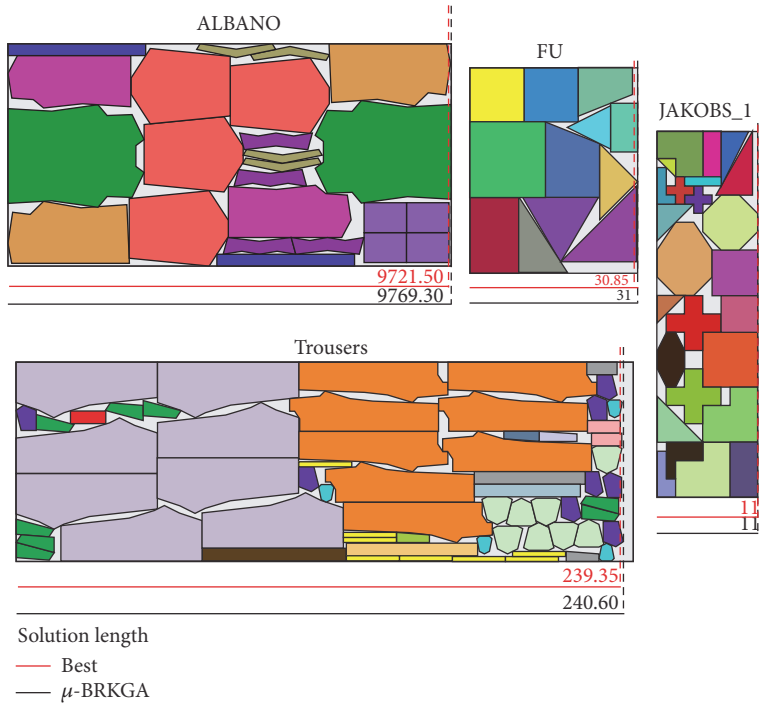


Figure 10: Results for datasets FU, Dighe 1, Dighe 2, and Trousers.

different placement procedures; these can be TOPOS [2], Bottom-Left-fill [21], best-fit algorithms, and a more effective method to change information between all populations to be combined with the parallel random-key genetic algorithm. Another possibility is to investigate two-dimensional block building algorithms to place the polygons in pairs and thus reduce the computational time since that approach is fundamentally based on a metaheuristic process.

Conflicts of Interest

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

Acknowledgments

The authors are thankful to Coordination for the Improvement of Higher Level Education Personnel (CAPES), National Counsel of Technological and Scientific Development (CNPq), via Grant no. 475239/2012-1.

References

[1] G. Wäscher, H. Haußner, and H. Schumann, "An improved typology of cutting and packing problems," *European Journal of Operational Research*, vol. 183, no. 3, pp. 1109–1130, 2007.

[2] J. F. Oliveira, A. M. Gomes, and J. . Ferreira, "Topos–a new constructive algorithm for nesting problems," *OR Spektrum. Quantitative Approaches in Management*, vol. 22, no. 2, pp. 263–284, 2000.

[3] J. F. Gonçalves and M. G. C. Resende, "Biased random-key genetic algorithms for combinatorial optimization," *Journal of Heuristics*, vol. 17, no. 5, pp. 487–525, 2011.

[4] A. K. Sato, T. C. Martins, and M. S. G. Tsuzuki, "An algorithm for the strip packing problem using collision free region and exact fitting placement," *CAD Computer Aided Design*, vol. 44, no. 8, pp. 766–777, 2012.

[5] R. Alvarez-Valdes, A. Martinez, and J. M. Tamarit, "A branch & bound algorithm for cutting and packing irregularly shaped pieces," *International Journal of Production Economics*, vol. 145, no. 2, pp. 463–477, 2013.

[6] B. S. Baker, E. G. Coffman, and R. L. Rivest, "Orthogonal packings in two dimensions," *SIAM Journal on Computing*, vol. 9, no. 4, pp. 846–855, 1980.

[7] M. A. Carravilla, C. Ribeiro, and J. F. Oliveira, "Solving nesting problems with non-convex polygons by constraint logic programming," *International Transactions in Operational Research*, vol. 10, no. 6, pp. 651–663, 2003.

[8] M. Fischetti and I. Luzzi, "Mixed-integer programming models for nesting problems," *Journal of Heuristics*, vol. 15, no. 3, pp. 201–226, 2009.

[9] K. K. Daniels, V. J. Milenkovic, and Z. Li, "Multiple containment methods," Tech. Rep., Center for Research in Computing Technology, Harvard University, Cambridge, UK, 1994.

[10] F. M. B. Toledo, M. A. Carravilla, C. Ribeiro, J. F. Oliveira, and A. M. Gomes, "The dotted-board model: a new MIP model for nesting irregular shapes," *International Journal of Production Economics*, vol. 145, no. 2, pp. 478–487, 2013.

[11] L. H. Cherri, M. A. Carravilla, and F. M. B. Toledo, "A model-based heuristic for the irregular strip packing problem," *Pesquisa Operacional*, vol. 36, no. 3, pp. 447–468, 2016.

[12] C. Ribeiro and M. A. Carravilla, "A global constraint for nesting problems," *Artificial Intelligence Review*, vol. 30, no. 1-4, pp. 99–118, 2008.

[13] B. K. Nielsen and A. Odgaard, "Fast neighborhood search for the nesting problem," *Sci. Report, University of Copenhagen*, 2003.

[14] K. A. Dowsland and W. B. Dowsland, "Solution approaches to irregular nesting problems," *European Journal of Operational Research*, vol. 84, no. 3, pp. 506–521, 1995.

[15] B. A. Junior, P. R. Pinheiro, and R. D. Saraiva, "A hybrid methodology for nesting irregular shapes: case study on a textile industry?" in *Proceedings of the 6th IFAC/ACM Conference on Management and Control of Production and Logistics (MCPL '13)*, pp. 15–20, 2013.

[16] S. Jakobs, "On genetic algorithms for the packing of polygons," *European Journal of Operational Research*, vol. 88, no. 1, pp. 165–181, 1996.

[17] B. Amaro Jr., P. R. Pinheiro, and R. D. Saraiva, "A hybrid methodology for tackling the irregular strip packing problem," in *Proceedings of the 11th IFAC Workshop on Intelligent Manufacturing Systems (IMS '13)*, 11, pp. 396–401, May 2013.

[18] B. A. Junior, P. R. Pinheiro, R. D. Saraiva, and P. G. Dantas Pinheiro, "Dealing with nonregular shapes packing," *Mathematical Problems in Engineering*, Article ID 548957, 10 pages, 2014.

[19] A. Ramesh Babu and N. Ramesh Babu, "A generic approach for nesting of 2-D parts in 2-D sheets using genetic and heuristic algorithms," *CAD Computer Aided Design*, vol. 33, no. 12, pp. 879–891, 2001.

[20] A. M. Gomes and J. F. Oliveira, "A 2-exchange heuristic for nesting problems," *European Journal of Operational Research*, vol. 141, no. 2, pp. 359–370, 2002.

[21] E. Burke, R. Hellier, G. Kendall, and G. Whitwell, "A new bottom-left-fill heuristic algorithm for the two-dimensional irregular packing problem," *Operations Research*, vol. 54, no. 3, pp. 587–601, 2006.

[22] T. Imamichi, M. Yagiura, and H. Nagamochi, "An iterated local search algorithm based on nonlinear programming for the irregular strip packing problem," *Discrete Optimization*, vol. 6, no. 4, pp. 345–361, 2009.

[23] S. C. H. Leung, Y. Lin, and D. Zhang, "Extended local search algorithm based on nonlinear programming for two-dimensional irregular strip packing problem," *Computers and Operations Research*, vol. 39, no. 3, pp. 678–686, 2012.

[24] J. Egeblad, B. K. Nielsen, and A. Odgaard, "Fast neighborhood search for two- and three-dimensional nesting problems," *European Journal of Operational Research*, vol. 183, no. 3, pp. 1249–1266, 2007.

[25] A. M. Gomes and J. F. Oliveira, "Solving irregular strip packing problems by hybridising simulated annealing and linear programming," *European Journal of Operational Research*, vol. 171, no. 3, pp. 811–829, 2006.

[26] B. A. Júnior and P. R. Pinheiro, "Approaches to tackle the nesting problems," *Advances in Intelligent Systems and Computing*, vol. 464, pp. 285–295, 2016.

[27] Y. Stoyan and L. Ponomarenko, "Minkowski sum and hodograph of the dense placement vector function," *Reports of the SSR Academy of Science, SER. A*, vol. 10, 1977.

[28] P. Rocha, R. Rodrigues, F. M. B. Toledo, and A. M. Gomes, "Circle covering using medial axis," in *Proceedings of the 11th IFAC Workshop on Intelligent Manufacturing Systems (IMS '13)*, pp. 402–407, 2013.

[29] R. C. Art, "An approach to the two-dimensional irregular cutting stock problem," Technical Report 36.008, IBM Cambridge Centre, 1966.

[30] M. Adamowicz and A. Albano, "A Solution of the Rectangular Cutting-Stock Problem," *IEEE Trans. Systems, Man, Cybernetics*, vol. SMC-6, pp. 302–310, 1976.

[31] J. A. Bennell and X. Song, "A beam search implementation for the irregular shape packing problem," *Journal of Heuristics*, vol. 16, no. 2, pp. 167–188, 2010.

[32] E. Flato and Dan Halperin, "Robust and efficient construction of planar Minkowski sums," in *In Abstracts 16th European Workshop Comput. Geom*, pp. 85–88, Ben-Gurion University of the Negev, 2000.

[33] P. K. Agarwal, E. Flato, and D. Halperin, "Polygon decomposition for efficient construction of Minkowski sums," *Computational Geometry: Theory and Applications*, vol. 21, no. 1-2, pp. 39–61, 2002.

[34] D. H. Greene, "Computational Geometry," in *Adv. Comput. Res*, F. P. Preparata, Ed., vol. 1, pp. 235–259, JAI Press, Greenwich, 1983.

[35] J. C. Bean, "Genetic algorithms and random keys for sequencing and optimization," *ORSA Journal on Computing*, vol. 6, no. 2, pp. 154–160, 1994.

[36] W. M. Spears and K. A. De Jong, "On the virtues of parameterized uniform crossover," in *Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 230–237, 1991.

[37] P. R. Pinheiro, A. J. Saraiva, and R. A. Saraiva, "A random-key genetic algorithm for solving the nesting problem," *International Journal of Computer Integrated Manufacturing*, vol. 28, pp. 1–7, 2015.

[38] B. A. Junior, P. R. Pinheiro, and R. D. Saraiva, "Tackling the irregular strip packing problem by hybridizing genetic algorithm and bottom-left heuristic," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '13)*, pp. 3012–3018, Cancun, Mexico, June 2013.

[39] L. R. Mundim, M. Andretta, and T. A. de Queiroz, "A biased random key genetic algorithm for open dimension nesting problems using no-fit raster," *Expert Systems with Applications*, vol. 81, pp. 358–371, 2017.

[40] ESICUP, 2016, http://paginas.fe.up.pt/~esicup/.

[41] A. Elkeran, "A new approach for sheet nesting problem using guided cuckoo search and pairwise clustering," *European Journal of Operational Research*, vol. 231, no. 3, pp. 757–769, 2013.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 7076583, 12 pages
https://doi.org/10.1155/2017/7076583

Research Article

Research on Optimized Torque-Distribution Control Method for Front/Rear Axle Electric Wheel Loader

Zhiyu Yang, Jixin Wang, Guangzong Gao, and Xiangyun Shi

School of Mechanical Science and Engineering, Jilin University, Changchun 130025, China

Correspondence should be addressed to Jixin Wang; jxwang@jlu.edu.cn

Received 7 April 2017; Revised 18 July 2017; Accepted 27 July 2017; Published 10 September 2017

Academic Editor: Nunzio Salerno

Optimized torque-distribution control method (OTCM) is a critical technology for front/rear axle electric wheel loader (FREWL) to improve the operation performance and energy efficiency. In the paper, a longitudinal dynamics model of FREWL is created. Based on the model, the objective functions are that the weighted sum of variance and mean of tire workload is minimal and the total motor efficiency is maximal. Four nonlinear constraint optimization algorithms, quasi-newton Lagrangian multiplier method, sequential quadratic programming, adaptive genetic algorithms, and particle swarm optimization with random weighting and natural selection, which have fast convergent rate and quick calculating speed, are used as solving solutions for objective function. The simulation results show that compared to no-control FREWL, controlled FREWL utilizes the adhesion ability better and slips less. It is obvious that controlled FREWL gains better operation performance and higher energy efficiency. The energy efficiency of FREWL in equipment transferring condition is increased by 13–29%. In addition, this paper discussed the applicability of OTCM and analyzed the reason for different simulation results of four algorithms.

1. Introduction

Hybrid wheel loader has raised much attention due to its green technology. It is considered to be the trend of future in the loader field [1–3]. Here are some released hybrid loader prototypes of several manufacturers, as shown in Table 1.

However, the energy-saving method of these above loaders is energy management strategy [4–6]. Besides, its dynamic performance, passing performance, and operation efficiency have no obvious difference with conventional diesel driven wheel loader. Hence, optimized torque-distribution control method (OTCM) of front/rear drive axle or four wheels is essential to improve operation efficiency, providing a new energy-saving strategy [7, 8]. Considering the cost and control technology, the configuration of front/rear axle independent drive is more possible to realize mass production than four-wheel drive, like mass-produced electric vehicles Tesla Model S [9] and BYD QIN [10].

There are many technologies demanding prompt solution about OTCM for FREWL. Enlightened by the relative research in on-road vehicle field, tire energy dissipation [11], total motor efficiency [12], and motor power loss [13] are used as energy efficiency optimization objective. Tire workload reflects the utilization of road adhesion ability [14]. Through the control of tire workload, the operation performance of FREWL is obviously improved.

In this paper, the proposed OTCM for FREWL is to gain better operation performance and higher energy efficiency. In the primary stage of FREWL dynamics research, it is more urgent to study longitudinal dynamics than to study the lateral stability because FREWL is often operated in low speed. This paper assumes that FREWL only moves in the longitudinal direction. In Section 2, the dynamic model is created based on the configuration of FREWL. In Section 3, depending on the target to improve the operation performance and energy efficiency, objective functions are that the weighted sum of variance and mean of tire workload is minimal and the total motor efficiency is maximal. Then constraint conditions of the optimization control are listed. Four nonlinear optimization algorithms with constraints, quasi-newton Lagrangian multiplier method (QNLM), sequential quadratic programming (SQP) [14, 15] adaptive genetic algorithms (AGA) [16, 17], and particle swarm optimization with random weighting and natural selection (PSO-RN) [18, 19],

Table 1: Outline of several prototypes.

Manufacturer	Powertrain configuration	Energy storage devices	Energy saving	Ref
Hitachi	Series	Battery	25%–30%	[20–23]
John Deere	Series	Battery	25%	[24]
Joy Global	—	Flywheel	45%	[25]
Volvo	—	Battery	—	[8]
XCMG	Parallel	Hydraulic accumulator	54%	[26]
Liu Gong	Series-parallel	Supercapacitor	—	[27]

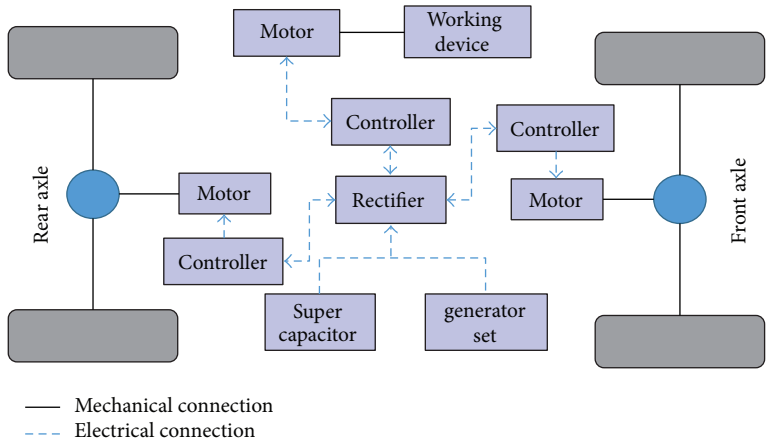


Figure 1: Transmission configuration of FREWL.

are introduced to solve the objective function. In Section 4, the effectiveness of OTCM is verified by simulation analysis. In Section 5, we discussed the applicability of OTCM and analyzed the reason for different simulation results of four algorithms.

2. Dynamical Model of FREWL

The distinctive transmission configuration of FREWL takes diesel generating set as main power source. Rectifier converts the alternating current generated by diesel generating set to a direct current which used to drive front motor, rear motor, and working motor. Supercapacitor is used as auxiliary source to effectively use braking energy and control diesel generating set in its high efficiency operating region. So diesel generating set can always operate in its high efficiency region. The transmission configuration of FREWL is shown in Figure 1.

A brief summary of the forces and torques in longitudinal dynamics is shown in Figure 2.

2.1. Wheel Vertical Load. Because FREWL operates in low speed, the influence of air resistance can be ignored. The wheel vertical load is, respectively, given by

$$F_{zf} = \frac{m\dot{v}_x h - mgh\sin\alpha + mgl_r\cos\alpha + F_z l_z}{l_f + l_r}$$
$$F_{zr} = \frac{-m\dot{v}_x h + mgh\sin\alpha + mgl_f\cos\alpha - F_z l_c}{l_f + l_r}, \quad (1)$$

where l_f is the distance from FREWL gravity center to front axle, l_r is the distance from FREWL gravity center to rear axle, l_c is the distance from the front axle to the tooth tip of bucket, $l_z = l_c + l_f + l_r$, h is the height of FREWL gravity center, m is the mass, α is the gradient of the slope, $\dot{v}_x$ is the longitudinal acceleration, F_{zf} and F_{zr} are the vertical loads of front and rear wheels, F_z is the vertical component of spading resistance on the tooth tip of bucket, and F_x is the horizontal component of spading resistance on the tooth tip of bucket.

2.2. Tire Driving Torque and Wheel Longitudinal Force. The relationship between the longitudinal force and driving torque on each tire is given by

$$I_e\dot{\omega}_{wi} = T_i - r_{\text{eff}}F_{xi}, \quad (2)$$

where I_e is the wheel rotational inertia, r_{eff} is the tire rolling radius, $\dot{\omega}_{wi}$ is the wheel angular acceleration, F_{xi} is the wheel longitudinal force, and T_i is the tire driving torque. i in subscript denotes the front or rear axle.

2.3. Motor Driving Torque. Suppose that longitudinal force and vertical load of the wheels in the same axle are equal; the relationship between the motor driving torque and tire driving torque is given by

$$T_{mi} = \frac{2T_i}{N} + I_m\dot{\omega}_{mi}, \quad (3)$$

where T_{mi} is the motor driving torque, I_m is the motor rotational inertia, $\dot{\omega}_{mi}$ is the motor angular acceleration, and N is the reduction ratio from the motor to the wheels.

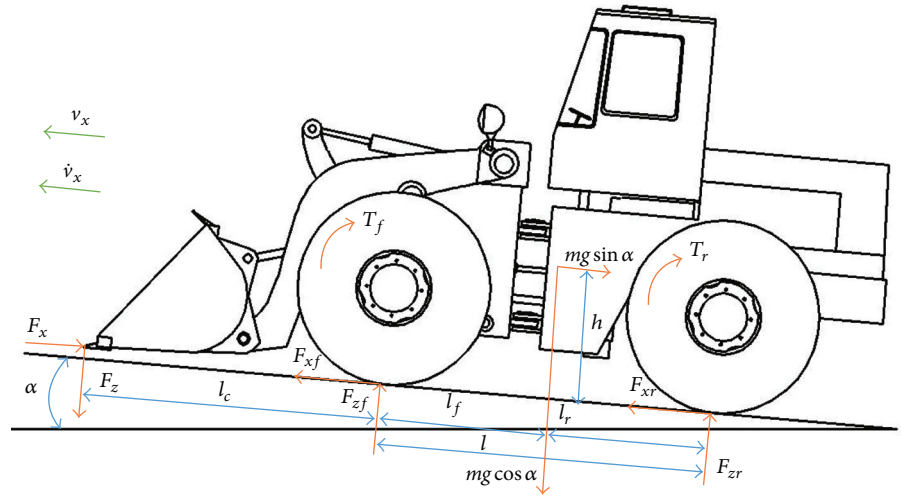


Figure 2: Illustration of FREWL forces and torques.

3. Optimal Torque-Distribution Control Method

OTCM consists of objective function, constraint conditions, and optimization algorithm.

3.1. Objective Function

3.1.1. Optimized Torque-Distribution Control Based on Tire Workload. The nonlinear optimization problem about enhancing operating performance can be formulated in this way. Because of the characteristic that the driving torque of both motors can be controlled online, the objective function is that the weighted sum of variance and mean of tire workload is minimal [5]. It can be defined as

$$\begin{aligned} \min \quad J &= \operatorname{var}(\gamma_i) + \varepsilon_v E(\gamma_i) \\ &= \frac{1}{4}\sum_{i=1}^{4}(\gamma_i - E(\gamma_i))^2 + \varepsilon_v E(\gamma_i). \end{aligned} \tag{4}$$

From (2) and (3), tire workload γ_i of each wheel is defined as

$$\gamma_i = \frac{T_{mi}^2 N^2}{4r_{\text{eff}}{}^2 \mu_i{}^2 F_{zi}{}^2}, \tag{5}$$

where μ_i is the tire-road friction coefficient of each wheel and N is the reduction ratio. By the characteristic that the driving torque of both front motor and rear motor can be controlled online, the objective function can be set by the minimum of weighted sum of variance and mean of tire workload. Distribution coefficient κ is defined as

$$\kappa = \frac{T_{mf}}{T_{mf} + T_{mr}} = \frac{T_{mf}}{T_m}, \tag{6}$$

where T_m is the sum of driving torque of the front motor and rear motor, T_{mf} is the driving torque of the front motor, and T_{mr} is the driving torque of the rear motor. Tire workload of each wheel is defined as

$$\begin{aligned} \gamma_{fl} = \gamma_{fr} &= \frac{\kappa^2 T_m^2 N^2}{4r_{\text{eff}}{}^2 \mu_f{}^2 F_{zf}{}^2} \\ \gamma_{rl} = \gamma_{rr} &= \frac{(1-\kappa)^2 T_m^2 N^2}{4r_{\text{eff}}{}^2 \mu_r{}^2 F_{zr}{}^2}, \end{aligned} \tag{7}$$

where γ_{fl} is the tire workload of left-front wheel, γ_{fr} is the tire workload of the right-front wheel, γ_{rl} is the tire workload of left-rear wheel, and γ_{rr} is the tire workload of right-rear wheel.

3.1.2. Optimized Torque-Distribution Control Based on Total Motor Efficiency. In order to improve the energy efficiency of the FREWL while transferring equipment, the objective function is that the total motor efficiency is maximal [12]. It can be defined as

$$\max \quad \eta = \frac{T_m n}{2\left[T_{mf}/\eta_f\left(T_{mf}, n_f\right) + T_{mr}/\eta_r\left(T_{mr}, n_r\right)\right] n}, \tag{8}$$

where η_f is the efficiency of front motor, η_r is the efficiency of rear motor, n_f is the front motor speed, and n_r is the rear motor speed. The relationship between motor efficiency η_f, motor torque T_f, and motor speed n_f is shown in Figure 3.

With (6), then (8) becomes

$$\max \quad \eta = \frac{1}{\kappa/\eta_f\left(T_{mf}, n_f\right) + (1-\kappa)/\eta_r\left(T_{mr}, n_r\right)}. \tag{9}$$

3.2. Constraint Conditions. The total driving torque should satisfy expected accelerator position firstly, as is shown in

$$T_m = f\left(\partial_{\text{pedal}}, n\right), \tag{10}$$

where ∂_{pedal} is the accelerator position and n is the motor speed. The surface about the three parameters is shown as Figure 4.

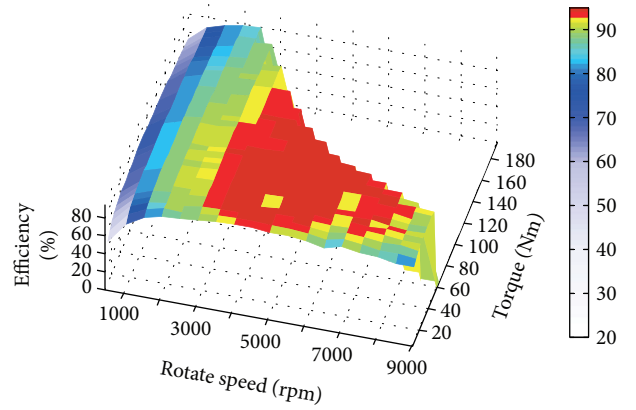


Figure 3: MAP diagram of motor.

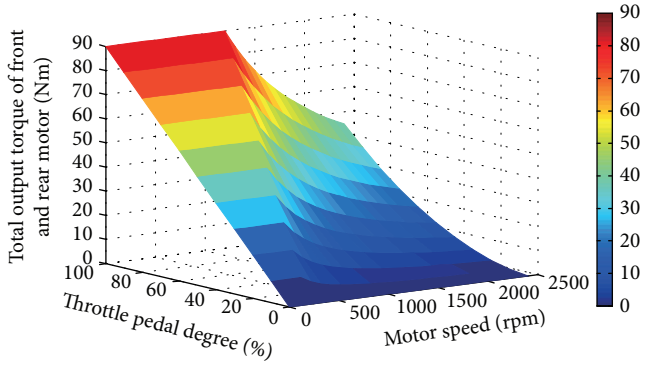


Figure 4: Defining surface of total driving torque of front/rear motor.

Adhesion force is influenced by wheel vertical load and tire-road friction coefficient. Because the influence of the motor inertia moment and wheel inertia moment is tiny, it is appropriate to ignore them. So in pure longitudinal slip condition, the maximum driving torque is limited by

$$|T_{mi}| \le \frac{2\mu_{i,x} F_{i,z} r_{\text{eff}}}{N}. \tag{11}$$

The motor driving torque and speed should be limited as follows:

$$\begin{aligned} |T_{mi}| &\le T_{\max} \\ n &\le n_{\max}. \end{aligned} \tag{12}$$

The range of driving torque-distribution coefficient κ is given by

$$0 \le \kappa \le 1. \tag{13}$$

3.3. Optimization Algorithms. Nonlinear optimization algorithms are widely used to solve nonlinear optimization problems [28]. Each nonlinear optimization algorithm has its capabilities and limitations, which have a significant impact on the performance of OTCM. The common optimization algorithms for nonlinear constraints optimization problems are quasi-Newton Lagrangian multiplier method (QNLM), sequential quadratic programming (SQP), adaptive genetic algorithms (AGA), and particle swarm optimization with random weighting and natural selection, (PSO-RN). The notable advantage that these four algorithms possess over their classic one is the fast solution speed, which satisfies the requirement of subsequent field test of FREWL.

3.3.1. Quasi-Newton Lagrangian Multiplier Method. Quasi-Newton method (QNM) is a special case of Newton method. The objective function is Taylor expanded in second order at x_{k+1}, as shown in (14).

$$\begin{aligned} f(x) &\approx f(x_{k+1}) + g_{k+1}^T (x - x_{k+1}) \\ &\quad + \frac{1}{2} (x - x_{k+1})^T G_{k+1} (x - x_{k+1}). \end{aligned} \tag{14}$$

The derivative of (14) is

$$g(x) \approx g_{k+1} + G_{k+1} (x - x_{k+1}). \tag{15}$$

An approximate matrix B_k is used to replace G_{k+1} in (15). The Broyden-Fletcher-Goldfarb-Shanno (BFGS) is used to update the B_k, as shown in (16).

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}. \tag{16}$$

Quasi-Newton method fails to solve the nonlinear constraints optimization problems. Thus Lagrangian multiplier (LM) is introduced into the problem. Inequality constraints are transformed into equality constraints by an auxiliary variable. Then with the original equality constraints, the Lagrangian function is transformed into

$$\begin{aligned} &\psi(x, y, \lambda, \sigma) \\ &= f(x) - \sum_{i=1}^{l} \mu_i h_i(x) + \frac{\sigma}{2} \sum_{i=1}^{l} h_i^2(x) \\ &\quad + \frac{1}{2\sigma} \sum_{i=1}^{m} \left(\left[\min \{0, \sigma g_i(x) - \lambda_i\} \right]^2 - \lambda_i^2 \right), \end{aligned} \tag{17}$$

where the iterative equation of the multiplier is

$$\begin{aligned} (\mu_{k+1})_i &= (\mu_k)_i - \sigma h_i(x_k), \quad i = 1, 2, \ldots, l, \\ (\lambda_{k+1})_i &= \max \{0, (\lambda_k)_i - \sigma g_i(x_k)\}, \quad i = 1, 2, \ldots, m. \end{aligned} \tag{18}$$

The termination criterion of iteration is

$$\left(\sum_{i=1}^{l} h_i^2(x_k) + \sum_{i=1}^{m} \left[\min \left\{ g_i(x_k), \frac{(\lambda_k)_i}{\sigma} \right\} \right]^2 \right)^{1/2} \le \varepsilon, \tag{19}$$

where λ is called a multiplier, σ is penalty factor, and ε is termination error ranging from 0 to 1.

The simulation parameters of the QNLM used in this paper are shown in Table 2. QNLM consists of QNM and LM. QNM and LM have their own maximum iteration number and termination error. The maximum iteration and termination error have influence on the computing accuracy and computing time. Penalty factor σ is used to penalize those individuals which do not satisfy the constraint condition for solving constrained optimization problems.

Table 2: Simulation parameters of the QNLM.

Parameters	Values
Maximum iteration number of QNM	200
Maximum iteration number of LM	200
Penalty factor of LM	5
Termination error of QNM	$1e-5$
Termination error of LM	$1e-5$

3.3.2. Sequential Quadratic Programming. SQP is an efficient method for nonlinear optimization with advantages of high computational efficiency and fast convergent rate. In SQP, positive definite matrix $B_0 \in R^{n\times n}$ is used to seek the optimal solution d_k of subquestion. The subquestion is described as

$$\begin{aligned} \min \quad & \frac{1}{2}d^T B_k d + \nabla f\left(x_k\right)^T d, \\ \text{s.t.} \quad & h\left(x_k\right) + \nabla h\left(x_k\right)^T d = 0, \\ & g\left(x_k\right) + \nabla g\left(x_k\right)^T d \geq 0. \end{aligned} \tag{20}$$

If the constraints $\|d_k\|_1 \leq \varepsilon_1$ and $\|h_k\|_1 + \|(g_k)_-\|_1 \leq \varepsilon_2$ are satisfied, the calculation will be determined. $[g_k(x)]_- = \max\{0, -g_k(x)\}$, and 1 means the initial value. A point under Karush–Kuhn–Tucker (KKT) condition is obtained and the termination error is $0 \leq \varepsilon_1, \varepsilon_2 \leq 1$.

To some value function $\phi(x, \sigma)$, the chosen penalty factor σ defines that d_k is decreasing at the point x_k. Suppose that m_k is the tiniest nonnegative integer satisfying the following equation:

$$\phi\left(x_k + \rho^m d_k, \sigma_k\right) - \phi\left(x_k, \sigma_k\right) \leq \eta\rho^m \phi'\left(x_k, \sigma_k; d_k\right), \tag{21}$$

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

From $\alpha_k = \rho^{mk}$, $x_{k+1} = x_k + \alpha_k d_k$, A_{k+1} is given by

$$A_{k+1} = \left(\nabla h\left(x_{k+1}\right)^T, \nabla g\left(x_{k+1}\right)^T\right)^T. \tag{22}$$

The least square multiplier is calculated by

$$\begin{pmatrix} \mu_{k+1} \\ \lambda_{k+1} \end{pmatrix} = \left[A_{k+1}A_{k+1}^T\right]^{-1} A_{k+1}\nabla f\left(x_{k+1}\right). \tag{23}$$

The approximate matrix B_k is the same as in (16).

The simulation parameters of the SQP used in this paper are shown in Table 3. SQP needs to solve a quadratic programming subproblem at every iteration step. It is necessary to set its and subproblem's maximum iteration number and termination error. The values of maximum iteration number, termination error, and penalty factor are the same as the function in QNLM.

3.3.3. Adaptive Genetic Algorithms. AGA is another significant and promising variant of genetic algorithms. AGA adjusts probabilities of crossover and probabilities of mutation in order to maintain the genetic model and to accelerate the convergence speed. In AGA, the evolution usually starts

Table 3: Simulation parameters of SQP.

Parameters	Values
Maximum iteration number	200
Iteration number of subproblem	200
Penalty factor	0.05
Termination error	$1e-5$
Termination error of subproblem	$1e-5$

Table 4: Simulation parameters of the AGA.

Parameters	Values
Lower bound of independent variable	1
Upper bound of independent variable	0
Scale of population	50
Maximum evolution generations	200
Discrete precision of independent variable	$1e-5$
Crossover constant k_1	0.5
Crossover constant k_2	0.9
Mutation constant k_3	0.03
Mutation constant k_4	0.07

from a population which consisted of randomly generated individuals. By Roulette strategy, individual fitness values are evaluated to judge if it agrees with the optimization criterion. The new individuals are generated by the optimal best mutation probability P_m and the best crossover probability P_c whose equations are shown in

$$\begin{aligned} P_c &= \begin{cases} \dfrac{k_1\left(f_{\max} - f\right)}{f_{\max} - f_{\text{avg}}}, & f \geq f_{\text{avg}} \\ k_2, & f < f_{\text{avg}} \end{cases} \\ P_m &= \begin{cases} \dfrac{k_3\left(f_{\max} - f'\right)}{f_{\max} - f_{\text{avg}}}, & f' \geq f_{\text{avg}} \\ k_4, & f' < f_{\text{avg}}, \end{cases} \end{aligned} \tag{24}$$

where $f_{\max}$ is the maximum fitness value population, f_{avg} is the mean fitness value population, f is the bigger fitness value in two individuals about to crossover, and f' is the fitness value of individuals about to mutate. k_1, k_2, k_3, and k_4 are the constants.

The simulation parameters of AGA used in this paper are shown in Table 4. Among them, the lower bound and upper bound of independent variable and discrete precision determine the encoding length required for the binary encoding. The values for lower bound and upper bound depend on the constraints. The scale of population and the maximum ecology affect the accuracy and computing time. In order to analyze the advantages and disadvantages of the various algorithms as much as possible, the maximum evolution generations are the same with the maximum number of the same number of QNLM and SQP. The values of k_1, k_2, k_3, and k_4 are usually based on different application objects. It generally requires that $k_1 < k_2, k_3 < k_4$.

Table 5: Simulation parameters of the PSO-RN.

Parameters	Values
Particle population	30
Acceleration constant c_1	2
Acceleration constant c_2	2
Upper boundary of inertia weight	0.9
Lower boundary of inertia weight	0.4
Maximum evolution generations	100
Dimension of search space	1
Maximum particle velocity v_{max}	0.2
Minimum particle velocity v_{min}	0
Maximum particle position x_{max}	1
Minimum particle position x_{min}	0

3.3.4. Particle Swarm Optimization with Random Weighting and Natural Selection. Particle swarm optimization (PSO) based on natural selection is one of the improved algorithms, which is characterized by iteratively trying to improve a candidate solution. In each iteration, the worst half of the particles in the population is replaced by the best half of the particles while preserving the original historical optimal value. Therefore, it improves the optimization ability and solving speed and significantly reduces the algorithm premature convergence situation.

The inertia weight α is an important parameter in the PSO which is used to control the ability of development and search. The core of avoiding falling into the local optimal is to determine a reasonable inertia weight. In order to accelerate up the convergence speed, the inertia weight α is set as a random value. The equation to calculate the random α is

$$\begin{aligned} \omega &= \alpha + \sigma \times N(0,1) \\ \alpha &= \alpha_{min} + (\alpha_{max} - \alpha_{min}) \times \text{rand}(0,1), \end{aligned} \tag{25}$$

where σ is the random weight, α_{max} and α_{min} are the maximum and minimum of the inertia weight, and $N(0,1)$ is the random number of standard state distribution.

The simulation parameters of the PSO-RN used in this paper are shown in Table 5. Acceleration constants c_1 and c_2 determine the influence of particle individual experience and group experience on the trajectory of particle movement, usually $c_1 = c_2$. Inertia weight can be used to control the algorithm development and search capabilities, which have different values according to different application problems. Dimension of search space has the same value as the number of its independent variables. Maximum particle position x_{max} and minimum particle position x_{min} are the boundary conditions of the algorithm, and the numerical value depends on the constraints of the actual problem. Maximum particle velocity v_{max} determines the maximum distance that the particle can move in one flight, usually $v_{max} = kx_{max}$, $0.1 \leq k \leq 1$.

Table 6: Simulation parameters based on tire workload.

	Condition 1	Condition 2	Condition 3
μ_f	0.8	0.2	0.2
μ_r	0.3	0.7	0.2
T_m (Nm)	300	300	360
F_x (N)	—	600	600
α (°)	—	—	20

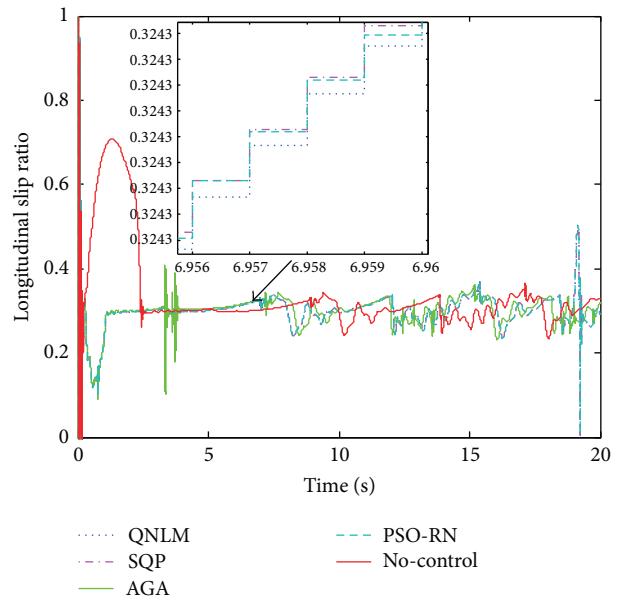


Figure 5: Longitudinal slip ratio of front wheel.

4. Simulation Results

4.1. Simulation of OTCM Based on Tire Workload. Conditions of traveling, spading, and stacking on bumpy road are common for wheel loader. The paper establishes these three conditions to verify the improvement of the FREWL operation performance through OTCM based on tire workload. Condition 1 simulates the traveling condition, while condition 2 simulates the spading condition. Condition 3 simulates the stacking condition in bumpy road with 20° slope. The simulation parameters are listed in Table 6.

From Figures 5 and 6, it is notable that in the first 2 s, namely, the starting stage, the longitudinal slip ratio of controlled FREWL is much smaller than no-control FREWL. The operation performance has significantly improved by OTCM. After 3 s, namely, the driving stage, the longitudinal ratio optimized by AGA is smoother than other three algorithms, so the operation performance of FREWL is the best.

Another parameter to evaluate the control effect is the driving distance of the controlled FREWL and no-control FREWL on bumpy road in the same time, which is shown in Figure 7. Table 7 shows the driving distance of the FREWL without spading. Controlled FREWL has better control to longitudinal slip ratio; thus it marches earlier and drives farther than no-control FREWL. The OTCM based on AGA has the best control effect, and the driving distance compared

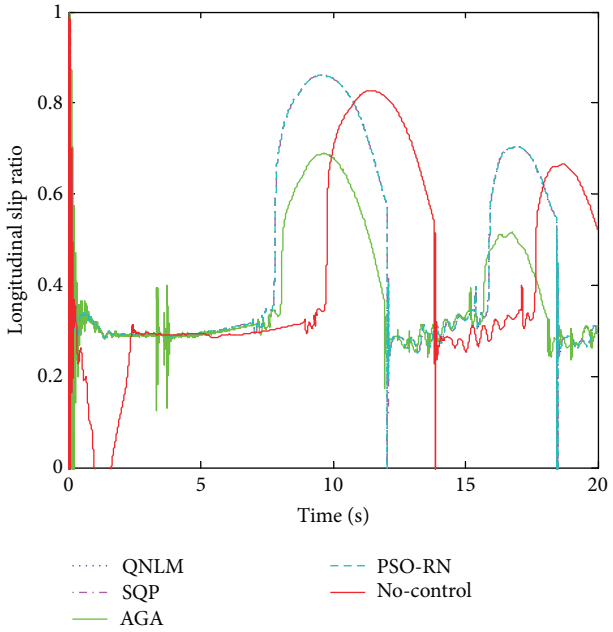


Figure 6: Longitudinal slip ratio of rear wheel.

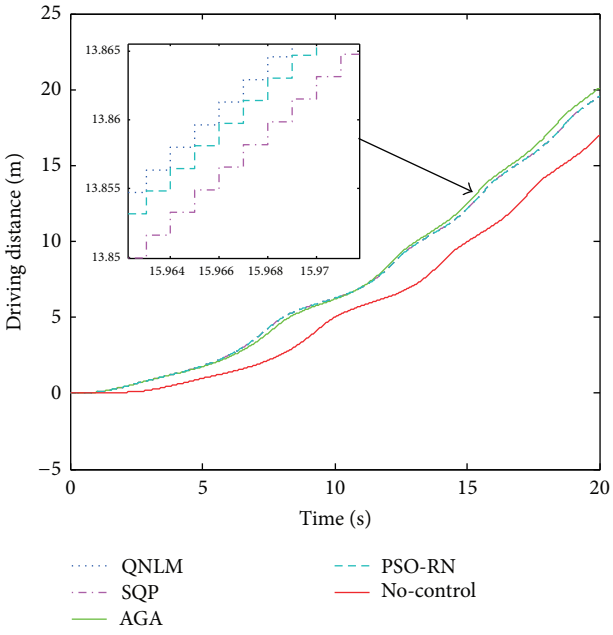


Figure 7: Driving distance in condition 1.

Table 7: Driving distance without spading resistance.

	Final distance (m)	Distance increase (%)
No-control	16.92	—
SQP	19.53	15.42
AGA	20.08	18.68
PSO-RN	19.54	15.48
QNLM	19.54	15.48

to no-control FREWL is increased by 18.68%. The control effect of other three optimization algorithms is basically the same; the driving distance is increased by 15.48% compared to the no-control FREWL.

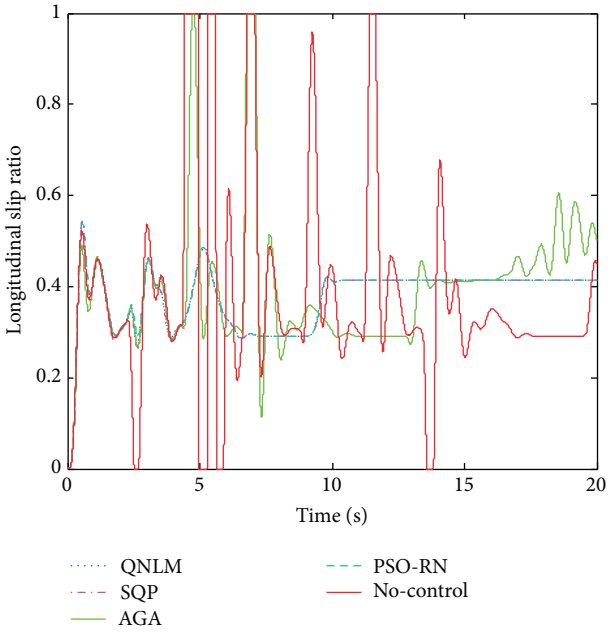


Figure 8: Longitudinal slip ratio of front wheel.

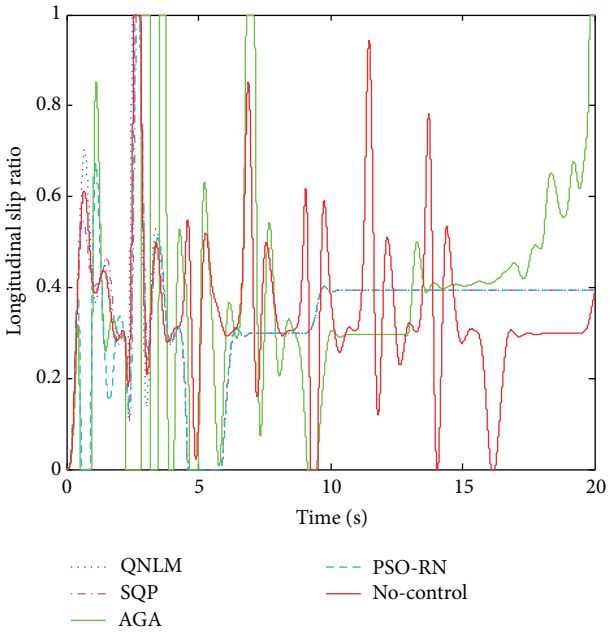


Figure 9: Longitudinal slip ratio of rear wheel.

Figures 8 and 9 show the longitudinal slip ratio of front and rear wheel in condition 2, respectively. Table 8 shows slip frequency of the front and rear wheel of the controlled FREWL and no-control FREWL. In whole simulation time, especially the starting stage, the control effect of OTCM based on SQP is the best. Only the rear wheel slips once.

Figure 10 shows the driving distance of FREWL in condition 2. Controlled FREWL utilizes adhesion ability better

Table 8: Slip frequency of the front and rear wheels.

	Front wheel slip times	Rear wheel slip times
No-control	4	1
SQP	0	1
AGA	2	4
PSO-RN	0	1
QNLM	0	1

Table 9: Forward driving time with spading resistance.

	Driving time (s)	Time decrease (%)
No-control	15.42	—
SQP	3.98	74.19
AGA	8.42	45.39
PSO-RN	3.92	74.59
QNLM	3.96	74.32

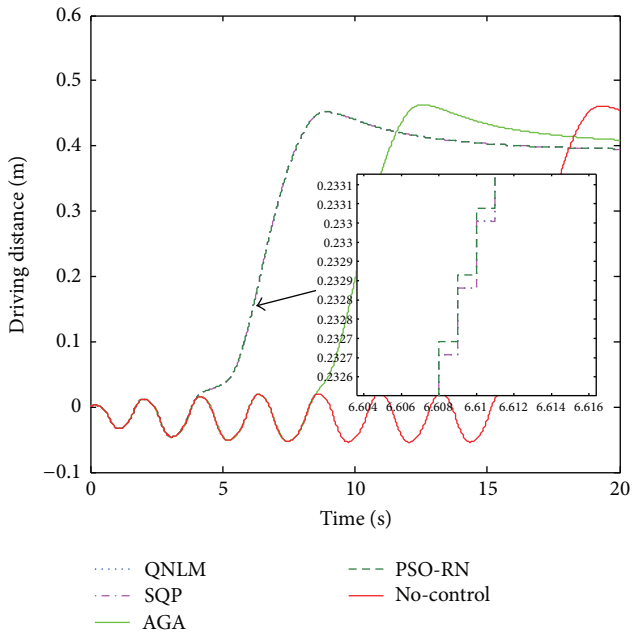


Figure 10: Driving distance in condition 2.

and slips less and also can go forward in a shorter period when encountering spading resistance, as shown in Table 9. In this case, FREWL controlled by QNLN, SQP, and PSO-RN can go forward in a shorter period and make significant improvement, but the control effect of OTCM based on AGA is undesirable.

Condition 3 simulates stacking condition that the FREWL operates on the bumpy road with 20° slope and encounters a continuous spading resistance after 5 s, and is gradually heavy-loaded. Compared to conditions 1 and 2, condition 3 is more complicated. Figures 11 and 12 show the longitudinal slip ratio of front and rear wheel in condition 3. It is obvious that after 10 s the front wheel of no-control FREWL is basically in the slippery state. While the FREWL applies four optimization algorithms, the slippery time is much less.

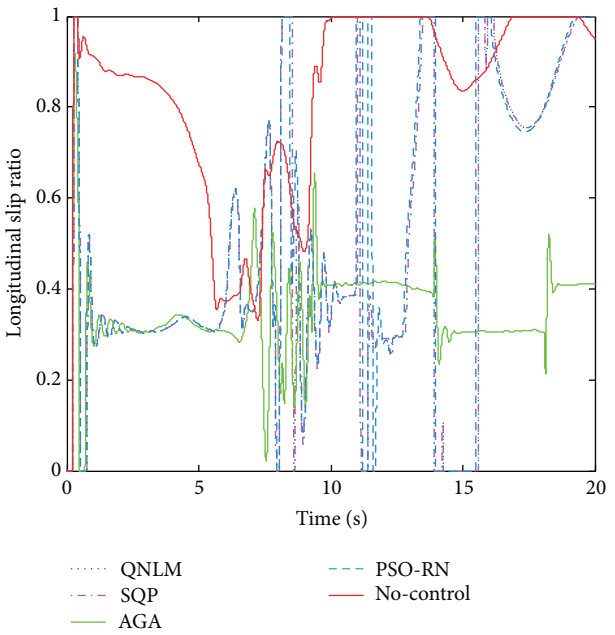


Figure 11: Longitudinal slip ratio of front wheel.

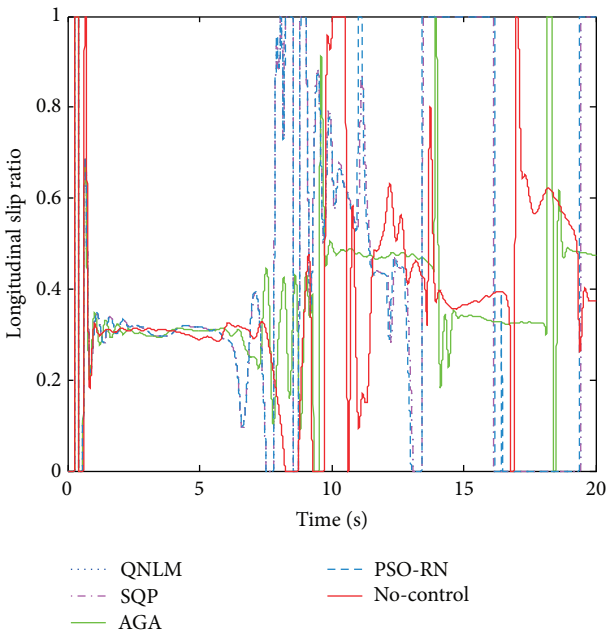


Figure 12: Longitudinal slip ratio of rear wheel.

Among them, AGA-controlled FREWL slips the least. The control effects of the other three algorithms are similar.

Figure 13 shows the driving distance of FREWL in condition 3. Since AGA is better in control of the longitudinal slip ratio of front/rear wheel in Figures 11 and 12, it drives farthest. The other three algorithms also have a good performance compared to no-control. This shows that FREWL, which applied OTCM based on tire workload, has better operation performance in stacking condition than no-control FREWL.

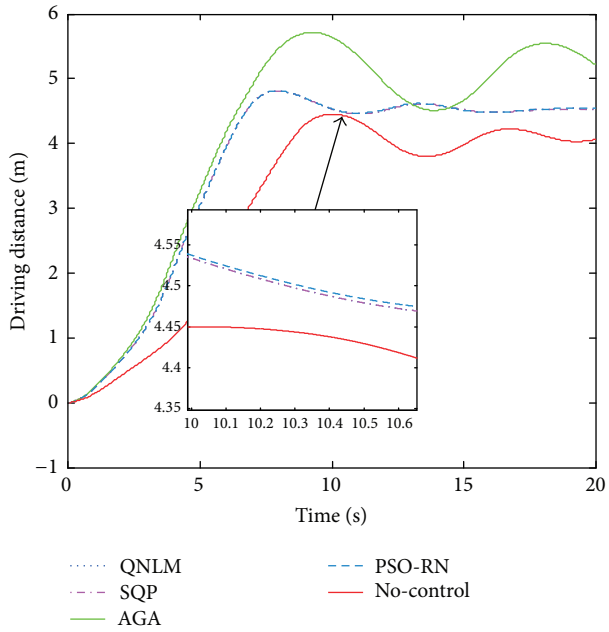


Figure 13: Driving distance in condition 3.

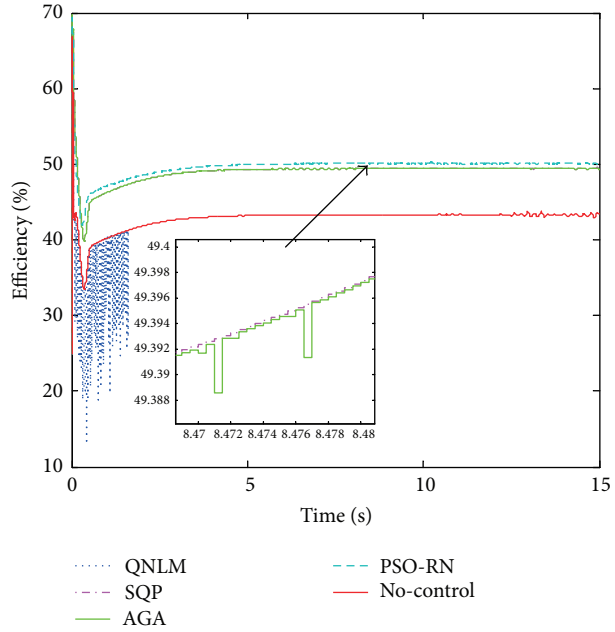


Figure 14: Total motor efficiency in condition 4.

Table 10: Simulation parameters based on total motor efficiency.

	Condition 4	Condition 5
Motion state	Straight driving	Reciprocating driving
μ_f	0.8	0.6
μ_r	0.8	0.6
v_0 (m/s)	0	10
T_m (Nm)	100	200

Table 11: Total motor efficiency in straight driving.

	Maximum total motor efficiency (%)	Efficiency increase (%)
No-control	43.63	—
SQP	49.58	13.64
AGA	49.57	13.48
PSO-RN	50.12	14.86

4.2. Optimized Torque-Distribution Control Based on Total Motor Efficiency. Two most common equipment transferring conditions are driving straightly on the bituminous road and reciprocating on bumpy road. This paper sets two conditions to simulate the improvement of energy efficiency of FREWL by OTCM while transferring equipment. The simulation parameters are listed in Table 10. Condition 4 is set to observe the energy efficiency of FREWL while driving straightly on the bituminous road. Condition 5 is set to observe the energy efficiency of FREWL while reciprocating on bumpy road.

Figure 14 shows the total motor efficiency of the FREWL when the FREWL is traveling on the bituminous road. Table 11 shows the maximum motor efficiency of the various optimization algorithms. The total motor efficiency optimized by OTCM based on QNLM fluctuates frequently and violently in the starting stage of simulation, which leads to calculation failure, which means that QNLM is not suitable for problems with strong-nonlinearity. The results of other three optimization algorithms are better than no-control FREWL. The PSO-RN works best; the total motor efficiency is increased by 14.86% comparing to no-control FREWL. The results of SPQ and AGA are roughly the same; the total motor efficiency is increased by 13.64% comparing to no-control FREWL.

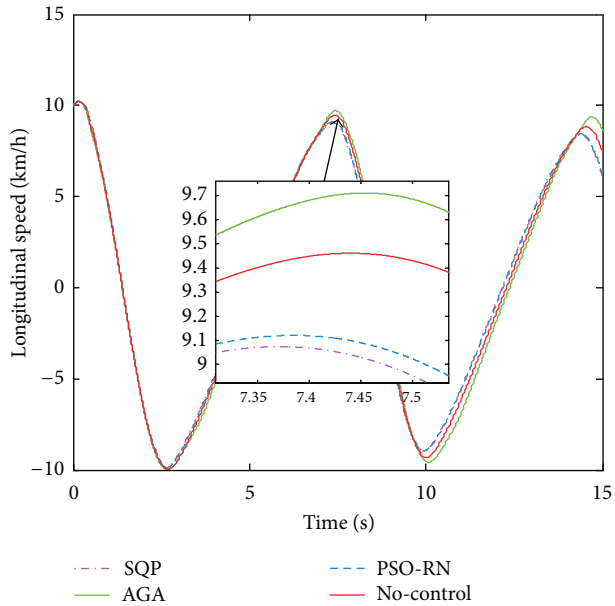


Figure 15: Longitudinal speed in condition 5.

Figures 15 and 16 show longitudinal speed and total motor efficiency of FREWL in condition 5. Table 12 shows the maximum total motor efficiency of FREWL in the forward

Table 12: Total motor efficiency in reciprocating driving.

	Forward-total motor efficiency (%)	Forward-efficiency increase (%)	Backward-total motor efficiency (%)	Backward-efficiency increase (%)
No-control	46.53	—	28.61	—
SQP	53.62	15.24	36.02	25.90
AGA	51.68	11.07	36.94	29.12
PSO-RN	52.25	12.29	34.52	20.66

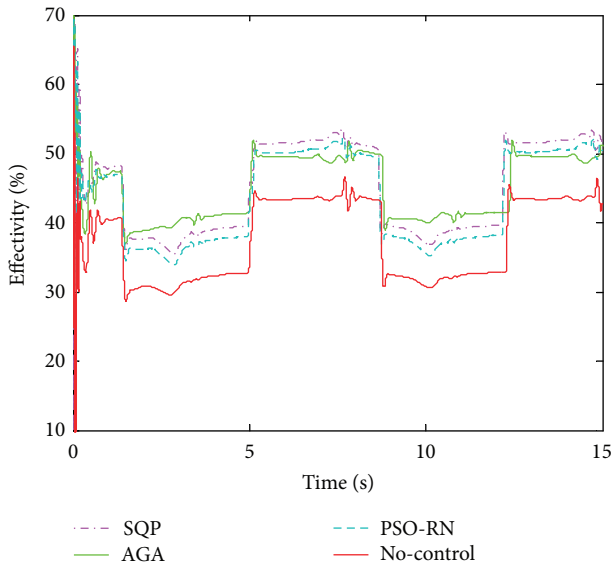


Figure 16: Total motor efficiency in condition 5.

stage and the backward stage. The total motor efficiency of controlled FREWL is greatly improved compared to no-control FREWL when driving reciprocally. The results by OTCM perform better in backward stage than in forward stage. The solution of SQP is the best in forward stage; the total motor efficiency is increased by 15.24% at most. The result of AGA works best in backward stage; the total motor efficiency is increased by 29.12% at most.

4.3. Simulation Time. Simulation time is an essential factor to affect the practicability of the method. Table 13 shows the simulation time of the optimization algorithms under various conditions. The simulation step size is 0.001 s in the Simulink/Carsim platforms. Taking into account the actual real test, the sampling step is generally set to 0.01 s. Therefore, QNLM and SQP can be used in the online control, but AGA and PSO-RN cannot. Because QNLM fails in equipment transferring condition, SQP is comprehensively the best optimization algorithm for OTCM in actual test.

5. Discussion

In this paper, we study the OTCM of FREWL and prove that OTCM can improve the operation performance and energy efficiency of FREWL through five simulation conditions. In addition to five simulation conditions mentioned in this

Table 13: Simulation time of OTCM on different conditions.

	QNLM (s)	SQP (s)	AGA (min)	PSO-RN (min)
Condition 1	32.7	55.6	69.2	27.3
Condition 2	23.5	36.1	88.3	54.8
Condition 3	32.1	41.6	64.9	29.5
Condition 4	—	17.9	52.7	15.7
Condition 5	—	13.9	54.7	18.1

Table 14: Parameters changed in other conditions.

	m	α	$\dot{v}_x$	F_x	F_z
Heavy-haul transportation	√	—	√	—	—
Loading/unloading material	√	—	√	√	√
Bulldozing	√	—	√	√	√
Climbing	—	√	√	—	—

paper, this method is also adaptable to other operation and equipment transferring conditions of FREWL in longitudinal motion. The changed parameters are shown in Table 14. Tick "√" means this parameter is changed in that situation and hyphen "—" means it is unchanged.

As can be seen from Table 14, in other conditions, all the changed parameters are taken into account in the dynamic model of FREWL. At the same time, the optimization algorithm and optimization goals do not change. Therefore, the method proposed in this paper is suitable for operation and equipment transferring conditions.

In the simulation case, compared to no-control FREWL, the FREWL controlled by four optimization algorithms have a great increase in the operation performance and energy efficiency. In the OTCM based on tire workload, the QNLM and SQP optimization solutions are almost identical, because their core algorithm is the BFGS algorithm, as shown in (16). PSO-RN and AGA are modern intelligent optimization algorithms, but their results are quite different because the optimal solution of PSO-RN is easy to fall into the local optimal solution. Comparing to PSO-RN, AGA increases the crossover probability and mutation probability when the optimal solution tends to local optimal solution, enhancing the ability to solve the global optimal solution. Thus, AGA generally has a better performance in most simulation conditions.

However, although the AGA has a better performance, it has the longest computing time because the movement of the whole population is more evenly moving to the optimal region. QNLM and SQP compute faster because they apply

the traditional BFGS method. Among them, SQP needs to solve a quadratic programming subproblem at each iteration step, so the calculation time is longer than QNLM.

6. Conclusion

OTCM is a critical technology to improve the operation performance and energy efficiency of FREWL. The driving torque of front motor and rear motor of FREWL can be controlled independently. The objective function minimizes the weighted sum of variance and mean value of tire workload and maximizes the total motor efficiency. The results show that the operation performance and energy efficiency are obviously improved by OTCM. While the FREWL is operating, the frequency of slip is obviously reduced, and the adhesion ability is improved. While the FREWL is driving straightly in equipment transferring, total motor efficiency is improved by 14.86% at most. While the FREWL is driving reciprocally in equipment transferring, total motor efficiency is improved by 29.12% at most. Considering the simulation results and simulation time comprehensively, SQP is the most suitable one of the four optimization algorithms for field test of FREWL.

Conflicts of Interest

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

Acknowledgments

The authors acknowledge the funding support from National Natural Science Foundation of China, no. 51375202, and also acknowledge the support of Programs for Science Research and Development of Jilin Province, no. 20160101285JC.

References

[1] Z. Naiwei, D. Qunliang, J. Yuanhua, Z. Wei, and R. Youcun, "Modeling and simulation research of coaxial parallel hybrid loader," *Applied Mechanics and Materials*, vol. 29–32, pp. 1634–1639, 2010.

[2] J. Wang, Z. Yang, S. Liu, Q. Zhang, and Y. Han, "A comprehensive overview of hybrid construction machinery," *Advances in Mechanical Engineering*, vol. 8, no. 3, pp. 1–15, 2016.

[3] F. Wang, M. A. Mohd Zulkefli, Z. Sun, and K. A. Stelson, "Energy management strategy for a power-split hydraulic hybrid wheel loader," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 230, no. 8, pp. 1105–1120, 2016.

[4] P. Song, M. Tomizuka, and C. Zong, "A novel integrated chassis controller for full drive-by-wire vehicles," *Vehicle System Dynamics*, vol. 53, no. 2, pp. 215–236, 2015.

[5] Y. Dai, Y. Luo, W. Chu, and K. Li, "Optimum tyre force distribution for four-wheel-independent drive electric vehicle with active front steering," *International Journal of Vehicle Design*, vol. 65, no. 4, pp. 336–359, 2014.

[6] H. Fujimoto and K. Maeda, "Optimal yaw-rate control for electric vehicles with active front-rear steering and four-wheel driving-braking force distribution," in *Proceedings of the 39th Annual Conference of the IEEE Industrial Electronics Society (IECON '13)*, pp. 6514–6519, IEEE, Vienna, AUSTRIA, November 2013.

[7] J. Kim, "Optimal power distribution of front and rear motors for minimizing energy consumption of 4-wheel-drive electric vehicles," *International Journal of Automotive Technology*, vol. 17, no. 2, pp. 319–326, 2016.

[8] A. Pennycott, L. De Novellis, A. Sabbatini, P. Gruber, and A. Sorniotti, "Reducing the motor power losses of a four-wheel drive, fully electric vehicle via wheel torque allocation," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 228, no. 7, pp. 830–839, 2014.

[9] T. Cheong, S. H. Song, and C. Hu, "Strategic alliance with competitors in the electric vehicle market: tesla motor's case," *Mathematical Problems in Engineering*, vol. 2016, Article ID 7210767, 10 pages, 2016.

[10] H. He, J. Fan, Y. Li, and J. Li, "When to switch to a hybrid electric vehicle: a replacement optimisation decision," *Journal of Cleaner Production*, vol. 148, pp. 295–303, 2017.

[11] Y. Suzuki, Y. Kano, and M. Abe, "A study on tyre force distribution controls for full drive-by-wire electric vehicle," *Vehicle System Dynamics*, vol. 52, no. 1, pp. 235–250, 2014.

[12] Z.-P. Yu, L.-J. Zhang, and L. Xiong, "Optimized torque distribution control to achieve higher fuel economy of 4WD electric vehicle with four in-wheel motors," *Journal of Tongji University*, vol. 33, no. 10, pp. 79–85, 2005.

[13] D. Wang and B. Wang, "Research on driving force optimal distribution and fuzzy decision control system for a dual-motor electric vehicle," in *Proceedings of the 34th Chinese Control Conference (CCC '15)*, pp. 8146–8153, IEEE, Hangzhou, China, July 2015.

[14] A. F. Izmailov, M. V. Solodov, and E. I. Uskov, "Globalizing stabilized sequential quadratic programming method by smooth primal-dual exact penalty function," *Journal of Optimization Theory and Applications*, vol. 169, no. 1, pp. 148–178, 2016.

[15] M. A. Z. Raja, A. Zameer, A. U. Khan, and A. M. Wazwaz, "A new numerical approach to solve Thomas–Fermi model of an atom using bio-inspired heuristics integrated with sequential quadratic programming," *SpringerPlus*, vol. 5, no. 1, article 1400, 2016.

[16] J. Apolinar and M. Rodríguez, "Three-dimensional microscope vision system based on micro laser line scanning and adaptive genetic algorithms," *Optics Communications*, vol. 385, pp. 1–8, 2017.

[17] B. Bao, Y. Yang, Q. Chen, A. Liu, and J. Zhao, "Task allocation optimization in collaborative customized product development based on double-population adaptive genetic algorithm," *Journal of Intelligent Manufacturing*, vol. 27, no. 5, pp. 1097–1110, 2016.

[18] Y. Chen and M. S. Li, "A harmonic parameter estimation method based on Particle Swarm Optimizer with Natural Selection," in *Proceedings of the 1st International Conference on Information and Communication Technology Research (ICTRC '15)*, pp. 206–209, IEEE, Abu Dhabi, Arab Emipates, May 2015.

[19] Y. Hu, J. Lu, Y. Deng, and Z. Zhang, "MPPT algorithm based on particle swarm optimization with natural selection," in *Proceedings of the 3rd International Conference on Machinery, Materials and Information Technology Applications (ICMMITA '15)*, pp. 1814–1817, Atlantis Press, Qingdao, China, November 2015.

[20] H. Inoue, "Introduction of PC200-8 hybrid hydraulic excavators," *Report, Komatsu*, vol. 54, no. 161, pp. 1–6, 2008.

[21] M. Ochiai and S. Ryu, "Hybrid in construction machinery," in *Proceedings of the 7th JFPS international symposium on fluid power*, pp. 41–44, Toyama, Japan, September 2008.

[22] M. Ochiai, "Technical trend and problem in construction machinery," *Construction Machinery*, vol. 38, no. 4, pp. 20–24, 2002.

[23] S. Riyuu, M. Tamura, and M. Ochiai, "Hybrid construction machine," Patent 2003328397, Japan, 2003.

[24] J. Deere, "644K Hybrid Wheel Loader," 2017, http://www.deere.com/wps/dcom/en_US/products/equipment/wheel_loaders/644k_hybrid/644k_hybrid.page.

[25] J. Global, "L-950 Wheel Loader," 2017, https://mining.komatsu/product-details/l-950#!technology.

[26] S. Hui and J. Junqing, "Research on the system configuration and energy control strategy for parallel hydraulic hybrid loader," *Automation in Construction*, vol. 19, no. 2, pp. 213–220, 2010.

[27] N. Zhou, E. Zhang, Q. Wang et al., "Compound hybrid wheel loader," Patent CN102653228A, China, 2012.

[28] M. P. Aghababa, "3D path planning for underwater vehicles using five evolutionary optimization algorithms avoiding static and energetic obstacles," *Applied Ocean Research*, vol. 38, pp. 48–62, 2012.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 4703106, 11 pages
https://doi.org/10.1155/2017/4703106

Research Article

A Simulated Annealing Approach for the Train Design Optimization Problem

Federico Alonso-Pecina[1] and David Romero[2]

[1]*Universidad Autónoma del Estado de Morelos, 62209 Cuernavaca, MOR, Mexico*
[2]*Instituto de Matemáticas, Universidad Nacional Autónoma de México, 62210 Cuernavaca, MOR, Mexico*

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

Received 10 March 2017; Revised 20 June 2017; Accepted 9 July 2017; Published 10 August 2017

Academic Editor: Jorge Magalhaes-Mendes

The Train Design Optimization Problem regards making optimal decisions on the number and movement of locomotives and crews through a railway network, so as to satisfy requested pick-up and delivery of car blocks at stations. In a mathematical programming formulation, the objective function to minimize is composed of the costs associated with the movement of locomotives and cars, the loading/unloading operations, the number of locomotives, and the crews' return to their departure stations. The constraints include upper bounds for number of car blocks per locomotive, number of car block swaps, and number of locomotives passing through railroad segments. We propose here a heuristic method to solve this highly combinatorial problem in two steps. The first one finds an initial, feasible solution by means of an ad hoc algorithm. The second step uses the simulated annealing concept to improve the initial solution, followed by a procedure aiming to further reduce the number of needed locomotives. We show that our results are competitive with those found in the literature.

1. Introduction

An ample range of investigations have been undertaken to develop optimization algorithms for various problems encountered in the realm of rail systems, like routing [1], scheduling [2–4], crew assignment [5, 6], and blocking [7, 8], among others [9–12]. However, to our knowledge, the Train Design Optimization Problem as defined below is the only attempt to simultaneously deal with block-to-train assignment, train routing, and crew assignment [13], arising as one of the most fundamental and difficult combinatorial optimization problems formulated in the railroad industry [13–17], with a huge potential to benefit from the application of operations research techniques.

In rail systems, a railroad car, railcar, or train car (*car*, for short), is a vehicle used for the carrying of cargo. Such cars, when coupled together and hauled by one or more locomotives, form a *train*. A car block (*block*, for short) is a semipermanently arranged formation of cars. Trains are then built of one or more blocks coupled together as needed. Also, in the operation of rail systems a *block swap* can occur, namely, when a locomotive delivers a block at a station distinct from the block's destination, and another locomotive picks up the block afterwards. Each time a train stops en route at a station to pick up and/or to deliver blocks, a *work-event* takes place.

In this paper we address the Train Design Optimization Problem (TDOP) arising in the operation of railroad freight transport as part of the logistics chain. It consists in determining, at minimal total cost, and subject to capacity and operational constraints, the number of locomotives and crews, together with the logistics related to the movement of locomotives, blocks, and crews through a railway network, so as to transport goods from a set of shippers to a set of destinations.

The total cost depends on the number of assigned locomotives, the distance traveled by locomotives and cars, the distance traveled by crews to return to their departure stations, the number of block swaps and work-events, the number of blocks not arriving to destination, and the difference between the number of locomotives arriving to and departing from stations. The constraints include maximum number of

blocks per locomotive, maximum number of block swaps, maximum number of work-events per locomotive, maximum length, weight, and number of trains passing through a railroad segment, and crew limitations as to the route to follow.

Only a handful of approaches have been developed for the solution of the TDOP. In [18] a mixed integer programming model is proposed; block routes are first generated together with train routes to cover them, then a matching is sought to minimize the objective function, and finally several greedy and local search rules are iteratively invoked to update the block-paths and train routes so as to improve the solution quality. In [19, 20] the problem is formulated as one of integer programming where the number of variables and constraints is exponential, proposing for its solution a column-row generation heuristic followed by clever tabu search methods. An iterative procedure is suggested in [21] to solve two subproblems of the TDOP: train design and block-to-train assignment, where the former consists in determining train routes to be operated, and the latter deals with the routing decision for blocks; both subproblems are solved by integer programming techniques. In [22, 23] a column-generation approach is designed: first, a set of promising train routes is generated based on the crew segments; then an integer linear programming model is developed for the subsequent decisions including train route selection and block-to-train assignment.

We propose hereafter a method to heuristically solve the TDOP in two steps. By means of an ad hoc procedure the first step aims to produce an initial feasible solution, namely, a solution satisfying all the constraints. The second step uses the simulated annealing method to improve the initial solution, followed by simple, specialized procedures that attempt to reduce the number of needed trains without increasing the overall cost. To test our proposal we solved the only three published instances to date, finding it competitive with other approaches in the literature. Also, we randomly generated 20 synthetic instances as well as (improvable) lower bounds for their corresponding optimal solutions; our heuristic came up with results averaging an error close to 25% on these bounds.

The rest of the paper is organized as follows: Section 2 provides the necessary terminology and makes a detailed description of constraints and objective function of the TDOP; a toy example is also furnished to help understand the problem. In Section 3 the TDOP is formulated in mathematical programming terms. Our solution approach—basically combining an ad hoc algorithm to find an initial feasible solution followed by the metaheuristic known as simulated annealing—is explained in Section 4. Further, Section 5 is devoted to the computational experiments that we conducted to test our procedures; for this experimentation we used the three available known instances as well as a new set of 20 randomly generated ones. Finally, Section 6 presents our conclusions and a proposal for future work.

2. Problem Description

This section is aimed to describe the TDOP, borrowing some terminology from [20]. An alternative description can be found in [13]. The TDOP terminology, constraints, and objective function are provided in Section 2.1. To help in grasping our description, an instance of the TDOP is presented in Example 1 by means of a toy example, followed by one of its feasible solutions in Section 2.2.

2.1. Terminology, Constraints, and Objective Function

Terminology

(i) A *block* is a formation of cars sharing origin and destination. Trains are built of one or more blocks coupled together as needed.

(ii) A *train* is a locomotive carrying or not carrying blocks.

(iii) A *block swap* occurs when a block is moved from one train to another.

(iv) A *block-path* is a sequence of railroad segments through which a block can be feasibly routed from its origin to its destination.

(v) A *work-event* occurs each time a train stops en route at a station to pick up and/or deliver blocks. The train stop at its destination (or origin) station is not considered as a work-event.

(vi) A *crew segment* is a minimal length route between two stations, called *end points*, on which crews operate trains in either direction.

(vii) The *crew imbalance* on a crew segment, defined by say end points x and y, is the absolute difference between the number of crews going from x to y, and the number of crews going from y to x.

(viii) The *train imbalance* on a station, say x, is the absolute difference between the number of trains originating in x and the number of trains terminating in x.

(ix) A car is *missed* if it is not transported from its origin to its destination.

Constraints

(i) *Blocks per train*: trains are constrained by an upper bound on the number of blocks they carry.

(ii) *Swaps per block*: each block is constrained by an upper bound on the number of times it can be swapped.

(iii) *End points*: crews must start and end traveling at end points.

(iv) *Crew-to-train assignment*: every train has to be assigned to a crew on each crew segment, and it must originate and terminate at the end points of a crew segment, even if the train has to move part of the way along a crew segment not carrying any blocks (the entire train routes should be decomposed in crew segments).

(v) *Upper bounds on segments*: railroad segments are constrained by upper bounds on the number, length, and weight of trains traversing them in either direction.

The objective of the TDOP is to minimize the sum of eight components:

(1) *Locomotive cost*: product of the number of scheduled locomotives and the unit locomotive cost C_1.

(2) *Locomotive travel cost*: product of the total traveled miles by all scheduled locomotives, and the per mile locomotive travel cost C_2.

(3) *Work-event cost*: product of the total number of work-events of all scheduled trains, and the unit work-event cost C_3.

(4) *Car travel cost*: product of the total traveled miles by all cars, and the per mile car travel cost C_4.

(5) *Block swap cost*: let $s(v)$ be the unit block swap cost in station v. For a given block b, denote $\beta(b)$ the set of stations where block b is swapped; then its (total) block swap cost is $\sum_{v\in\beta(b)} s(v)$.

(6) *Crew imbalance cost*: product of the number of all crew imbalances and the unit crew imbalance cost C_5.

(7) *Train imbalance cost*: product of the number of all train imbalances and the unit train imbalance cost C_6.

(8) *Missed car cost*: product of the total number of missed cars and the cost per missed car C_7.

Example 1. Consider a railroad network with five stations A, B, C, D, and E and six railroad segments as schematically depicted in Figure 1, where distances (in miles) are assumed symmetrical and all segments are bidirectional. Seven blocks must be delivered. Table 1 furnishes the relevant data.

The only nonadjacent end points defining a crew segment are B and D. The corresponding crew segment is BCD because its length is minimal among all possible paths connecting B and D. Clearly, the other crew segments are trivially found. Thus, if the route of some train is, say, D → C → B → A, then at D a crew from crew segment BCD could be assigned to this train in the subroute D → C → B, and subsequently at B a crew from crew segment BA could be assigned to the train in the subroute B → A. When a train crosses over from one crew segment to another, the onboard crew gets off the train and a new crew gets onboard. Further, crew segments are bidirectional. Hence, crews in crew segment BA can take a train from A to B or vice versa. It is assumed that crews always travel along the shortest path between any pair of stations.

2.2. A Feasible Solution of Example 1. Consider the solution shown in Table 2. Two trains, t_1 and t_2, are scheduled to transport blocks. Note that train t_1 picks up 63 cars at station D and then goes to C, where it delivers 63 cars and picks up 5 cars. From station C train t_1 travels to B to pick up 42 cars; then it goes to station A to deliver 47 cars and to pick up 13 cars. Finally, it travels to station B to deliver 13 cars.

The distance traversed by both t_1 and t_2 is 1,273 miles. Train t_1 stops to pick up or drop blocks in stations C, B, and A, while train t_2 stops to pick up or drop blocks in stations D, B, and C. Hence, the total number of work-events produced by both trains is six.

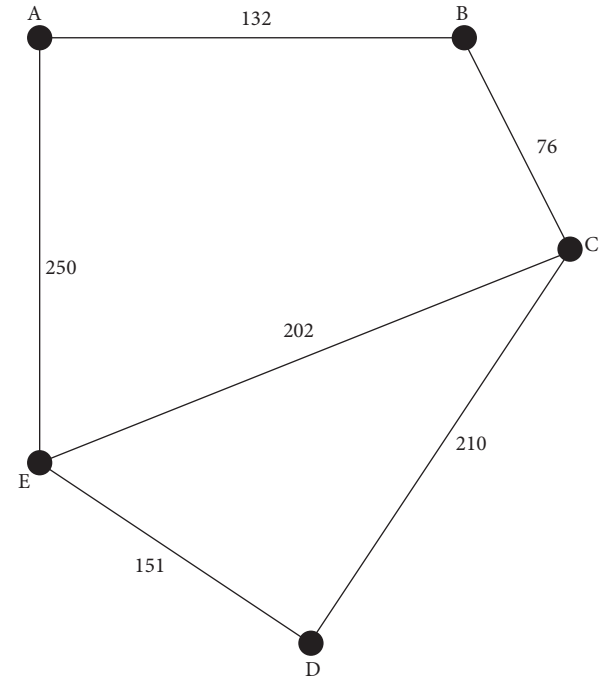


Figure 1: Diagram of railroad networks. Example 1.

Table 3 indicates the train on which each block travels between stations. For instance, block b_3 travels first on train t_1 from A to B, then on train t_2 from B to D, yielding one block swap (with cost 60).

The column labeled "Car miles" is computed as the product of the distance traveled by a block on a segment, and the number of cars in the block. Note that the block-to-train assignment satisfies the constraints on the maximum number of block swaps, and on the maximum number of blocks per train (see Tables 1(a) and 1(b)).

In Table 4 the crew-to-train assignment information is provided. It is shown, for instance, that a crew travels from D to B on train t_1. Then another crew travels from B to A on train t_1. Finally, this crew travels from A to B on the same train.

Note that there is one train imbalance in station B because it is the origin of no train, and one train terminates there. Similarly, there is one train imbalance in station E. Hence, this solution yields two train imbalances.

The objective function value of this solution is as much as 47,603, computed in Table 5 (costs in \$).

3. The Model

We consider the railroad network as a directed graph $G = (V, E)$ whose set of nodes V corresponds to the set of stations, and the set of arcs E is derived from the railroad segments between stations, associating a couple of arcs with opposite directions to each rail segment. The length of arc $e = (u, v) \in E$ is equal to the distance between stations $u, v \in V$. Also, for $e \in E$, let $\overline{L}(e)$, $\overline{W}(e)$, and $\overline{T}(e)$ denote, respectively, the maximum train length, the maximum weight, and the

Table 1: Data of Example 1.

(a)

Station	Block swap cost
A	60
B	60
C	80
D	20
E	20

(b)

Block	Origin	Destination	# of cars	Length (feet)	Weight (tons)
b_1	C	A	5	290	420
b_2	C	D	48	2,976	3,696
b_3	A	D	13	819	962
b_4	D	B	4	228	316
b_5	E	D	12	708	936
b_6	D	C	63	3,969	4,914
b_7	B	A	42	2,730	3,570

(c)

Segment	Length (miles)	Maximum train length (feet)	Maximum train weight (tons)	Max number of trains
BC	76	4,100	5,600	6
AB	132	4,400	6,300	12
DE	151	5,700	5,400	6
CE	202	4,500	7,900	11
CD	210	4,000	10,000	9
AE	250	6,200	6,500	6

(d)

End points defining five crew segments
{B, A}
{B, D}
{A, E}
{C, E}
{D, E}

(e)

Description	Value
Unit locomotive cost (C_1)	\$400
Train travel cost per mile (C_2)	\$10
Cost per work-event (C_3)	\$350
Car travel cost per mile (C_4)	\$0.75
Unit crew imbalance penalty (C_5)	\$600
Unit train imbalance penalty (C_6)	\$1,000
Cost per missed car (C_7)	\$5,000
Maximum number of blocks per train (M_B)	8
Maximum number of swaps per block (M_S)	3
Maximum number of work-events per train (M_W)	4

Table 2: Train routes solution of Example 1.

Train	Sequence	Station	Cumulative miles	Deliver cars	Pick up cars	Cars in the train	Crew change
t_1	1	D	0	0	63	63	NO
	2	C	210	63	5	5	NO
	3	B	286	0	42	47	YES
	4	A	418	47	13	13	NO
	5	B	550	13	0	0	NO
t_2	1	E	0	0	12	12	NO
	2	D	151	12	4	4	YES
	3	C	361	0	0	4	NO
	4	B	437	4	13	13	NO
	5	C	513	0	48	61	NO
	6	D	723	61	0	0	NO
	Total mileage		1,273				

Table 3: Block-to-train assignment of Example 1.

Block	Sequence	Train	Start station	End station	block swap cost	Segment miles	# of cars	Car miles
b_1	1	t_1	C	A	0	208	5	1,040
b_2	1	t_2	C	D	0	210	48	10,080
b_3	1	t_1	A	B	60	132	13	1,716
b_3	2	t_2	B	D	0	286	13	3,718
b_4	1	t_2	D	B	0	286	4	1,144
b_5	1	t_2	E	D	0	151	12	1,812
b_6	1	t_1	D	C	0	210	63	13,230
b_7	1	t_1	B	A	0	132	42	5,544
Totals					60			38,284

Table 4: Crew-to-train assignment of Example 1.

Crew segment	Train	Forward	Backward
BD	t_1	0	1
BA	t_1	1	1
DE	t_2	0	1
BD	t_2	1	1

maximum number of trains allowed to traverse arc (or segment) e in either direction.

Denote C the set of crew segments, where each crew segment $c \in C$ is a path through a set of stations $\chi(c)$—including end points—and let $H \coloneqq \bigcup_{c \in C} \chi(c)$. We assume crews are always traveling on crew segments.

Let B be the set of blocks to be delivered. For block $b \in B$, its origin, destination, number of cars, length, and weight are denoted $o(b)$, $d(b)$, $r(b)$, $\ell(b)$, and $w(b)$, respectively.

Let Γ be the set of all possible trains, and let $P(\cdot)$ be a generic sequence of arcs of E, with $L_{P(\cdot)}$ denoting its length. Thus, the arc sequence followed by train $t \in \Gamma$ is $P(t)$, and that of block b is $P(b)$; namely, $P(b)$ is a block-path of b. In case train $t \in \Gamma$ is used, let $CR(t)$ and $WE(t)$ be, respectively, the set of crews traveling on train t and the number of work-events of train t. Our mathematical formulation below closely follows the one proposed in [20].

Let $ct(t)$ denote the cost of train $t \in \Gamma$, which consists of the unit locomotive cost C_1, plus the train travel cost, and plus the cost of work-events. Hence,

$$ct(t) = C_1 + \left(C_2 \times L_{P(t)}\right) + \left(C_3 \times WE(t)\right). \quad (1)$$

Each train $t \in \Gamma$ can be seen, in fact, as a sequence of crew segments, forming a path $P(t)$. The subset of trains passing through arc $e \in E$ is $T_e \in \Gamma$, and K_t^e is the number of times train t passes through e. From set Γ and graph G, one can obtain a multigraph $MG = (V, A)$, where A is obtained by replacing each arc $e \in E$ with $\sum_{t \in T_e} K_t^e$ copies of it. If a train passes several times through arc e, then each passage induces a different copy.

The binary variable λ_t states whether train $t \in \Gamma$ is used or not in the solution. Let $\Delta_v(t)$ be equal to -1 if train t starts its route in node v, and equal to $+1$ if train t ends its route in v. Thus, $\Delta_v = |\sum_{t \in \Gamma} \Delta_v(t)\lambda_t|$ is the train imbalance in node v.

Let u and v be the end points of a crew segment $c \in C$. Then $\Delta_c(t)$ denotes the difference between the number of times train t goes through c, from u to v, and the number of times it goes through c from v to u. So, $\Delta_c = |\sum_{t \in \Gamma} \Delta_c(t)\lambda_t|$ is the crew imbalance of crew segment c.

Let Ω_b denote the set of all block-paths $P(b)$ associated with block $b \in B$, which includes a dummy block-path of one arc from $o(b)$ to $d(b)$ with cost $C_7 \times r(b)$. Further, let $\Omega_b^t \subseteq \Omega_b$ be the subset of paths in Ω_b that use train t, and let $\Omega = \bigcup_{b \in B} \Omega_b$ be the whole set of block-paths.

Table 5: Total costs of Example 1.

Locomotive cost	2 (number of locomotives) × 400 = 800
Train travel cost	1,273 × 10 = 12,730 (see Table 2)
Car travel cost	38,284 × 0.75 = 28,713 (see Table 3)
Work-event cost	6 (number of train work-events) × 350 = 2,100
Blocks swap cost	60 (see Table 3)
Crew imbalance cost	2 × 600 = 1,200 (see Table 4)
Train imbalance cost	2 × 1,000 = 2,000
Missed cars cost	0

For $b \in B$, let $cb(b)$ denote the block-path cost for block b, namely, in case of a dummy block-path; then $cb(b) = C_7 \times r(b)$; otherwise

$$cb(b) = \left(C_4 \times r(b) \times L_{P(b)}\right) + \sum_{v \in \beta(b)} s(v), \tag{2}$$

whose two terms correspond to block travel cost and block swap cost, respectively. Recall that $r(b)$ is the number of cars of block b, each block swap at node $v \in V$ costs $s(v)$, and $\beta(b)$ is the set of stations where b is block swapped.

Let Ω_e be the set of block-paths that use arc $e \in A$ of multigraph MG. Note that there is one train corresponding to Ω_e. Binary variable $x_{P(b)}$ states whether block-path $P(b) \in \Omega$ is used or not in the solution.

Therefore, our mathematical model of the TDOP is

$$\begin{aligned} \min \quad & z \\ & = \sum_{t \in \Gamma} ct(t)\,\lambda_t + \sum_{P(b) \in \Omega} cb(b)\,x_{P(b)} \\ & \quad + C_5 \sum_{c \in C} \Delta_c + C_6 \sum_{v \in V} \Delta_v \end{aligned} \tag{3}$$

$$\text{subject to} \quad \sum_{P(b) \in \Omega_b} x_{P(b)} \geq 1, \quad \forall b \in B \tag{4}$$

$$\sum_{P(b) \in \Omega_b^t} x_{P(b)} \leq \lambda_t, \quad \forall t \in \Gamma \tag{5}$$

$$\sum_{b \in B} \sum_{P(b) \in \Omega_b^t} x_{P(b)} \leq M_B, \quad \forall t \in \Gamma \tag{6}$$

$$\sum_{P(b) \in \Omega_e} \ell(b)\,x_{P(b)} \leq \overline{L}(e), \quad \forall e \in E \tag{7}$$

$$\sum_{P(b) \in \Omega_e} w(b)\,x_{P(b)} \leq \overline{W}(e), \quad \forall e \in E \tag{8}$$

$$\sum_{t \in T_e} K_t^e \lambda_t \leq \overline{T}(e), \quad \forall e \in E \tag{9}$$

$$WE(t) \leq M_W, \quad \forall t \in \Gamma \tag{10}$$

$$x_{P(b)}, \lambda_t \in \{0, 1\}, \quad \forall P(b) \in \Omega,\ t \in \Gamma. \tag{11}$$

The terms in the objective function (3) correspond to the costs of locomotives, blocks, crew imbalances, and train imbalances, in that order. Constraint (4) imposes the existence of a block-path—real or dummy—for each block $b \in B$. Constraint (5) forces selecting train t if any of the selected block-paths uses it. Constraints (6)–(10) stand for upper bounds on number of blocks that a train can transport, length and weight of trains passing through arcs, number of trains passing through arcs, and number of work-events per train, respectively. We emphasize that every sequence of train $t \in \Gamma$ must be decomposed in a sequence of crew segments.

4. Algorithm

This section presents the algorithm developed by us to tackle the TDOP. It consists of two main steps.

Step 1 (initial feasible solution). Composed by substeps (1.1)–(1.3), this step is aimed to produce from scratch a feasible solution of the TDOP, namely, where constraints (6)–(11) are satisfied.

(1.1) Crew Selection. For every block $b \in B$ determine a minimum length path ξ_b from $o(b)$ to $d(b)$, where each arc can be operated by at least one crew. The set of crews $\pi(b)$ to operate along path ξ_b is chosen following substeps (1.1.1)–(1.1.5) below. For every $\pi(b)$ a minimum set of trains is constructed, so as only one train corresponds to every pair of crews such that one ends where the other starts. Note that with this rule the corresponding trains to carry b can be easily deduced. Let $\mathcal{Q}(b) := \{Q \mid Q \text{ be a set of crew segments covering the arcs of } \xi_b\}$.

In substep (1.1.4) every possible effort is made to avoid the overlapping of crews in $\mathcal{Q}(b)$; therefore, even though $\mathcal{Q}(b)$ contained overlapping crews, this is not an issue since the probability of this happening is truly negligible. In the sequel $\mathcal{Q}(b)$ will be simply written $\mathcal{Q}$. Although $\mathcal{Q}$ is theoretically of exponential size, in all our experiments we were able to enumerate it completely.

(1.1.1) For $Q \in \mathcal{Q}$, let $\gamma(Q)$ be the number of swaps of block b when transported through crew segments Q. Determine $\gamma^* := \min\{\gamma(Q) : Q \in \mathcal{Q}\}$ and reduce $\mathcal{Q}$ by eliminating from it every Q such that $\gamma(Q) > \gamma^*$. If $\mathcal{Q}$ has a unique element, say, $\widehat{Q}$, make $\pi(b) := \widehat{Q}$; otherwise go to (1.1.2).

(1.1.2) If in $\mathcal{Q}$ there is a crew set starting in $o(b)$, then reduce $\mathcal{Q}$ by eliminating from it every Q not starting in $o(b)$. If $\mathcal{Q}$ has a unique element, say, $\widetilde{Q}$, make $\pi(b) := \widetilde{Q}$; otherwise go to (1.1.3).

(1.1.3) If in $\mathcal{Q}$ there is a crew set ending in $d(b)$, then reduce $\mathcal{Q}$ by eliminating from it every Q not ending in $d(b)$. If $\mathcal{Q}$ has a unique element, say, $\ddot{Q}$, make $\pi(b) \coloneqq \ddot{Q}$; otherwise go to (1.1.4).

(1.1.4) Let $\rho(Q)$ be the total distance traveled by the trains corresponding to the crew segments in $Q \in \mathcal{Q}$, and let $\rho^* \coloneqq \min\{\rho(Q) : Q \in \mathcal{Q}\}$. Reduce $\mathcal{Q}$ by eliminating from it every Q such that $\rho(Q) > \rho^*$. If $\mathcal{Q}$ has a unique element, say, $\overline{Q}$, then make $\pi(b) \coloneqq \overline{Q}$; otherwise go to (1.1.5). For example, assume we have a block with origin A and destination D, together with crews A-B-C, B-C-D, and A-B. One solution is to use crews A-B-C and B-C-D, while another uses crews B-C-D and A-B; with this criterion the second solution is chosen.

(1.1.5) Let $q^* = \min\{|Q| : Q \in \mathcal{Q}\}$. Reduce $\mathcal{Q}$ by eliminating from it every Q such that $|Q| > q^*$. If $\mathcal{Q}$ has a unique element, say, $\check{Q}$, make $\pi(b) \coloneqq \check{Q}$; otherwise make $\pi(b)$ equal to an arbitrarily selected $Q \in \mathcal{Q}$.

(1.2) Block Ordering. An order $\overline{B}$ on the set of blocks B is constructed as follows. Let $L(\xi_b)$ be the length of path ξ_b. If $L(\xi_{b'}) > L(\xi_{b''})$ then block b' precedes block b'' in $\overline{B}$. However, if $L(\xi_{b'}) = L(\xi_{b''})$, then b' precedes b'' in $\overline{B}$ whenever $\nu(b') > \nu(b'')$, where $\nu(\cdot)$ denotes the number of blocks with origin $o(\cdot)$ and destination $d(\cdot)$. In case $L(\xi_{b'}) = L(\xi_{b''})$ and $\nu(b') = \nu(b'')$ hold together, block b' precedes b'' whenever $r(b') > r(b'')$. Finally, if $L(\xi_{b'}) = L(\xi_{b''})$, $\nu(b') = \nu(b'')$, and $r(b') = r(b'')$, this tie between b' and b'' is arbitrarily broken to produce $\overline{B}$.

(1.3) Train Assignment. Blocks are considered one after the other, according to order $\overline{B} = (b_1, \ldots, b_m)$. First, assign block b_1 to the crews selected in Step (1.1), together with the required train(s). For $j = 2, \ldots, m$, if path ξ_{b_j} is contained in path ξ_{b_i} for some $i \in [1, j-1]$, then assign block b_j to path ξ_{b_i} whenever upper bounds on feet, tons, and max blocks per train allow. Otherwise, assign block b_j to the crews selected in Step (1.1), together with the required train(s). This step produces a feasible solution S.

Step 2 (simulated annealing). Proposed more than three decades ago [24, 25] Simulated Annealing (SA, for short) is one of the most successful metaheuristics to find good solutions of many combinatorial optimization problems (see [26] and the references therein), including the railroad freight transportation design problem [27]. In its theoretical formulation, SA converges with probability 1 to a global minimum under certain assumptions on the control parameters. In practice, these assumptions are impossible to be implemented, but adequate cooling schemes increase the likelihood of obtaining a near optimal solution [28].

Central to SA is the neighborhood concept. For our proposal we hand tailored it as follows. Let $\Phi(\mathcal{J})$ denote the set of feasible solutions of an instance $\mathcal{J}$ of the TDOP. The *neighborhood* of a solution $S \in \Phi(\mathcal{J})$ is

$$N(S) = N_1(S) \cup N_2(S) \cup N_3(S), \tag{12}$$

where (i) $N_1(S) \subseteq \Phi(\mathcal{J})$ contains every solution obtainable from S by adding one or more trains to deliver one block alone, say b, through ξ_b. (ii) $N_2(S) \subseteq \Phi(\mathcal{J})$ contains the solutions obtainable from S by removing one block, say b, from the set Ψ of trains carrying it and delivering it through a minimal length route from $o(b)$ to $d(b)$ without using any train of Ψ, using another trains in S. (iii) $N_3(S) \subseteq \Phi(\mathcal{J})$ contains every solution obtainable from S by the removal of one block, say b, from the set Ψ of trains carrying it, and delivering it through a minimal length route from $o(b)$ to $d(b)$ without using any train of Ψ, with at least one added train, and using at least one existing train.

```
τ ← T_o; S* ← S;
While τ ≥ T_f do
    k ← 0;
    While k ≤ θ do
        S' ←Neighbor(S);
        δ ← z(S') − z(S);
        If Rand(0, 1) < e^{−δ/τ} or (δ < 0) then
            S ← S';
        EndIf
        k ← k + 1;
        If z(S) < z(S*) then
            S* ← S; k ← 0;
        EndIf
    EndWhile
    τ ← ατ;
EndWhile
call Fusion.
```

Algorithm 1: Algorithm simulated annealing.

We implemented a simulated annealing procedure with geometrical cooling scheme as shown below, where the variable τ stands for the system temperature, and the parameters T_o, T_f, α, θ, denote, respectively, the initial and the final temperature, the cooling factor, and the internal cycle length.

Procedure Neighbor(S) randomly produces (with uniform distribution) a feasible solution in $N(S)$, and Rand delivers a uniformly distributed random number in the interval $(0, 1)$. SA starts with the solution S delivered by Step 1 above (see Algorithm 1).

Throughout the process, the variable S^* holds the best solution so far, and the counter k helps to control the number of iterations of the internal cycle. Thus, the system temperature τ drops if and only if θ iterations occur without any improvement on S^*.

Added at the end of the external cycle, procedure Fusion attempts to improve solution S^* by successively applying the following four operations in the order shown. Before passing to the next operation, for each operation all possibilities are considered, repeating it exhaustively as long as the cost is lowered and feasibility is preserved.

(1) When two trains have identical routes, one locomotive is eliminated once its blocks are passed to the other.

Table 6: Average results—objective function value rounded to nearest integer—of ten runs for instance *Data Set 2*. Time shown in seconds and italics.

$\theta \setminus \alpha$	0.70	0.75	0.80	0.85	0.90	0.95	0.99
500	3,209,687	3,210,058	3,212,176	3,209,065	3,205,954	3,195,864	3,185,775
	2,790	*2,226*	*2,446*	*2,666*	*2,783*	*3,934*	*10,877*
1,000	3,205,775	3,204,303	3,203,858	3,200,948	3,201,772	3,191,151	3,184,551
	2,459	*2,593*	*2,771*	*3,351*	*4,259*	*5,181*	*19,899*
1,500	3,200,260	3,201,520	3,192,792	3,190,837	3,187,281	3,190,916	3,173,312
	2,605	*2,634*	*2,901*	*3,450*	*4,355*	*7,264*	*29,082*
2,000	3,201,157	3,196,194	3,192,166	3,195,499	3,188,878	3,188,311	3,175,760
	3,070	*2,890*	*2,540*	*4,332*	*4,997*	*9,066*	*36,945*
4,000	3,195,857	3,187,514	3,187,182	3,185,158	3,184,931	3,176,745	3,173,650
	3,808	*4,771*	*5,339*	*6,844*	*8,904*	*16,043*	*76,952*
6,000	3,192,045	3,189,039	3,185,947	3,180,655	3,181,356	3,175,733	3,169,176
	5,857	*5,976*	*7,398*	*8,711*	*16,026*	*25,530*	*116,304*
8,000	3,188,106	3,185,158	3,184,032	3,181,325	3,179,988	3,172,099	3,167,935
	6,741	*6,844*	*9,216*	*12,206*	*18,319*	*37,695*	*153,931*
10,000	3,186,663	3,184,824	3,182,085	3,178,864	3,176,411	3,168,984	3,164,844
	7,466	*8,622*	*10,846*	*14,772*	*21,081*	*42,862*	*218,506*

(2) If there are stations u, v, such that $\sum_{t\in\Gamma} \Delta_u(t)\lambda_t < 0$ and $\sum_{t\in\Gamma} \Delta_v(t)\lambda_t > 0$, then a train with no blocks is added starting in v and ending in u.

(3) If a train starts its route in a station where another train terminates its travel, one locomotive is eliminated once its blocks are passed to the other.

(4) If there are trains t', t'', such that the t'' route is contained in the t' route, then the t'' locomotive is eliminated once its blocks are passed to t'.

Example 2. For clarity sake, consider a railroad network as schematically depicted in Figure 2. Assume a block b with $o(b)$ = A and $d(b)$ = D, and let the crew segments be AF, AD, AI, AP, FD, IL, LD, QR, and DP. Also, let S be a feasible solution with six trains in correspondence to routes: A → E → F, F → G → D, A → H → I, I → J → F → K → L, A → N → O → P, and L → M → D, where block b is delivered by the trains assigned to routes A → E → F and F → G → D; hence 63 is their total traveled distance. Thus, solution $S' \in N_1(S)$ is identical to S with the exception that block b is delivered by a new train through route A → B → C → D, with length 42.

Solution $S'' \in N_2(S)$ is identical to S with the exception that block b is delivered by the trains with routes A → H → I, I → J → F → K → L, and L → M → D, with total length 91, and without adding a new train.

Solution $S''' \in N_3(S)$ is identical to S with the exception that block b is delivered through route A → N → O → P, adding a new train for segment PD; the total length of this route is 47.

In regard to simulated annealing parameters, we tested diverse settings, modifying one parameter at a time. Among the possible combinations of the cooling factor α = 0.70, 0.75, 0.80, 0.85, 090, 0.95, and 0.99, and the internal cycle lengths θ = 500, 1,000, 1,500, 2,000, 4,000, 6,000, 8,000, and 10,000, for every pair $\{\alpha, \theta\}$ we run ten times the algorithm on a PC with Windows 8.

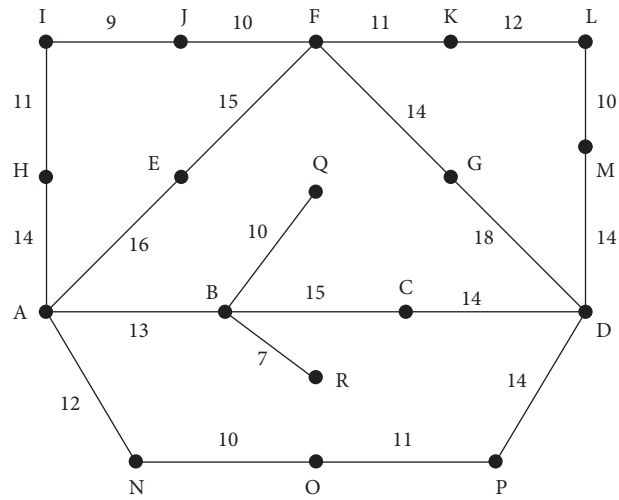


Figure 2: Diagram of railroad networks. Example 2.

Table 6 displays the average results obtained for instance *Data Set 2*, where computer times are shown in italics. Thus, we chose α = 0.90 and θ = 1,000 since together they yielded a good trade-off between computer time and solution quality. As initial temperature we chose T_o = 30,000, because on average it yielded an acceptance rate of around 95%. For the final temperature T_f = 1 was selected, yielding an acceptance rate of around 1% on average.

5. Numerical Results

Our simulated annealing approach, as described in Section 4 to deal with the TDOP, was implemented on a computer with Xeon E5-2643 v2 3.5 GHz processor (we run in a single

Table 7: Performance comparison of our SA approach and other methods; computing times are given in seconds. The bottom row shows a lower bound on the optimal cost, computed as explained in Section 5.3. (all figures rounded to integers).

Number of	*Example*		*Data Set 1*		*Data Set 2*	
Nodes	4		94		221	
Blocks	5		239		369	
Arcs	5		134		294	
Crew Segments	5		154		154	
	cost	time	cost	time	cost	time
[21]		—	2,043,471	45	3,187,999	225
[20]		—	2,032,516	2,201	3,188,500	2,067
[18]		—	2,018,644	80,000	3,152,019	75,000
[23]	14,916	0	2,069,752	342	3,240,177	2,700
SA	14,916	1	2,005,514	181	3,170,815	1,991
Lower bound σ	10,956		1,660,361		2,751,213	

processor), 64 GB RAM, and g++ compiler. Two experiments were conducted to investigate its efficiency.

In the first experiment—see Section 5.1—we tested SA on three instances available to us: *Example* (a toy, synthetic instance), *Data Set 1* (real), and *Data Set 2* (real), from [13].

The second experiment was designed to evaluate the performance of SA from the point of view of the quality of the results and the required computer time. We did extensive testing on randomly generated instances of various sizes; they are dealt with in Section 5.2.

5.1. Testing SA on Specific Instances. Employing parameters $\theta = 1000$ and $\alpha = 0.9$, we compared SA results on instances *Example*, *Data Set 1*, and *Data Set 2*, as can be seen in Table 7, where computation times are also shown to give an idea of the performance of the implementation, although the platforms used were not the same. So far [18] had provided—using a mixed integer programming model—the best results for the two real instances but at the expense of very high computing time. On the other hand, the network-oriented formulation in [21] yielded the fastest algorithm to date.

Other approaches include column-generation [23], with better results than those found in [22], and column-generation combined with tabu search [20] improving on [19]. In the case of instance *Example* the only published results come from [23]. The best results found with SA were 1,999,315 with $\theta = 8000$ and $\alpha = 0.99$, in 19,819 seconds for instance *Data Set 1*, and 3,155,267 with $\theta = 10{,}000$ and $\alpha = 0.95$ in 41,233 seconds for instance *Data Set 2*, with the more relevant results found in the literature; our results yield lower cost for *Data Set 1* (−0.95%), same cost for *Example*, and higher cost for *Data Set 2* (+0.10%).

5.2. Testing SA on Random Instances. A set of 20 random instances was generated as follows with the number n of stations and number m of blocks shown in column II of Table 8. All random choices were made with uniform distribution.

For each pair $\{n, m\}$ we first construct a complete graph $K_n = (V, E)$, whose vertex set V corresponds to a set of n integer points randomly generated in a square of side 2^{15}; the length of each edge $(u, v) \in E$ is equal to the Euclidean distance between vertices u and v. Also, we determine the set $E_t \subset E$ of edges in a minimum length spanning tree on K_n, as well as the set $E_h \subset E$ of edges in the convex hull of V. Finally, we make the railroad network of the instance correspond to graph $G = (V, E_h \cup E_t)$.

Then, crew segments are randomly created so that every edge of the network belongs to one crew segment. The swap cost $s(v)$ in every station $v \in V$ is also randomly assigned in the integer range[10, 100].

We create next a set of m blocks, one after the other, verifying that each does not exceed the maximum number of swaps allowed to arrive to destination when traveling along a minimal length path. The number of cars $r(b)$, length $\ell(b)$, and weight $w(b)$ of every block b are randomly chosen in the integer ranges [1, 100], [56, 65], and [74, 86], respectively, as these ranges are similar to those found in real instances. Each time we tentatively form a block, Step 1 of Section 4 is performed for feasibility verification.

Also, for every edge $e \in E$, the values of $\overline{L}(e)$, $\overline{W}(e)$, and $\overline{T}(e)$ are randomly chosen in the ranges [7,000, 14,000] and [9,000, 18,000], [6, 12], respectively.

Once random instances were generated, SA run 50 times for each. Results are displayed in columns III–V of Table 8 (indeed, instance p5_7D was taken as Example 1 in Section 2).

5.3. Lower Bounds. Establishing good lower bounds for the optimal solution of the TDOP seems a very difficult task. However, to assess the SA performance, we propose here a lower bound σ for each solved instance; see Tables 7 and 8. To this aim we computed as explained below lower bounds for total travel cost (σ_1), train start cost (σ_2), train travel cost (σ_3), work-event cost (σ_4), and missed car cost (σ_5). Thus, $\sigma = \sum_{j=1}^{5} \sigma_j$.

In regard to travel cost let $\sigma_1 = C_4 \sum_{b \in B} (\psi(b) \times r(b))$, where $\psi(b)$ is the length of the shortest route from $o(b)$ to $d(b)$. In the case of the two real instances σ_1 is identical to the bound computed in [23].

Table 8: Relevant data (columns I, II) and results (columns III, IV, and V) of 50 runs of SA on 20 random instances. A lower bound—explained in Section 5.3—on the optimal solution cost is shown in column VI. The last column is an indicator of the solution quality.

I	II	III	IV	V	VI	VII
Instance	Number of stations, blocks, arcs, crew segments	Minimum cost (μ)	Average cost	Avg time (sec)	Lower bound (σ)	Difference in % $100 \times \frac{\mu - \sigma}{\sigma}$
p5_7D	5 7 5 6	47,193	47,193	1	32,957	43.18
p5_10D	5 10 4 6	77,691	77,726	1	60,336	28.76
p10_15D	10 15 11 14	224,186	226,845	4	187,862	19.34
p10_20D	10 20 8 13	294,595	299,021	4	248,649	18.48
p20_30D	20 30 13 25	297,952	301,013	13	243,932	22.15
p20_40D	20 40 15 25	336,651	339,876	13	267,840	25.69
p40_60D	40 60 22 45	565,954	576,906	49	446,538	26.74
p40_80D	40 80 21 45	568,089	573,991	46	446,411	27.26
p80_120D	80 120 38 86	791,072	800,660	124	614,225	28.79
p80_160D	80 160 38 89	987,646	997,005	168	791,041	24.85
p160_240D	160 240 62 169	2,214,593	2,240,989	376	1,791,395	23.62
p160_320D	160 320 72 169	2,201,347	2,221,606	320	1,765,637	24.68
p200_300D	200 300 82 216	4,043,845	4,071,617	409	3,306,843	22.29
p200_400D	200 400 81 210	1,591,058	1,607,980	463	1,203,586	32.19
p250_375D	250 375 96 261	2,518,148	2,548,918	856	1,958,093	28.60
p250_500D	250 500 104 260	1,881,881	1,905,522	723	1,445,175	30.22
p300_450D	300 450 117 311	8,919,042	8,969,957	606	7,480,671	19.23
p300_600D	300 600 115 313	10,419,523	10,453,791	744	8,782,219	18.64
p320_480D	320 480 138 333	8,172,082	8,218,034	662	6,676,505	22.40
p320_640D	320 640 138 332	2,053,295	2,077,627	900	1,516,815	35.37

For the train start cost a trivial lower bound is $\sigma_2 = C_1 \times \lceil m/M_B \rceil$, where M_B stands for the maximum number of blocks per train.

Let $(\psi_1, \ldots, \psi_m)$ be an order on $\Psi = \{\psi(b) : b \in B\}$, with $\psi_j \geq \psi_{j+1}$, for $j = 1, \ldots, m-1$. Thus, for the train travel cost our proposed lower bound is $\sigma_3 = C_2 \sum_{j \in J} \psi_j$, where $J = \{j : (1 = j \bmod M_B)$ and $1 \leq j \leq m\}$. Bound σ_3 slightly improves on that from [20].

For every station $v \in V$, let $\rho(v)$ be the number of blocks b such that $o(b) = v$ or $d(b) = v$. Then $\sigma_4 = C_3 \sum_{v \in X} \lceil \rho(v)/M_B \rceil$, where X is the set of stations belonging to a crew segment with the exclusion of its end points. As far as we know this lower bound for the work-event cost is proposed here for the first time.

Let Y denote the set of blocks b such that $o(b)$ or $d(b)$ does not belong to H (refer to Section 3 for a definition of vertex set H). Thus, it is impossible to deliver any block in Y. Our lower bound for the missed car cost is then $\sigma_5 = C_7 \sum_{b \in Y} r(b)$. Although in practice it is not likely to get $\sigma_5 > 0$, instance *Data Set 02* contains some blocks belonging to Y.

Thus, in regard to the best solutions found for instances *Example*, *Data Set 1*, and *Data Set 2* (see Table 7) our lower bound σ yields differences of 36.14%, 20.79%, and 14.57%, respectively, namely, 23.83% on average. Column VII of Table 8 shows a similar behavior of σ for the synthetic instances. These facts lead us to believe that σ, although the highest lower bound known to us is far from the true optimum.

6. Conclusion

In this paper we proposed a simulated annealing approach for the Train Design Optimization Problem. This approach was computationally tested with three instances well-known in the specialized literature, and with 20 instances randomly generated by us, of sizes up to 320 stations and 640 car blocks. Its results show superiority—in regard to the objective function value—over those obtained elsewhere for instance *Data Set 01*, and competitivity (in fact the runner-up) for instance *Data Set 02*, both results using reasonable computing resources.

We think however that much research, be it theoretical or empirical, must still be carried out to successfully deal with this most challenging, combinatorial optimization problem. The TDOP being so fundamental in the railroad industry, there is a need to develop new heuristics or metaheuristics, which, working alone or hybridized, improve the best results obtained so far. Also, a pending and difficult task is to produce efficient procedures that yield good lower bounds for the TDOP objective function, so as to help assess the performance of proposed approaches.

Disclosure

David Romero is on sabbatical leave at Laboratorio Nacional de Informática Avanzada, Xalapa, Veracruz, Mexico.

Conflicts of Interest

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

References

[1] A. A. Khaled, M. Jin, D. B. Clarke, and M. A. Hoque, "Train design and routing optimization for evaluating criticality of freight railroad infrastructures," *Transportation Research Part B: Methodological*, vol. 71, pp. 71–84, 2015.

[2] V. Cacchiani and P. Toth, "Nominal and robust train timetabling problems," *European Journal of Operational Research*, vol. 219, no. 3, pp. 727–737, 2012.

[3] M. F. Gorman, "An application of genetic and tabu searches to the freight railroad operating plan problem," *Annals of Operations Research*, vol. 78, pp. 51–69, 1998.

[4] X. Zhou and M. Zhong, "Single-track train timetabling with guaranteed optimality: branch-and-bound algorithms with enhanced lower bounds," *Transportation Research Part B: Methodological*, vol. 41, no. 3, pp. 320–341, 2007.

[5] A. Balakrishnan, A. Kuo, and X. Si, "Real-Time decision support for crew assignment in double-ended districts for U.S. freight railways," *Transportation Science*, vol. 50, no. 4, pp. 1337–1359, 2016.

[6] B. Vaidyanathan and R. K. Ahuja, "Crew Scheduling Problem," in *Handbook of Operations Research Applications at Railroads*, pp. 163–175, Springer Verlag, 2015.

[7] R. K. Ahuja, K. C. Jha, and J. Liu, "Solving real-life railroad blocking problems," *Interfaces*, vol. 37, no. 5, pp. 404–419, 2007.

[8] C. Van Dyke and M. Meketon, "Railway blocking process," in *Handbook of Operations Research Applications at Railroads*, B. W. Patty, Ed., pp. 119–162, Springer Verlag, 2015.

[9] A. A. Assad, "Models for rail transportation," *Transportation Research A: General*, vol. 14, no. 3, pp. 205–220, 1980.

[10] J. Cordeau, P. Toth, and D. Vigo, "A survey of optimization models for train routing and scheduling," *Transportation Science*, vol. 32, no. 4, pp. 380–404, 1998.

[11] S. Harrod and M. F. Gorman, "Operations research for freight train routing and scheduling," in *Wiley Encyclopedia of Operations Research and Management Science*, John Wiley and Sons, 2010.

[12] A. K. Nemani and R. K. Ahuja, "OR models in freight railroad industry," in *Wiley Encyclopedia of Operations Research and Management Science*, John Wiley and Sons, Wiley Encyclopedia of Operations Research and Management Science, 2010.

[13] INFORMS, "RAS Problem Solving Competition," 2011, https://www.informs.org/Community/RAS/Problem-Solving-Competition/2011-RAS-Problem-Solving-Competition.

[14] M. F. Gorman, "Operations research in rail pricing and revenue management," *International Series in Operations Research and Management Science*, vol. 222, pp. 243–254, 2015.

[15] C. Van Dyke and M. Meketon, "Train Scheduling," in *Handbook of Operations Research Applications at Raildroads*, B. W. Patty, Ed., pp. 1–42, Springer Verlag, 2015.

[16] C. Van Dyke and M. Meketon, "Car scheduling/trip planning," in *Handbook of Operations Research Applications at Railroads*, B. W. Patty, Ed., pp. 79–118, Springer Verlag, 2015.

[17] N. Azad, E. Hassini, and M. Verma, "Disruption risk management in railroad networks: an optimization-based methodology and a case study," *Transportation Research Part B: Methodological*, vol. 85, pp. 70–88, 2016.

[18] I.-L. Wang, H.-Y. Lee, and Y.-T. Liang, "Train design optimization," in *INFORMS 2011 Annual Meeting - RAS 2011*, Problem Solving Competition, Charlotte, NC, USA, 2011.

[19] F. Colombo, R. Cordone, and M. Trubian, "A column-row generation heuristic for the train design optimization problem," in *Proceedings of the INFORMS 2011 Annual Meeting–RAS 2011 Problem Solving Competition*, vol. 10, Charlotte, NC, USA, 2011, https://www.informs.org/content/download/254681/2405878/file/AllReportsFrom-RAS-2011_Problem_Competition.pdf.

[20] F. Colombo, *Mathematical Programming Algorithms for Network Optimization Problems. Dissertation, [Dissertation, thesis]*, Università degli Studi di, Milano, Italy, 2014.

[21] L. Lozano, J. E. González, and A. L. Medaglia, "A network-oriented formulation and decomposition approach to solve the 2011 RAS problem," in *Proceedings of the INFORMS 2011 Annual Meeting - RAS 2011 Problem Solving Competition*, vol. 10, Charlotte, NC, USA, https://www.informs.org/content/download/254681/2405878/file/AllReportsFrom-RAS-2011_Problem_Competition.

[22] J. G. Jin, J. Zhao, and D. H. Lee, "INFORMS RAS Problem Solving Competition 2011," in *Proceedings of the INFORMS 2011 Annual Meeting*, vol. 10, Charlotte, NC, USA, https://www.informs.org/content/download/254681/2405878/file/AllReports-From-RAS-2011_Problem_Competition.pdf.

[23] J. G. Jin, J. Zhao, and D. H. Lee, "A column generation based approach for the train network design optimization problem," *Transportation Research Part E*, vol. 50, p. 17, 2013.

[24] V. Cerný, "Minimization of Continuous Functions by Simulated Annealing," Research Institute for Theoretical Physics HU-TFT-84–51, University of Helsinki, 1984.

[25] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[26] E. Aarts, J. Korst, and W. Michelis, "Simulated annealing," in *Search Methodologies*, E.K. Burke and G. Kendall, Eds., 285, p. 265, Springer Science and Business Media, New York, NY, USA, 2014.

[27] Á. Marín and J. Salmerón, "A simulated annealing approach to the railroad freight transportation design problem," *International Transactions in Operational Research*, vol. 3, no. 2, pp. 139–149, 1996.

[28] P. J. Van Laarhoven and E. H. Aarts, "Simulated annealing," in *Simulated Annealing: Theory and Applications*, pp. 7–15, Springer, Netherlands, 1987.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 1517435, 10 pages
https://doi.org/10.1155/2017/1517435

Research Article

Reconfiguration of Distribution Networks with Distributed Generation Using a Dual Hybrid Particle Swarm Optimization Algorithm

Caoyuan Ma,[1,2] Chunxiao Li,[1,2] Xuezi Zhang,[1,2] Guoxin Li,[1,2] and Yonggang Han[1,2]

[1] *School of Electrical and Power Engineering, China University of Mining and Technology, Xuzhou, Jiangsu 221116, China*
[2] *Jiangsu Province Laboratory of Electrical and Automation Engineering for Coal Mining, China University of Mining and Technology, Xuzhou, Jiangsu 221116, China*

Correspondence should be addressed to Guoxin Li; li_guo_xin1@163.com

Received 23 December 2016; Accepted 5 July 2017; Published 6 August 2017

Academic Editor: Maria Patrizia Pera

This paper proposes a reconfiguration strategy of distribution network with distribution generation (DG) based on dual hybrid particle swarm optimization algorithm. By the network structure simplification and branches grouping, network loss was selected as objective function, an improved binary particle swarm optimization algorithm (IBPSO) was used in branch group search, and the proposed group binary particle swarm optimization search algorithm was used in searching within the group to improve search efficiency and avoid early maturing. The proposed algorithm was tested on the IEEE 33-bus distribution power system and compared with other existing literature methods. The influence on the power flow of distribution network by DG position and capacity was studied. Simulation results illustrate that the proposed algorithm can get the optimal configuration results and significantly reduce system energy losses with fast convergence rate. In order to control the smart grid, using a dual hybrid particle swarm optimization algorithm to reconstruct a model, the result of simulation verifies the validity of the model. At the same time, the distributed power grid after reconstruction after optimization can effectively reduce the network loss and improve power supply quality.

1. Introduction

As clean and renewable energy sources, DG (distributed generation), such as photo voltaic and wind power, serves as mostly distributed and independent small power supply installed near the load or user [1, 2]. With the powerful support by governments and the development of smart grid construction, the application of distributed power technology in power grids has been vigorously promoted.

Distributed power supply is a small generating unit directly arranged in distribution networks or users nearby to meet the specific requirements and support economic operation of distribution networks with environmental compatibility. And its power ranges from a few kilowatts to fifty megawatts [3–5]. Reconfiguration of distribution network refers to the change of network switch combination and adjustment of the structure of network operation by closing or opening the disconnected sectional switches and tie switches with constraints satisfied [4–6]. Its purpose is to prevent overload of transmission lines, balance the equilibrium electricity of users, reduce the power loss of the system, and maintain a good electrical quality. The optimization of the network with DG is intrinsically a multiple objective, noncontinuous, and multiple stage hybrid searching process where subgoals and constraints are mutually restricted, resulting in a kind of Nondeterministic Polynomial problem in mathematics [7]. The grid-connected of DG has a great influence on the distribution network, including the voltage distribution, power flow distribution, network line loss, power quality, and short circuit current, which cannot be ignored [8, 9].

The performance and efficiency of distribution network reconfiguration are largely dependent on an efficient search algorithm. Particle swarm optimization algorithm (PSO) is a swarm intelligence optimization algorithm. The principle of PSO is simple, robust, easy to achieve, and so on [10].

This paper proposes a reconfiguration strategy of distribution network with DG. The influences on power flow and voltage quality by the location, capacity, quantity, and type of the DG were analyzed. The improvements of particle swarm optimization algorithm and the static reconfiguration method were also analyzed. Distribution network simplification with DG was studied.

2. Problem Representation

Based on certain constraints, distribution network reconfiguration is to ensure that the network topology is radial and determine the distribution network achieving a certain indicator or a number of indicators to achieve the best operation state of the distribution network [11].

Distribution Network Reconfiguration with DG is a nonlinear, multiobject, discrete, multistage planning problem [12, 13]. In this paper the objective function is to minimize active power loss of the whole distribution network.

$$\min f = \sum_{i=1}^{N} K_i R_i \frac{P_i^2 + Q_i^2}{U_i^2}. \tag{1}$$

i is branch number, N is total branch numbers, K_i is the state of switch (0 means open state and 1 means close state), R_i is the resistor of branch i, P_i and Q_i are the active and reactive power of branch i, and U_i is the up-layer node voltage of branch i.

The constraint conditions are as follows.

(1) Power Flow Constraint Conditions

$$\begin{aligned} P_i + P_{\mathrm{DG}i} &= P_{Di} + U_i \sum_{j=1}^{Nb} U_j \left(G_{ij} \cos\delta_{ij} + B_{ij} \sin\delta_{ij}\right) \\ Q_i + Q_{\mathrm{DG}i} &= Q_{Di} + U_i \sum_{j=1}^{Nb} U_j \left(G_{ij} \sin\delta_{ij} - B_{ij} \cos\delta_{ij}\right). \end{aligned} \tag{2}$$

P_i and Q_i are the active and reactive power into node i, P_{Di} and Q_{Di} are active and reactive power of node i load, $P_{\mathrm{DG}i}$ and $Q_{\mathrm{DG}i}$ are DG's active and reactive power into node i, U_i and U_j are voltage of nodes i and j, G_{ij} and B_{ij} are admittance matrixes, and δ_{ij} is the phase difference between nodes i and j.

(2) Voltage and Current Constraint Conditions

$$\begin{aligned} U_{i\min} < U_i < U_{i\max} \\ I_{i\min} < I_i < I_{i\max}. \end{aligned} \tag{3}$$

(3) Branch Active Power Constraint Condition

$$\left|P_{ij}^{\mathrm{line}}\right| < \left|P_{ij,\max}^{\mathrm{line}}\right|. \tag{4}$$

P_{ij}^{line} is actual active power between nodes i and j, and $P_{ij,\max}^{\mathrm{line}}$ is the maximum active power between nodes i and j.

(4) DG Power Constraint Conditions

$$\begin{aligned} P_{\mathrm{DG},\min} < P_{\mathrm{DG},i} < P_{\mathrm{DG},\max} \\ Q_{\mathrm{DG},\min} < Q_{\mathrm{DG},i} < Q_{\mathrm{DG},\max}. \end{aligned} \tag{5}$$

$P_{\mathrm{DG},i}$ and $Q_{\mathrm{DG},i}$ are active and reactive power of DG i.

(5) Network Topological Constraint. The network topology of the distribution network is radial.

(6) Branch Capacity Constraint

$$\left|S_l\right| \le S_{l,\max}. \tag{6}$$

$S_{l,\max}$ is the maximum value for the complex power injection branch l. This is the branch capacity constraint condition.

3. Distribution Network Simplification

3.1. Analysis of the Radial Power Distribution Network. To ensure the high quality electrical energy in faults state, the distribution network was designed as a ring structure, which performs radial topology on grid operation state [14]. Due to the complexity of the distribution network reconfiguration, there are plenty of switch state combinations, leading to "ring" or "island" scenario that will damage the distribution network power supply and lose electrical power charge [15].

The isolated island is a subsystem which is not connected with the root node of the distribution system and is separated from the main power grid. The ring network means that there are a number of paths from the root node, resulting in a loop in the distribution system. The existence of isolated island or ring network shows that the switch combination is a unfeasible solution. Excessive unfeasible solutions can restrain optimization speed and reduce the efficiency of the search process of the reconfiguration algorithm [16, 17].

In this paper, the matrix method is adopted to judge the radial topological condition of distribution network. The main definition and theorem are as follows.

Definition 1. Set n order simple directed graph as $G = \langle V, E\rangle$, $V = \{v_1, v_2, \ldots, v_n\}$. In the directed graph, V is the node set and E is the edge set. Define a matrix $P = [P_{ij}]_{n\times n}$ and assume

$$P_{ij} = \begin{cases} 1 & v_i \text{ can reach } v_j \\ 0 & v_i \text{ cannot reach } v_j. \end{cases} \tag{7}$$

Then we call the matrix P as the reachable matrix of graph G. The reachable matrix shows whether there is a path between any two nodes in the distribution network.

Theorem 2. *If there is a directed graph $G = \langle V, E\rangle$ with adjacent matrix,*

$$A = \begin{vmatrix} a_{11}, a_{12}, \ldots, a_{1n} \\ a_{21}, a_{22}, \ldots, a_{2n} \\ \cdots \cdots \cdots \cdots \\ a_{n1}, a_{n2}, \ldots, a_{nn} \end{vmatrix}. \tag{8}$$

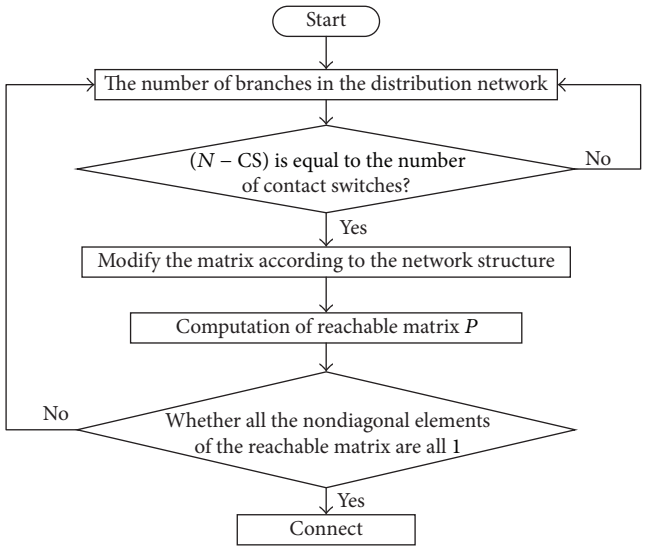


Figure 1: Flow chart of radial distribution network analysis.

Then, it can be concluded that the reachable matrix of the network is $P = A(+)A^2(+)A^3(+)\cdots(+)A^n$, where (+) is the Boolean sum.

Theorem 3. *Suppose $G = \langle V, E\rangle$ is a simple directed graph with no self-loop and P is its reachable matrix. Then the necessary and sufficient condition for the strong connectivity of the graph G is that the elements of P are 1 except diagonal elements.*

The indirect graphs can be seen as a special case of directed graph when every side in indirect graph is equivalent to two sides which have opposite directions. Therefore, the connectivity of indirect graphs can be identified by the strong connection criterion of directed graphs. The distribution system is radial network only when the number of rings is equal to the number of the contact switches. That is, the number of disconnected branches is $N - \mathrm{CS}$ = (total network branch) − (effective node number) + 1 [18].

The flow chart to identify the connectivity of distribution network is shown in Figure 1.

3.2. Distribution Network Simplification. In the reconfiguration of distribution network, the value of the particle position in every dimension is random between 0 and 1, resulting in a large number of unfeasible solutions. Therefore, it is necessary to reduce the particle encoding dimension, simplify the structure of the distribution network, and improve the search efficiency. The IEEE 33 nodes network was shown in Figure 2(a). The network can be simplified according to the rules as follows: the branches with the nodes whose degree (the number of edges adjacent to the node) is 2 is merged into a group and numbered, and the number greater than 2 is retained, as shown in Figure 2(b). Then renumber nodes and branches, as shown in Figure 2(c). Each branch group contains a number of branches; for example, the branch group 6 was composed of the branches 12-13, 13-14, and 14-15. Hence the coding dimension is greatly reduced and the dimension of solution vector is simplified from 37 to 13. Many unfeasible solutions were avoided and efficiency of the algorithm was improved.

In addition, assuming that there are M branches in the network topology, the particle optimization coding dimension of the branch groups is M. Each particle code is a vector composed of 0 and 1, which represents a kind of topology structures of distribution network.

And the particle encoding needs to satisfy

$$N - \mathrm{CS} = NT - NE + 1. \tag{9}$$

NT is total number of branches, and NE is effective node number.

In this way, the number of branch groups disconnected in the reconfiguration of distribution network with DG is the basic mesh number. When we break a branch which belongs to a disconnected branch group, it must satisfy the radial topological condition, avoid the production of the infeasible solutions, and thus improve the search efficiency of the algorithm.

4. Improved Binary Particle Swarm Optimization Algorithm (BPSO)

4.1. Particle Swarm Optimization. Particle Swarm Optimization (PSO) is a kind of global optimization algorithm based on social group behavior, which is derived from the simulation of bird foraging process [19].

Suppose a population consists of n particles, which is generally 20 in the distribution network reconfiguration. Each particle in the population is searching for optimal solutions in a m-dimensional space with a finite speed as the real problem solution is m-dimension [20]. The particle of j in the population is composed of three m-dimensional vectors, that is, respectively, the individual optimal location in history $p_j = (p_{j1}, p_{j2}, \ldots, p_{jm})$, the current position $x_j = (x_{j1}, x_{j2}, \ldots, x_{jm})$, and particle velocity $v_j = (v_{j1}, v_{j2}, \ldots, v_{jm})$ $i = 1, 2, \ldots, n$.

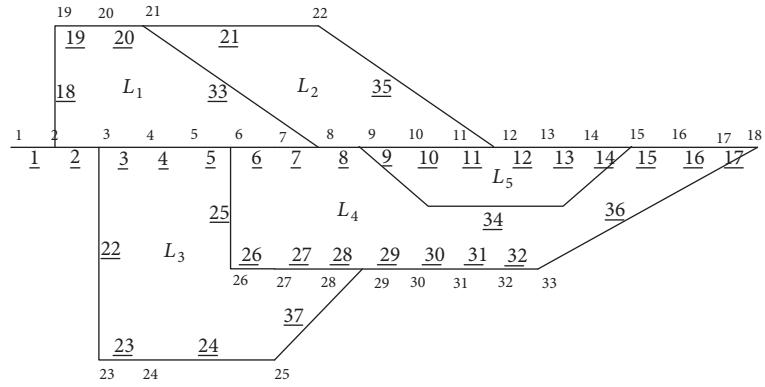
The position of each particle will be calculated and evaluated before next iteration according to the actual problem represented. If the current position is superior to the individual or the best record of population position in history, this position will be deemed as the best position. Otherwise, the location will not be updated.

For each individual in the population, the iteration formula of the velocity and position is

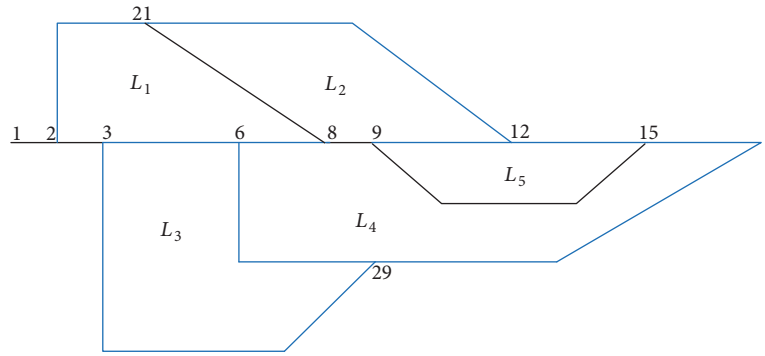
$$\begin{aligned} v_{jm}^{k+1} &= w v_{jm}^{k} + c_1 \mathrm{rand}\left(p_{jm}^{k} - x_{jm}^{k}\right) \\ &\quad + c_2 \mathrm{rand}\left(g_{jm}^{k} + x_{jm}^{k}\right) \end{aligned} \tag{10}$$

$$x_{jm}^{k+1} = x_{jm}^{k} + v_{jm}^{k+1}, \tag{11}$$

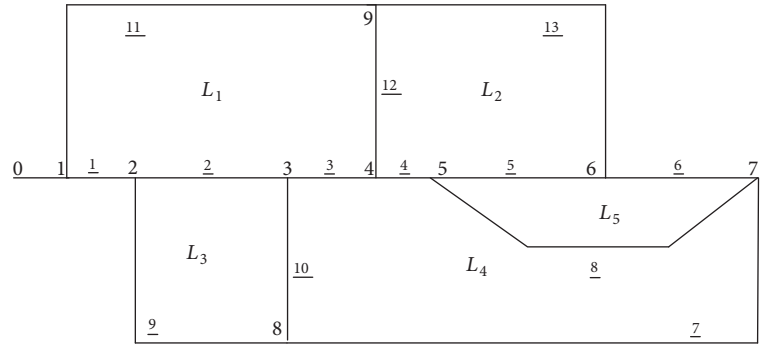
where c_1 and c_2 are the individual learning factor and social learning factor. Their values are nonnegative and range from 0 to 4, which are usually assigned to 2. They can approach to the global optimum by learning by themselves and the experience



(a) IEEE 33 nodes network



(b) Network structure after branches simplification



(c) Network structure after renumbering nodes and branches

Figure 2: Sketch of distribution network simplification.

of the population. rand is a uniform random number between [0, 1]. $v_{\max}$ is the maximum velocity of the particle, and the range of $v_{\max}$ is $[-v_{\max}, v_{\max}]$, which is usually the range of particle value per dimension.

4.2. Improved Binary Particle Swarm Optimization Algorithm (IBPSO). Compared with the continuous PSO, the velocity update formula of binary particle swarm optimization algorithm (BPSO) is the same as that of the former, but the position update formula is different. During the initialization of BPSO particle population, the solution of problem is transformed into binary code [13].

In BPSO, the position change of the particle is the probability that the velocity of each dimension is converted to the bit value 1. Assuming v_{im}^k is the velocity of particle i, the mth is dimension of the kth generation, and the probability of the bit value 1 is $s(v_{im}^k)$, so the probability of the bit value 0 is $1 - s(v_{im}^k)$. Therefore, if the mth dimension and $(k-1)$th generation of the ith particle is 1, then the probability of the occurrence of the mth and kth generation is $1 - s(v_{im}^k)$. Similarly, if the mth dimension and $(k-1)$th generation of the ith particle is 0, then the probability of the occurrence of the mth dimension and kth generation is $s(v_{im}^k)$. If the mth dimension and $(k-1)$th generation of the ith particle

is $s(v_{im}^{k-1})$, then the probability of 0 is $1 - s(v_{im}^{k-1})$. So the probability of occurrence of the mth dimension and kth generation of the ith particle is

$$p_{im}^{k} = s\left(v_{im}^{k-1}\right)\left(1 - s\left(v_{im}^{k}\right)\right) + \left(1 - s\left(v_{im}^{k-1}\right)\right)s\left(v_{im}^{k}\right). \quad (12)$$

According to (12), the probability of particle i change, the mth dimension and kth generation, is related to the velocity of the two generations. When the particles' velocities of both generations are close to 0, the probabilities of bit value change and the probability of random search are probable. The probability of particles convergence to the global optimal position is very small. Expression (10) is composed of three parts. Firstly, wv_{jm}^{k} is the velocity inertia showing the impact of the particle's history velocity on its current velocity. Secondly, $c_1\text{rand}(p_{jm}^{k} - x_{jm}^{k})$ reflects the individual's self-learning ability, a part of "self-cognition." Thirdly, $c_2\text{rand}(g_{jm}^{k} - v_{jm}^{k})$ reflects the individual's ability to learn from the population, a part of "social cognition." Research results show that when $(p_{jm}^{k} - x_{jm}^{k})$ and $(g_{jm}^{k} - x_{jm}^{k})$ are 0, 1, and −1, respectively, 0 means that the values of p_{jm}^{k} g_{jm}^{k} and x_{jm}^{k} are equal; 1 means that p_{jm}^{k} or g_{jm}^{k} is 1, while the value of x_{jm}^{k} is 0; −1 means that p_{jm}^{k} or g_{jm}^{k} is 0, while the value of x_{jm}^{k} is 1. In the latter part of particle swarm search, when v_{jm}^{k} is close to 0, the position of the particle is close to the optimal particle population and the optimal solution, and the particle value should remain intact. When $v_{jm}^{k} < 0$, the value of the historical optimal solution of the optimal particle in population is equal to 0. At this point, the probability of the particle position is 1, and the position of the particle should be inverted to 0. When $v_{jm}^{k} > 0$, the value of the historical optimal solution for the population of the optimal particle equals reversely 1.

The relationship between the particle position value and the optimal particle position of the population, according to the above method, can balance the local and global search ability at the later stage and make the result tend to the optimal particle of population and converge to the global optimal particle eventually.

Improvements of the Original BPSO

(1) If Iteration $< \lambda \times$ MaxIter, the velocity and position of particles will be updated. If Iteration $\geq \lambda \times$ MaxIter, according to (13), (14), and (15), the velocity and position will be updated.

(2)

$$s\left(v_{jm}^{k}\right) = \begin{cases} 1 - \dfrac{2}{1 + \exp\left(-v_{jm}^{k}\right)} & v_{jm}^{k} \leq 0 \\ \dfrac{2}{1 + \exp\left(-v_{jm}^{k}\right)} - 1 & v_{jm}^{k} > 0. \end{cases} \quad (13)$$

When $v_{jm}^{k} \leq 0$,

$$x_{jm}^{k} = \begin{cases} 0 & \text{rand}\,() \leq s\left(v_{jm}^{k}\right) \\ x_{jm}^{k} & \text{otherwise.} \end{cases} \quad (14)$$

When $v_{jm}^{k} > 0$,

$$x_{jm}^{k} = \begin{cases} 1 & \text{rand}\,() \leq s\left(v_{jm}^{k}\right) \\ x_{jm}^{k} & \text{otherwise.} \end{cases} \quad (15)$$

$\lambda \in [0, 1]$. Iteration is the number of iterations, and its maximum value is MaxIter.

IBPSO and the standard BPSO are different in the probability mapping function. The goal of the new probability mapping function is that the probability mapping function value is 0 when the velocity tends to 0, so the local optimization efficiency is enhanced. Then, by new probability function, the particle's bit value remains intact when it approximately equals 0. When $V < 0$, the position of the particle is set to 0. And when $V > 0$, it is set to 1. When the velocity is 0, the probability of particle's position change is close to 0, and its local search ability is enhanced. By this way, the particle swarm was converged to global optimal particle finally.

5. Reconfiguration of Distribution Network with DG

5.1. Distribution Network Reconfiguration Based on Dual Hybrid Particle Swarm Optimization Algorithm. To finish the distribution network reconfiguration, distribution network simplification, branch groups selection and optimization, branches selection and optimization in branch group, and power flow calculation under constraint conditions, dual hybrid particle swarm optimization algorithm is proposed in this paper. IBPSO was used in branch group search, and the proposed group BPSO was used in searching within the group. This is reconfiguration of distribution network by dual hybrid particle swarm optimization algorithm.

Take the IEEE 33 node distribution system as an example. Firstly, determine the matrix A, which is the relationship between branch group number and branch within group number, as shown in (16). Secondly, use 0 or 1 to describe branches' states within the group (open or closed) and set the vector of each branch group, as shown in (17). There is only one branch disconnected in every group when updating the speed and location. Thirdly, based on the disconnected branch group number, select the disconnected branch within group. The speed updating formula is given as (18) and the location formula is the same as before.

$$A = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 4 & 5 & 0 & 0 & 0 & 0 & 0 \\ 6 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9 & 10 & 11 & 0 & 0 & 0 & 0 & 0 \\ 12 & 13 & 14 & 0 & 0 & 0 & 0 & 0 \\ 15 & 16 & 17 & 29 & 30 & 31 & 32 & 36 \\ 34 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 22 & 23 & 24 & 37 & 0 & 0 & 0 & 0 \\ 25 & 26 & 27 & 28 & 0 & 0 & 0 & 0 \\ 18 & 19 & 20 & 0 & 0 & 0 & 0 & 0 \\ 33 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 21 & 35 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{16}$$

$$\begin{aligned} A1 &= [0]; \\ A2 &= [1, 0, 1]; \\ A3 &= [0, 1]; \\ A4 &= [0]; \\ A5 &= [1, 1, 0]; \\ A6 &= [0, 1, 1]; \\ A7 &= [1, 1, 1, 1, 0, 1, 1, 1]; \\ A8 &= [0]; \\ A9 &= [1, 0, 1, 1]; \\ A10 &= [1, 1, 0, 1]; \\ A11 &= [1, 0, 1]; \\ A12 &= [0]; \\ A13 &= [1, 0]; \end{aligned} \tag{17}$$

$$v_{imr}^{k+1} = \begin{cases} wv_{imr}^{k} + c_1 \text{rand}\left(p_{imr}^{k} - x_{imr}^{k}\right) + c_2 \text{rand}\left(g_{imr}^{k} - x_{imr}^{k}\right) & x_{im}^{k} = 0,\ p_{im}^{k} = g_{im}^{k} = 0 \\ wv_{imr}^{k} + c_1 \text{rand}\left(p_{imr}^{k} - x_{imr}^{k}\right) + c_2 v_{\max}(2\text{rand} - 1) & x_{im}^{k} = 0,\ p_{im}^{k} = 0,\ g_{im}^{k} = 1 \\ wv_{imr}^{k} + c_1 v_{\max}(2\text{rand} - 1) + c_2 \text{rand}\left(g_{imr}^{k} - x_{imr}^{k}\right) & x_{im}^{k} = 0,\ p_{im}^{k} = 1,\ g_{im}^{k} = 0 \\ wv_{imr}^{k} + c_1 v_{\max}(2\text{rand} - 1) + c_2 v_{\max}(2\text{rand} - 1) & x_{im}^{k} = 0,\ p_{im}^{k} = 1,\ g_{im}^{k} = 1 \\ v_{imr}^{k} & x_{im}^{k} = 1, \end{cases} \tag{18}$$

where v_{imr}^{k+1} is the speed of particle i at $k+1$th iteration, which in the branch of group m, p_{imr}^{k} and g_{imr}^{k} are particles' history optimal positions and population optimal position, and x_{imr}^{k} is connection condition of the branch r in group m at kth iteration.

5.2. Reconfiguration Flow of Distribution Network with DG. The IBPSO and BPSO in branch group are core algorithms in distribution network reconfiguration. The branch selection is optimized by IBPSO, while the branch group selection is optimized by BPSO within group. The detailed steps and the

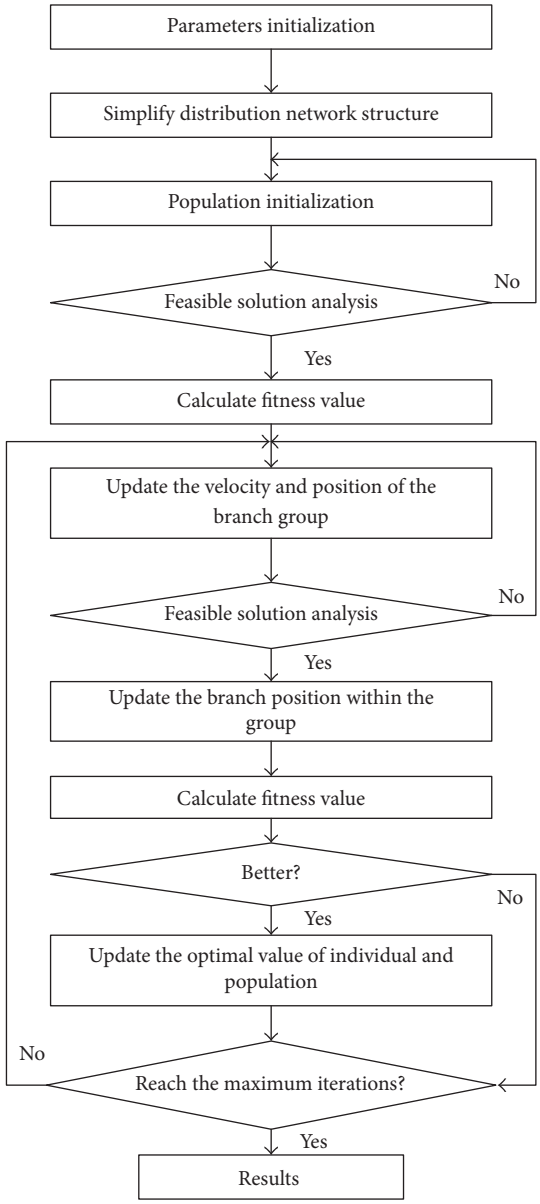


Figure 3: Flow chart of distribution network reconfiguration algorithm.

specific implementation flow chart of network reconfiguration algorithm are shown in Figure 3.

Step 1. Set the values of particle population number N, the number of groups, and all other parameters.

Step 2. According to the simplified structure of distribution network, the nodes and branches of the distribution network are renumbered. Set the internal branch number within branch group, such as the column number of the elements in a matrix A. Then, the particle dimension of branch group D and dimension of branches within group D' are determined, and then parameters of the population particles are initialized.

Table 1: Reconfiguration results without DG connected into grid.

Reconfiguration condition	Disconnect network switch	Minimum node voltage	Network loss/kW
Before reconfiguration	8-21 9-15 12-22 18-33 25-29	0.9185	202.65
Literature	7-8 9-10 14-15 28-29 32-33	0.9412	139.90
Algorithm in this paper	7-8 9-10 14-15 25-29 32-33	0.9378	139.47

Step 3. Update the speed and position of the branch group. If the structure of the distribution network is radial, go to the next step; otherwise, repeat this step.

Step 4. Update disconnected branch location within the group branch organization. According to the matrix A, find the power flow calculation in actual disconnected branch number and calculate its fit value.

Step 5. Compare the particle updated fitness value with the optimal particle fitness value in history, as well as the particle swarm optimum fitness value. If the current position is better, the value of current position will be selected as the particle's best individual historical fitness value and the particle swarm optimum fitness value. This position will be selected to update the best position of individual history and the best position of population in history.

Step 6. If the iteration number is less than the predetermined, go back to Step 3.

Step 7. If the iteration number reaches the predetermined, the optimum reconfiguration results are obtained and the program ends.

6. Case Studies

The proposed algorithms are used on the IEEE 33 node distribution system to verify the validity, convergence, and stability. IEEE 33 nodes network structure was shown in Figure 2(a). Assuming the particle population $N = 20$, the loop termination condition is

$$\max \left| U^k(i) - U^{k-1}(i-1) \right| \leq 0.000001. \tag{19}$$

As shown in Table 1, the active power loss of the network is 139.47 kW after reconfiguration. It is 31.18% lower than the active power loss before reconfiguration which is 202.65 KW. As shown in Figure 4, the overall voltage of the reconfigured network is generally higher than that of the network before reconstruction. The minimum value of node voltage rose from 0.9185 P.U. to 0.9378 P.U.

DG nodes were considered as PQ nodes for computation purpose, as in Table 2.

Table 3 shows the reconfiguration results with DG connected into grid by the algorithm in this paper. The active power loss of the network is 112.58 kW, which is 44.44% lower than active power loss before reconfiguration. As

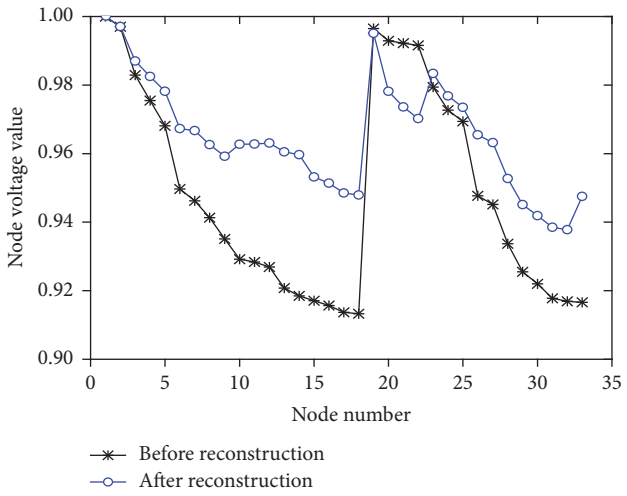


Figure 4: Voltage change tendency of nodes.

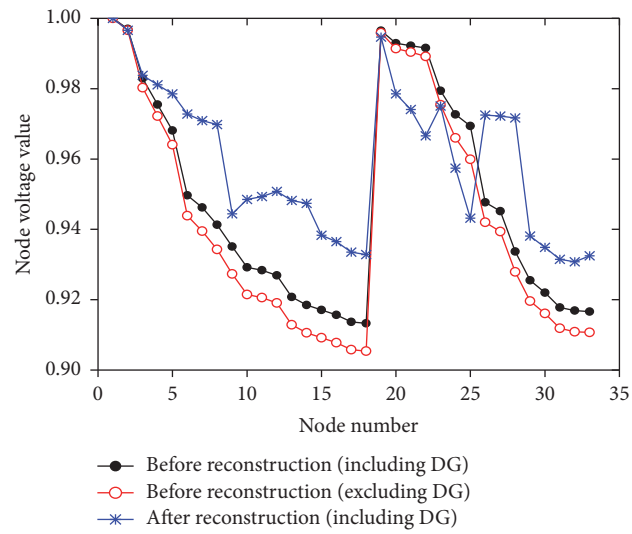


Figure 5: Nodes voltage change tendency with and without DG connected.

Table 2: Location and capacity of DG.

Distributed power serial number	Access point	Active power/kW	Power factor
DG1	7	100	0.8
DG2	9	50	0.9
DG3	22	50	0.85
DG4	25	250	0.9

Table 3: Reconfiguration results with DG connected into grid.

Reconfiguration condition	Disconnect network switch	Minimum node voltage	Network loss/kw
Before reconfiguration	8-21 9-15 12-22 18-33 25-29	0.9185	202.65
Before reconfiguration (DG connected)	8-21 9-15 12-22 18-33 25-29	0.9194	169.50
Literature	7-8 9-10 14-15 24-25 32-33	0.9482	115.04
Algorithm in this paper	8-9 8-21 14-15 28-29 32-33	0.9448	112.58

shown in Figure 5, the overall voltage of the reconfigured network is generally higher than that of the network before reconstruction. The minimum value of node voltage rises from 0.9185 P.U. to 0.9448 P.U. The reconfiguration result proves the superiority of the proposed algorithm, which was suitable to the reconfiguration of Distribution Network with DG.

Figure 6 shows the active power loss convergence curve of the network, which shows that the network loss converges when the iteration number was greater than 3. This proves that the proposed dual hybrid particle swarm optimization algorithm can quickly jump out of local optimal solution and converge to the optimal switch combination simultaneously.

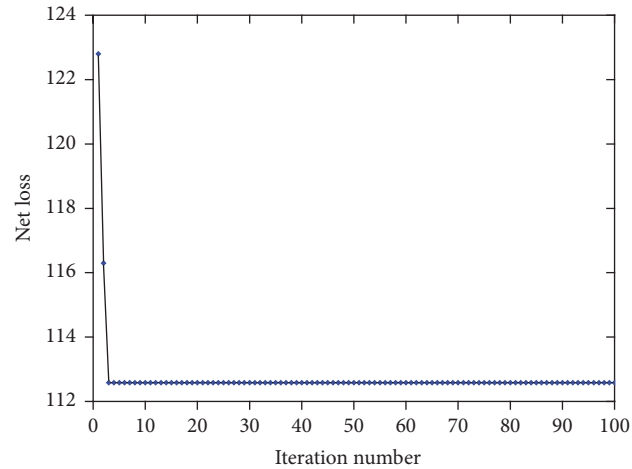


Figure 6: Loss convergence curve.

After 50 times simulations of the example by the computer with Windows 7 operating system, 2.20 GHz Intel CPU, 4.0 G memory, the results are shown in Figure 7, which shows the average number of iterations is 5 times, and the active power loss is about 112 kW.

DG with different types and capacities are selected to verify the validity of the algorithm, as shown in Table 4.

Table 5 shows that active power loss of the network is 112.77 kW after reconfiguration. It is 44.35% lower than the active power loss of the network before reconfiguration which is 202.65 KW. The minimum value of node voltage increases from 0.9185 P.U. to 0.9380 P.U. Figure 8 shows that the overall voltage of the reconfigured network is generally higher than that of the network before reconstruction.

After 100 times simulations of the example, the results are shown in Figure 9, and the average number of iterations is 16 times. The above analysis results show that the economy of

Table 4: Location and capacity of DG.

Distributed power serial number	Access point	Distributed power type	Parameter
DG1	4	Doubly fed induction generator (PQ)	P = 100 kW, Q = 100 kVar
DG2	17	Photovoltaic cells (PI)	P = 150 kW, I = 10 A
DG3	25	Fuel cell (PV)	P = 200 kW, U = 12.66 kV
DG4	30	Asynchronous fan ($PQ(V)$)	P = 150 kW, Q = 100 kVar

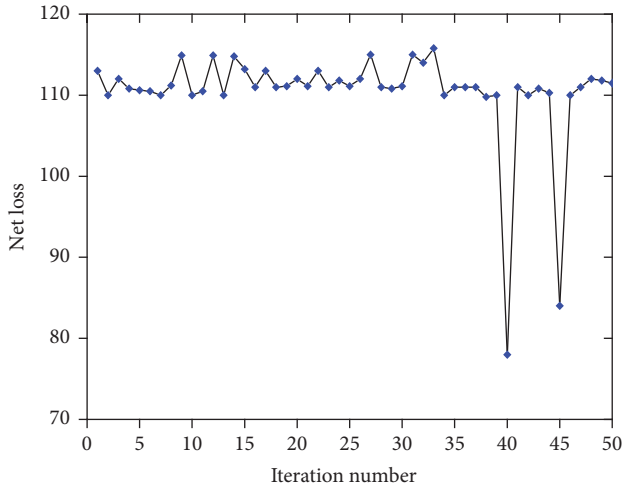


Figure 7: Stability analysis of double hybrid PSO method.

Table 5: Reconfiguration results with DG connected.

Reconfiguration condition	Disconnect network switch	Minimum node voltage	Network loss/kW
Before reconstruction	8-21 9-15 12-22 18-33 25-29	0.9185	202.65
Before reconfiguration (including DG)	8-21 9-15 12-22 18-33 25-29	0.9274	175.62
After reconfiguration (including DG)	11-12 28-29 30-31 8-21 9-15	0.9380	112.77

the system was improved, and the stability and reliability of the system were enhanced after reconfiguration.

7. Summary

Based on dual hybrid particle swarm algorithm combined with the network structure simplification and branches grouping, this paper deals with the problem of distribution systems reconfiguration. The proposed algorithm is tested on the IEEE 33-bus distribution power system and compared with other existing literature methods. The simulation results show that the proposed algorithm has a faster optimization speed for the distribution network with DG, which can not only reduce the network loss and support system voltage but also improve the economy and reliability of the system. The proposed algorithm was suitable to search for the best switch combination of distribution network with DG.

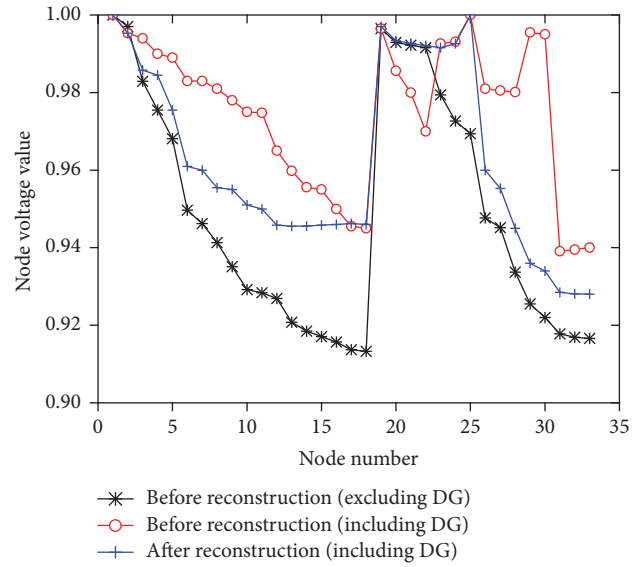


Figure 8: Nodes voltage with and without DG connected.

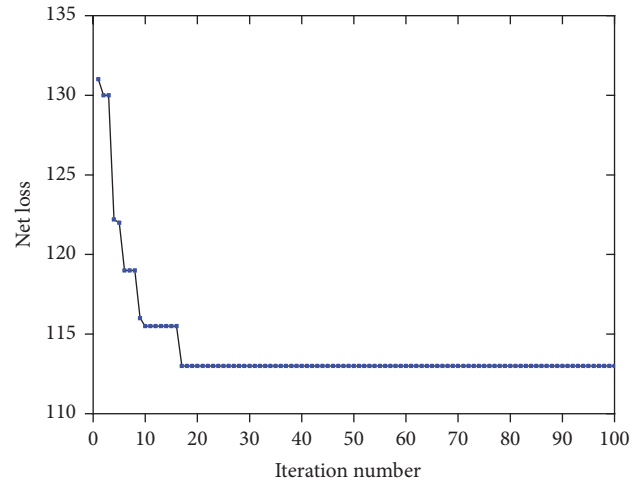


Figure 9: Loss convergence curve.

Distributed power grid connected is an inevitable trend with the development of smart city and modern power grid technology. It is of great significance for energy saving and emission reduction and improving energy structure. In this paper, the reconfiguration of smart grid with distributed generation is studied. From the simulation results, it can be seen that after the reconfiguration of the distributed power grid, it can effectively reduce the loss of distribution network,

improve the quality of the power supply voltage, and improve the power quality of the grid. The method of research and development of this paper has certain reference function for the optimization of power grid configuration in the future.

Conflicts of Interest

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

Acknowledgments

This work is supported by the Jiangsu Provincial Natural Science Foundation of China (Grant no. BK20130187) and China Scholarship Council (CSC no. 201406425007).

References

[1] F. Z. Peng, "Special issue on distributed power generation," *IEEE Transactions on Power Electronics*, vol. 19, no. 5, pp. 1157-1158, 2004.

[2] E. M. Carreno, R. Romero, and A. Padilha-Feltrin, "An efficient codification to solve distribution network reconfiguration for loss reduction problem," *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1542–1551, 2008.

[3] W. El-Khattam and M. M. A. Salama, "Distributed generation technologies, definitions and benefits," *Electric Power Systems Research*, vol. 71, no. 2, pp. 119–128, 2004.

[4] W. Hongjian and X. Lei, "The dynamic reconfiguration of distribution network with wind turbine and electric vehicle based on genetic algorithm," *Journal of Electrical Engineering*, vol. 31, no. 2, pp. 197–220, 2016.

[5] W. Jiajia, L. Lin, L. Junyong et al., "Based on improved hierarchical forward and backward substitution method with distributed generation units of distribution network reconfiguration," *Power System Technology*, vol. 34, no. 3, pp. 109–112, 2010.

[6] C. Li, Q. Yang, Y. Wei, H. Luo, and Q. Zhang, "Distribution network reconfiguration method considering the correlation between wind speed and load," *Dianli Zidonghua Shebei/Electric Power Automation Equipment*, vol. 36, no. 2, pp. 148–153, 2016.

[7] X. Chen and J. weighted, "Application of the mutation particle swarm optimization algorithm in the vehicle routing problem," *Computer And Digital Engineering*, vol. 34, no. 9, pp. 19–21, 2006.

[8] T. Jen-Hao and M. L. Whe, "Current-based power flow solutions for distribution systems," ICPST Beijing, China, l994: 414–418.

[9] R. D. Zimmerman and H.-D. Chiang, "Fast decoupled power flow for unbalanced radial distribution systems," *IEEE Transactions on Power Systems*, vol. 10, no. 4, pp. 2045–2052, 1995.

[10] Y. L. Mei, X. Mingxia, X. Jianjun et al., "Power flow algorithm for distribution network with distributed generation," *Power System Protection And Control*, vol. 41, no. 5, pp. 18–22, 2013.

[11] X. Xiang, D. Liu, N. Xiang, and B. Wang, "Distribution network reconfiguration based on parallel Tabu search algorithm," *Power System Technology*, vol. 36, no. 8, pp. 101–105, 2012.

[12] X. Lixiong, L. Lin, and L. Junyong, "Reconfiguration of distribution network based on improved particle swarm optimization algorithm," *Automation of Electric Power Systems*, vol. 30, no. 7, pp. 27–30, 2006.

[13] W.-C. Wu and M.-S. Tsai, "Application of enhanced integer coded particle swarm optimization for distribution system feeder reconfiguration," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 1–9, 2011.

[14] L. Zhenkun et al., "A hybrid particle swarm optimization algorithm for distribution network reconfiguration," *Proceedings of the Chinese Society of Electrical Engineering*, vol. 28, no. 31, pp. 35–41, 2008.

[15] R. Huaipu, S. Siqing, and W. Xiaowei, "Based on the improved binary particle swarm optimization algorithm of network reconfiguration," *Power Grid And Clean Energy and Infected*, vol. 27, no. 8, pp. 41–43, 2011.

[16] V. Calderaro, A. Piccolo, and P. Siano, "Maximizing DG penetration in distribution networks by means of GA based reconfiguration," in *Proceedings of the 2005 International Conference on Future Power Systems*, Netherlands, November 2005.

[17] D. I. H. Sun, S. Abe, R. R. Shoults, M. S. Chen, P. Eichenberger, and D. Farris, "Calculation of energy losses in a distribution system," *IEEE Transactions on Power Apparatus and Systems*, vol. 99, no. 4, pp. 1347–1356, 1980.

[18] Z. Fang and C. S. Cheng, "A modified newton method for radial distribution system power flow analysis," *IEEE Transactions on Power Systems*, vol. 12, no. 1, pp. 389–397, 1997.

[19] W. Jincheng, W. Jiesheng, and W. Wang, "Parameter self-tuning PID control and decision based on Improved Particle Swarm Optimization (PSO)," 2005,20 (1):73–76.

[20] D. P. Bernardon, A. P. C. Mello, L. L. Pfitscher, L. N. Canha, A. R. Abaide, and A. A. B. Ferreira, "Real-time reconfiguration of distribution network with distributed generation," *Electric Power Systems Research*, vol. 107, pp. 59–67, 2014.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 2048475, 12 pages
https://doi.org/10.1155/2017/2048475

Research Article

A Developed Optimization Method of Tight Reservoir Porosity

Fenqiang Wu[1,2,3] and Jingling Xu[1,2,3]

[1]*Key Laboratory of Geo-Detection, China University of Geosciences, Ministry of Education, Beijing 100083, China*
[2]*State Key Laboratory of Geological Processes and Mineral Resources, China University of Geosciences, Beijing 100083, China*
[3]*School of Geophysics and Information Technology, China University of Geosciences, Beijing 100083, China*

Correspondence should be addressed to Jingling Xu; jlxu@cugb.edu.cn

Received 26 November 2016; Accepted 19 June 2017; Published 18 July 2017

Academic Editor: David Greiner

The optimization method to evaluate the tight reservoir porosity is a difficult technique to use due to its complexity and instability. This paper proposes an improved optimization method to calculate the porosity of tight reservoirs. First, we applied the matrix model which modified the multicomponent model to the problem and it improved the results by deducing a mathematical model. Second, we used the Simulated Annealing Algorithm to calculate the incoherence function, and then based on statistical theory, we obtained the most optimal results. Examples show that the method is effective, and despite the lack of the local experience parameters, its application is valuable in order to evaluate the porosity.

1. Introduction

The optimization log interpretation method provides a variety of logging information which is an effective way to evaluate complex oil and gas reservoirs. After it was proposed in 1980 [1], optimization log interpretation method technology has developed and improved. Although the firm Schlumberger Ltd. boasted about how the technology could be suitable for all types of reservoirs around the world, it was difficult to obtain a satisfactory result when evaluating tight reservoirs. Because the pore system in tight reservoirs presents a much greater variety of geometrical characteristics and structures compared with clastic reservoir, then the complexity of the pore system affects the results because the parameter is difficult to ascertain. Therefore, the development and improvement of the optimization log interpretation method to solve the above problem are required.

The traditional optimization log interpretation method contains two important processes: (1) the establishment of the pore model and the response equations and (2) the calculation of the response equations.

The traditional pore model is a multimineral model or multicomponent model [2, 3]. In the model, the reservoirs act as a unit of a space geological body and are made up of partially homogeneous components, such as minerals and porosity. The purpose of this division is to simplify the calculation process and programming design [4]. However, arguments presuppose that all types of minerals and geologic parameters in multimineral or multicomponent models must be known. Those models cannot be used in geographical areas where the mineral components are difficult to distinguish, especially at tight reservoirs. In this paper, we put forward the mixed-matrix model to improve traditional model following the Fuzzy Clustering analysis. An important characteristic of this new model is synthesizing multimineral as a synthesis. For instance, with this model, the user does not need to know the physical parameters of all the types of mineral present in the system. Therefore, the response equations are calculated for the optimization log interpretation mathematical model.

The calculation of the response equations is the optimization algorithm. In the process of algorithm research, a large number of scholars have used different algorithms to propose mathematical models as it is shown in Table 1 [5]. For example, Global Program started using the steepest descent algorithm [1], and Optima Program used the conjugate gradient algorithm [6]. These programs were a good approach evaluating conventional reservoir wells. In China, Ouyang et al. used the simplex algorithm to evaluate clastic reservoir [7]

Table 1: Classification optimization algorithms.

Unconstrained optimization algorithms									
Cyclic Coordinate	Steepest descent		Newton's algorithm	Conjugate gradient		Powell's algorithm	BFGS or DFP		Simplex
Constrained optimization algorithms									
Random floating	Random searching	Complex	CONMIN	Sequential linear programming	GRG	Interior penalty	Exterior Penalty	Mixed penalty	SQP
Intelligent algorithm									
Local search	Simulated annealing		Genetic algorithm	Tabu search algorithm			Artificial neural network		PSO

For example, BFGS or DFP is quasi-Newton methods; GRG is penalty function method; SQP is sequence quadratic program; PSO is particle swarm optimization.

and lamprophyre reservoir [8], using BFGS or DFP to evaluate oil-bearing igneous for better results [9]. All of the above optimization algorithms have an initial value and the initial value greatly influenced the results [10]. Stoffa and Sen introduced the genetic algorithm (GA), in the optimization algorithms [11], and GA does not need the initial value of the input which means GA is not restricted by the initial value [12]. Unfortunately, GA was not stable method to use due to the repeated calculation it involves. An important improvement in the algorithm would be to eliminate the influence of the initial value and the computational stability. Kirkpatrick et al. used the simulated annealing to solve a similar problem [13]. In this paper, we apply the Simulated Annealing Algorithm (SAA) to calculate the response equations, because not only is it a global optimization algorithm for functions of continuous variables but also it depends slightly on the initial value [14].

Firstly, we start comparing the correlation between well logging and core porosity to select the well logging types. Secondly, we have an introduction on mixed-matrix model to improve multimineral model. Thirdly, we use SAA for optimization and then we use the Monte Carlo random method to select the initial parameters of the optimal values. The method was then tested with set data from a tight reservoir. Finally, we conclude with a discussion of advantages and limitations of the method.

2. The Improved Optimization Method

2.1. The Selection of Sensitive Logging Curves. The use of more logging curves is likely to prove more reliable results [15]. However, it becomes less reliable with increasing the response equations and increasing the parameters [16]. Thus, it is crucial to select the more correlated logging curves as possible.

The porosity logs available for this study were obtained from ten wells, the parameters cover acoustic (AC), density (DEN), compensated neutron log (CNL), and natural gamma rays (GR), among others. The porosity datum from the ten wells come from 1773 samples. The data was obtained from tight sandstone and shales reservoir. And then correlations were calculated using the core porosity data and AC, DEN, CNL, and GR as shown in Figures 1(a)–1(d). The overall porosity ranged from 0% to 12% (average is 6.48%), the overall AC is within the scope of 156.8 us/m to 338.9 us/m (average is 225.5 us/m), the overall DEN was in a scale of 2.17 g/cm^3 to 2.99 g/cm^3 (average is 2.54 g/cm^3), the overall CNL was from 2.9% to 39.3% (average is 15.9%), and the overall GR ranged from 10 API to 330 API (average is 90 API).

In Figures 1(a)–1(d), the correlation coefficients between the core porosity and the well log value of AC, DEN, CNL, and GR are 0.1854, 0.2198, 0.2634, and 0.069, respectively. The AC, DEN, and CNL (triporosity loggings) showed a higher correlation coefficient. In this paper, we selected AC, DEN, and CNL to calculate the porosity, similar to the usual optimization interpretation used in conventional reservoir.

2.2. The Establishment of the Mixed-Matrix Model. In order to solve the multimineral model or the multicomponent model, which cannot be used in geographical areas where the mineral components are difficult to distinguish especially the tight reservoir, we modified multicomponent model to the mixed-matrix model in this research. Figure 2 shows the following:

(1) In the model, the complex formation as a unit of a space geological body is made up of three components, which are V_{ma}, V_{cl}, and ϕ.

(2) V_{ma} is the mixed-brittle matrix composed of V_{QUA}, V_{FEL}, V_{CAL}, V_{DOL}, and so on.

(3) V_{cl} is the mixed-clay matrix mainly composed of V_{KAO}, V_{MON}, V_{CHL}, and so on.

(4) ϕ is total porosity mainly composed of V_{OG} and V_W.

Three kinds of component are determined to reduce the volume for the target porosity to build the linear equation

$$\begin{bmatrix} \Delta t \\ \rho \\ \Phi_N \end{bmatrix} = \begin{bmatrix} \Delta t_{ma} & \Delta t_{cl} & \Delta t_{\phi} \\ \rho_{ma} & \rho_{cl} & \rho_{\phi} \\ \Phi_{Nma} & \Phi_{Ncl} & \Phi_{N\phi} \end{bmatrix} \cdot \begin{bmatrix} V_{ma} \\ V_{cl} \\ \phi \end{bmatrix}. \quad (1)$$

2.3. The Establishment of Incoherence Function and Constraints. The building of optimization log interpretation

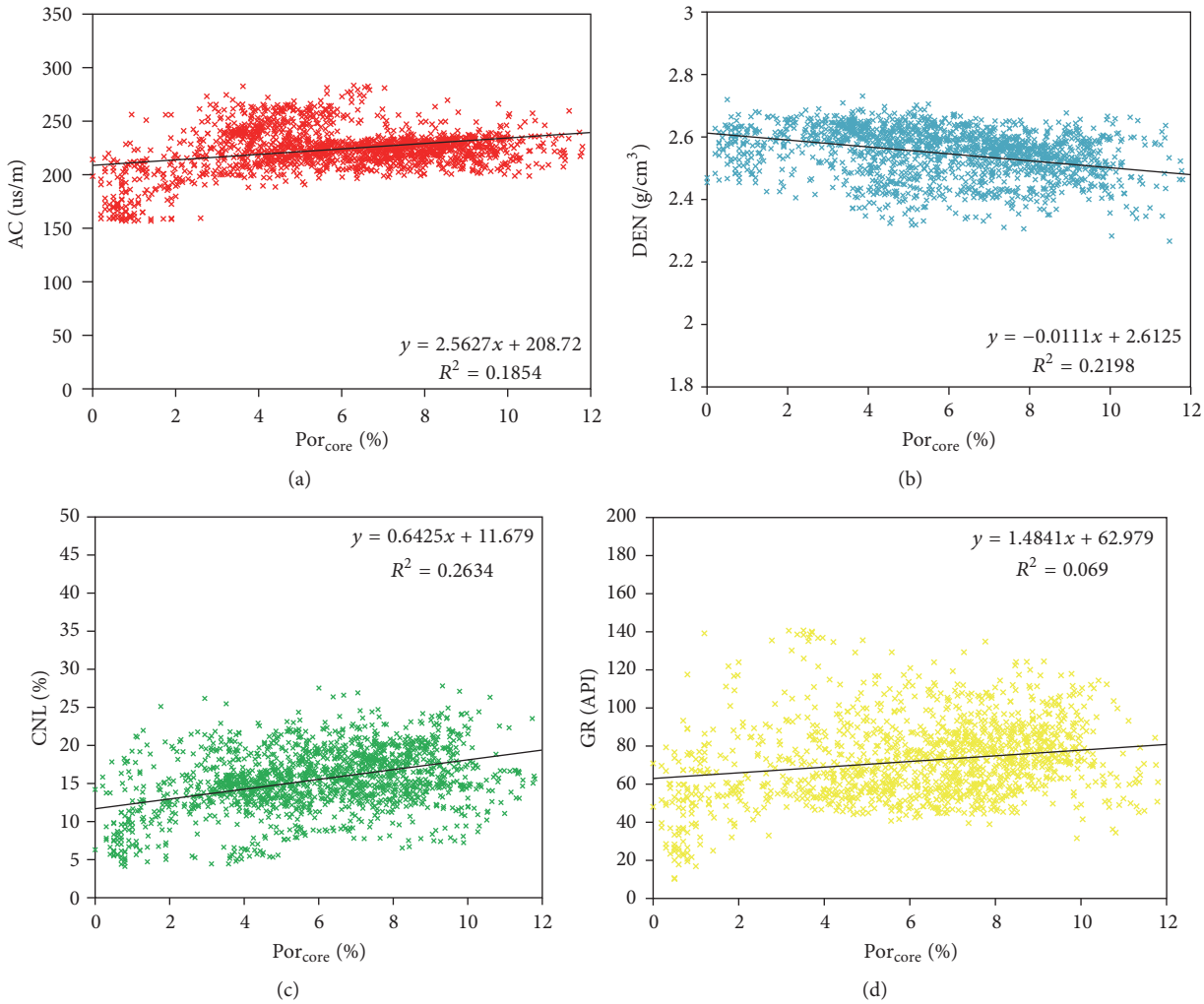


Figure 1: The crossplot between core data and well logging. (a) The crossplot of core porosity and acoustic. (b) The crossplot of core porosity and density. (c) The crossplot of core porosity and neutron. (d) The crossplot of core porosity and natural gamma ray.

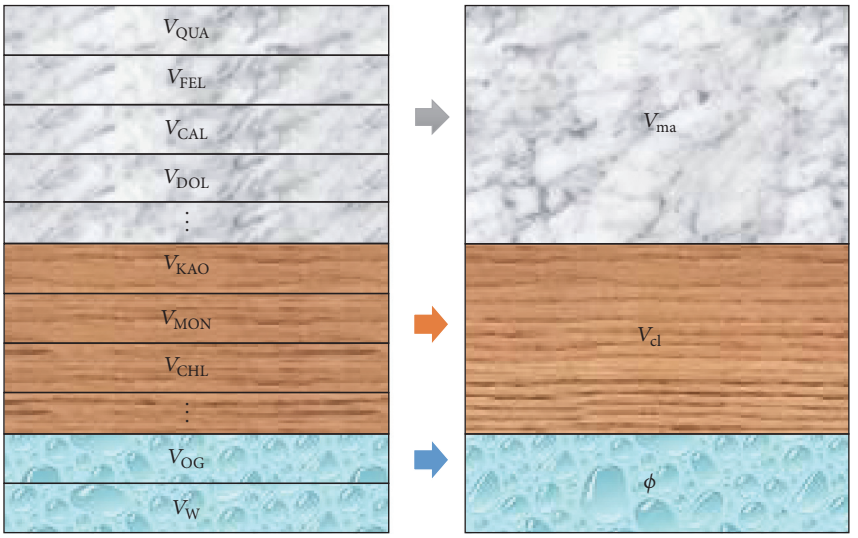


Figure 2: Mixed-matrix model based on the improved multimineral model.

Table 2: The physical parameter range.

Value range	ma_{min}	ma_{max}	cl_{min}	cl_{max}	ϕ_{min}	ϕ_{max}
Δt (us/m)	114.83	196.85	213.25	360.89	620	656
ρ (g/cm^3)	2.04	2.87	2.02	3	0.8	1.1
Φ_N (%)	−5	3.4	10	52	90	100

Table 3: The samples of triporosity logging data and calculated results.

Point	AC (us/m)	DEN (g/cm^3)	CNL (%)	Por_{core} (%)	Por_{ave} (%)	Por_{sd} (%)
a	207.875	2.636	11.959	2.77	2.78	1.47
b	216.057	2.486	14.887	3.99	3.95	1.86
c	246.092	2.519	11.798	4.38	4.34	1.88
d	260.563	2.635	20.680	5.74	5.75	2.11

mathematical model is the basis of the optimization log interpretation, including the establishment of incoherence function and constraints.

According to the principle of least squares methods, the incoherence function of optimization interpretation is established as

$$F'(v) = \sum_i \left[c_i - f_i(v)\right]^2. \tag{2}$$

Extraordinarily, each logging data has different order of magnitude, so the data should be processed. The procession is prior to normalization, which is for the different measuring dimension. The incoherence equation (2) turns into

$$F(v) = \sum_i \left(1 - \frac{f_i(v)}{c_i}\right)^2. \tag{3}$$

After establishing the incoherence function, the constraints also need to be defined to ensure the practical significance. Its constrains are shown on

$$\begin{aligned} 1 &= V_{ma} + V_{cl} + \phi, \\ 0 &\le V_{ma} \le 100\%, \\ 20\% &\le V_{cl} \le 100\%, \\ 0 &\le \phi \le 12\%. \end{aligned} \tag{4}$$

2.4. The Selection of the Geologic Parameters in Mixed-Matrix Model. The conventional reservoir matrix is mainly composed of quartz, feldspar, calcite, and dolomite; however, the tight reservoir matrix composition is hard to be determined. In mixed-matrix model, we modified the components as V_{ma}, V_{cl}, and ϕ. The logging parameters of each component are no longer a certain value but an interval, as shown in Table 2. For example, the value of Δt_{ma} is determined by the relative content of quartz, feldspar, calcite, dolomite, and so on. At the same time, the minimum incoherence function was obtained like all the other parameters; therefore, the components and physical parameters can be calculated.

2.5. The Simulated Annealing Algorithm and Monte Carlo Initial Value. Szucs and Civan used the simulated annealing method to interpret the multilayer well log [17]; however, the method required layering of reservoirs in advance. In order to find approximate minimum of the incoherence function, we present the Simulated Annealing Algorithm (SSA) combined with Monte Carlo random method to find approximate results. SAA is essentially an iterative random search procedure with adaptive moves along the coordinate directions, depending only slightly on the starting point. It allows uphill moves under the control of a probabilistic criterion, thus tending to avoid the first local minima encountered.

As is shown in Figure 3, SSA proceeds iteratively: starting from a given point v_0 which is randomly generated by Monte Carlo random method, and it generates a succession of points: $v_0, v_1, \ldots, v_i, \ldots$, tending to the global minimum of the incoherence function. New candidate points are generated around the current point v_i applying random moves along each coordinate direction, in turn. If the point falls outside the definition domain, a new point is randomly generated by Monte Carlo random method until a point belonging to the definition domain is found. A candidate point v' is accepted or rejected according to the Metropolis criterion [18].

If $\Delta F \le 0$, then accept the new point: $v_{i+1} = v'$;

else accept the new point with probability: $p_{\Delta f} = e^{-\Delta f/T}$,

where $\Delta F = f(v') - f(v_i)$, T is a parameter called temperature and the value must be higher than 0, N is the termination cycle, N_T is temperature reduction, and f' is the error tolerance for the function.

We wrote the optimization algorithm using MATLAB software according to the schematic diagram on Figure 3 [19].

2.6. The Calculation Results of Normal Distribution. Four samples were chosen from the 1773 samples (Table 3) to illustrate and eliminate the differences. In Table 3, each of them was repeated 100 times as shown in Figure 4.

In Figure 4, the abscissa is the calculation porosity and the ordinate is the frequency of the results. Figures 4(a)–4(d)

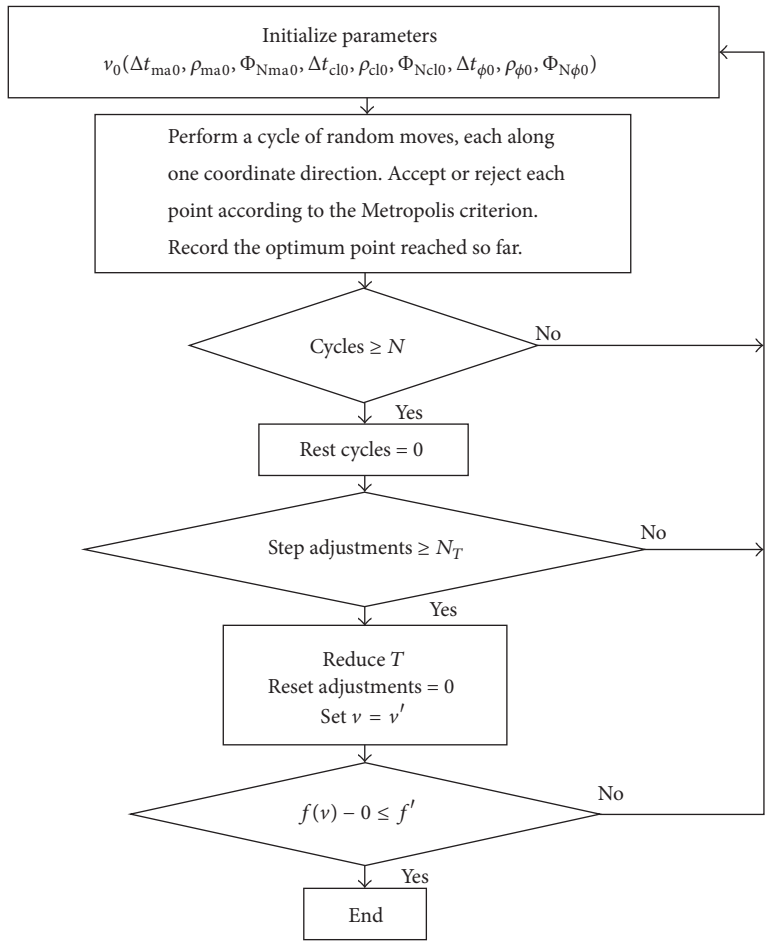


Figure 3: The schematic diagram of SSA minimization.

Table 4: The distribution of calculation results.

Point	a	b	c	d	The theoretical value
$[\mu - \sigma, \mu + \sigma]$	74.00%	77.00%	67.00%	76.00%	68.3%
$[\mu - 2\sigma, \mu + 2\sigma]$	93.00%	91.00%	93.00%	94.00%	95.4%
$[\mu - 3\sigma, \mu + 3\sigma]$	98.00%	97.00%	99.00%	97.00%	99.7%

present that Por_{core} are 2.77%, 3.99%, 4.38%, and 5.74%, respectively, Por_{ave} are 2.78%, 3.95%, 4.34%, and 5.75% respectively, and Por_{sd} are 1.47%, 1.86%, 1.88%, and 2.11%, respectively, which means that Por_{ave} is very close to Por_{core}; therefore, Por_{sd} is almost half of Por_{core}. The calculated results trend is a normal-like distribution. According to the Pauta criterion, theoretically, the probability of one to three standard deviations is 68.3%, 95.4%, and 99.7% in Table 4.

2.7. The Iterate Index. The normal-like distribution will help eliminate the differences or make computational stability. We can calculate many times and average the results to make computational stability. Apparently, the iterate index can lead to prolonging computation time. Therefore, it is crucial to determine the iterate index of calculation. We chose X5 well 3671 m–3679 m for the test.

Figure 5 presents that the iterate indexes are 5, 30, 50, and 100. It can be noted that (a) Por_{ave} tends to be stable gradually with increasing the iterate index. (b) The stable Por_{ave} tends to Por_{core} with increasing the iterate index. (c) Por_{ave} does not change much as the iterate index is up to approximately 30.

Figure 6 shows that it has been calculated 5 times when the iterate index is 30. Figure 6 shows the repeated calculation using the previously mentioned approach. In this case, each calculation is not the same but the overall trend is resemblance.

In Figure 7, the four samples were designed for the convergence which were calculated 100 times. Figures 7(a)–7(d) present that convergences are 2.78%, 3.95%, 4.34%, and 5.75%, respectively.

3. Applications and Contrast

3.1. The Influence of Initial Value. Using the last sampling point to be the next point initial value is helpful technique used by different studies, for instance, Yong and Sun [20]. Figure 8 shows the results of using Yong's initial method for

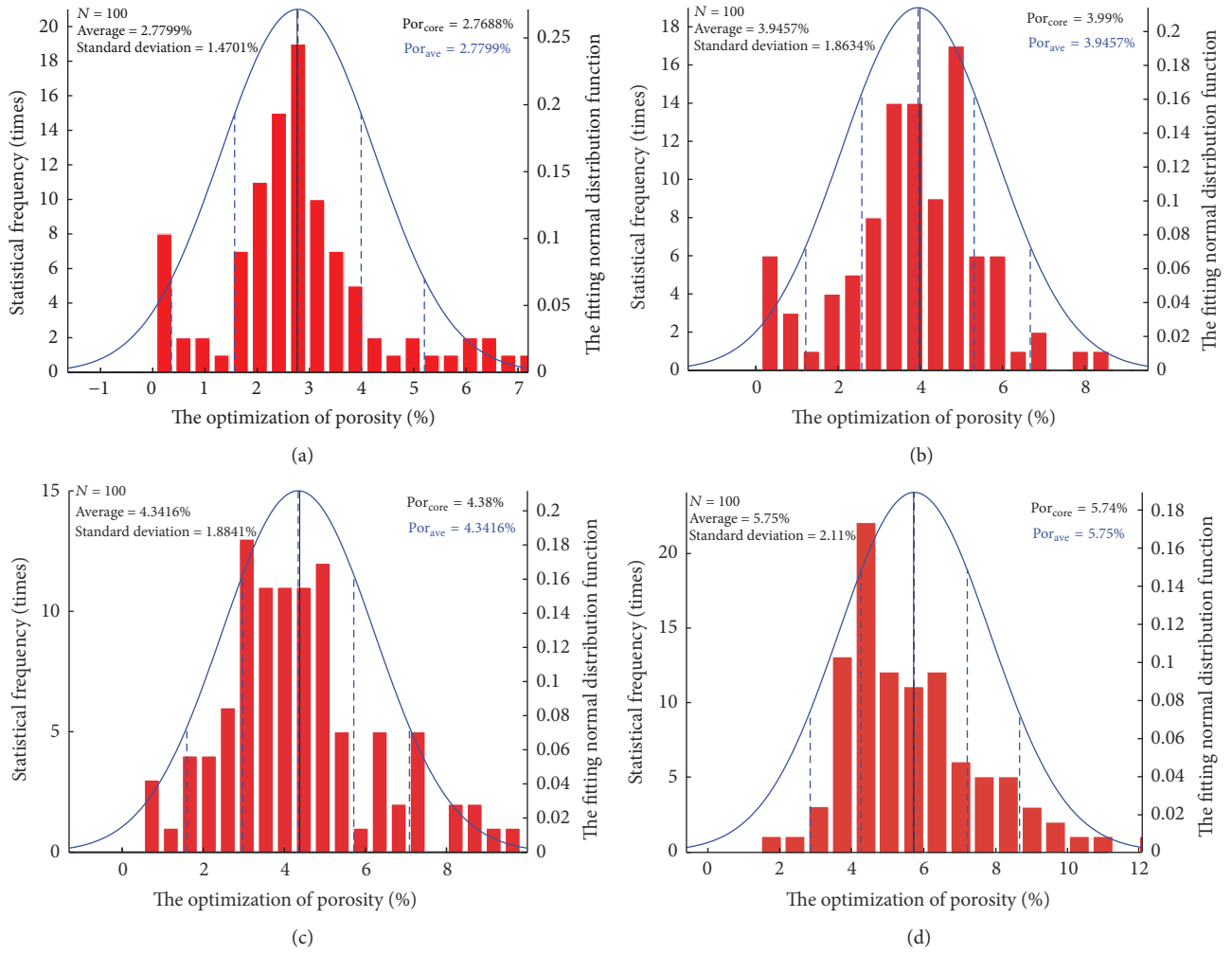


FIGURE 4: The calculation results of the statistical distribution. (a) The statistical distribution of point a. (b) The statistical distribution of point b. (c) The statistical distribution of point c. (d) The statistical distribution of point d.

the 1773 samples in this study (Figure 8(a)) and also by Monte Carlo random method (Figure 8(b)); the iterate index used was 30 for the graphics, where Por_Y is the calculation porosity by Yong's method and Por_{MC} is the calculation porosity by Monte Carlo random method.

The abscissa is the calculation porosity and the ordinate is Por_{core}, as shown in Figure 7. It is clear that the correlation coefficient is 0.58 and it increases to 0.8261 from Por_Y to Por_{MC}. Therefore, we believe that Monte Carlo random method is better than Yong's method.

3.2. The Contrast with Other Methods. Mixed-matrix model is based on multicomponent model. The contrast of the two types of models has great significance due to their strong connection. We chose JY1 well 2330 m–2352 m which belongs to the shale gas reservoir. We used Wyllie formula, multicomponent model, and our mixed-matrix model to calculate the porosity as shown in Figure 9.

From Table 5, the results differ slightly with the core which is due mainly to the 2340 m interval. Figure 9 reveals the different porosity by different methods, and mixed-matrix method is better than multicomponent model and multicomponent model is better than Wyllie formula.

TABLE 5: The contrast with other methods.

Method	Minimum value	Maximum value	Average value
Wyllie formula	3.05%	18.42%	10.78%
Multimineral model	3.96%	12.15%	5.94%
Mixed-matrix model	3.47%	11.70%	5.18%
Por_{core}	2.43%	7.03%	4.61%

3.3. The Physical Parameters. As it is shown in (1), there are two sorts of the parameters with three kinds of components (V_{ma}, V_{cl}, and ϕ) and also the physical parameters (Δt_{ma}, ρ_{ma}, Φ_{Nma}, Δt_{cl}, ρ_{cl}, Φ_{Ncl}, Δt_{ϕ}, ρ_{ϕ}, and $\Phi_{N\phi}$). It is also significant regarding the other parameters except for ϕ. We

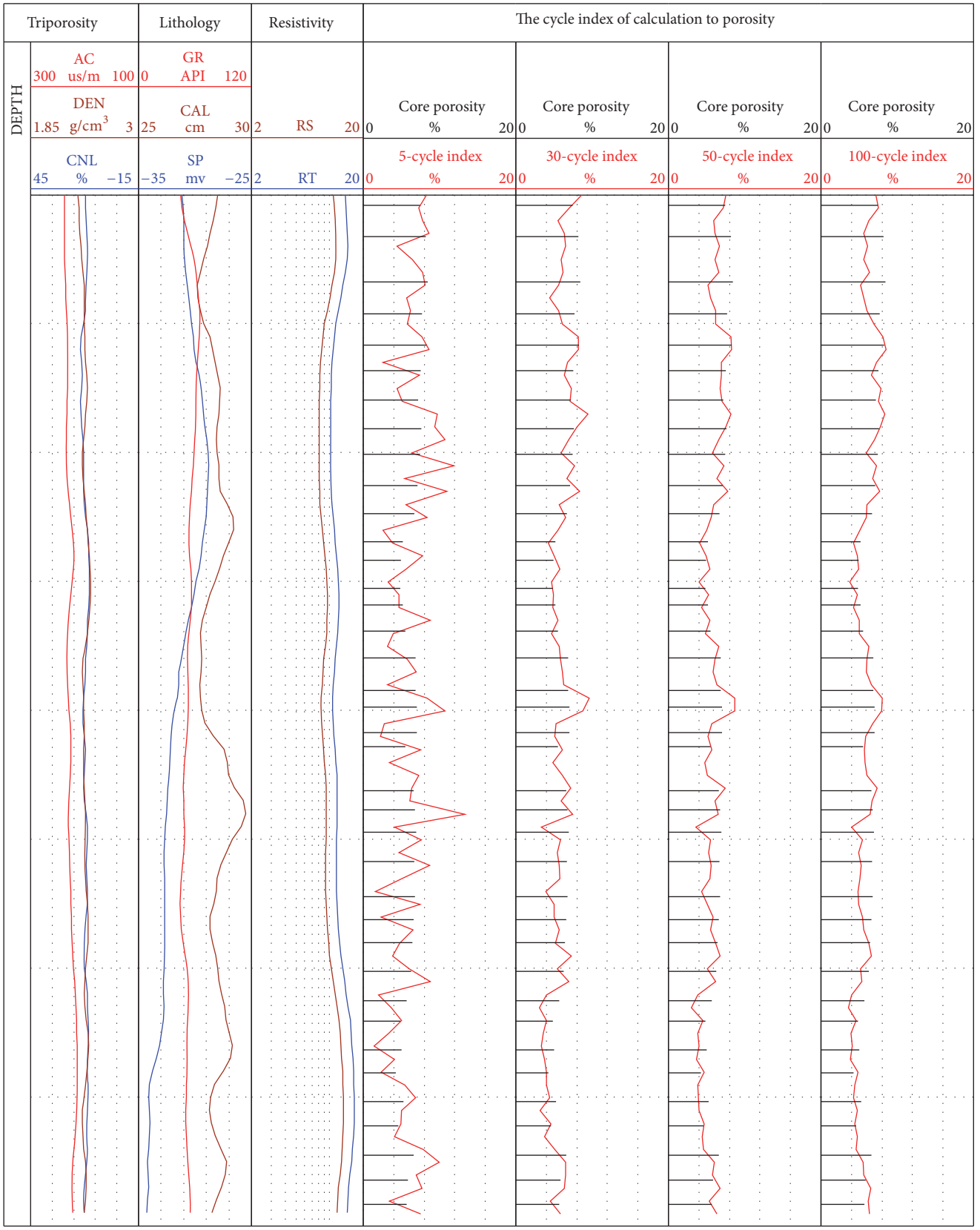


Figure 5: The influence of the cycle index for calculation.

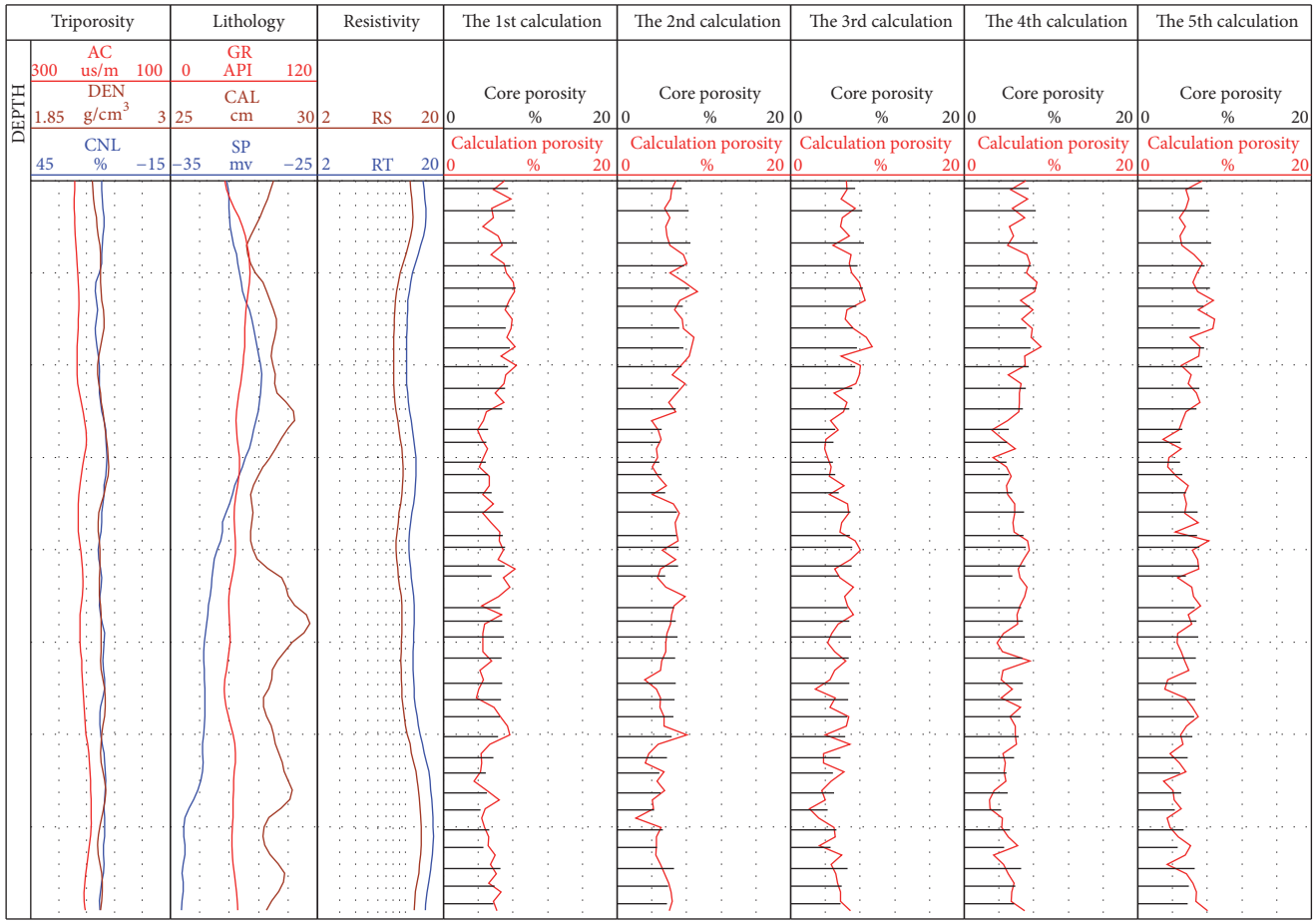


Figure 6: The stable of the method.

Table 6: The range of calculation results and the core.

	V_{ma}	V_{cl}	ϕ
The core data	$36.03\% < V_{ma} < 58.76\%$ (average is 41.73%)	$37.57\% < V_{cl} < 59.04\%$ (average is 52.88%)	$3.68\% < \phi < 6.68\%$ (average is 5.39%)
The calculation data	$33.95\% < V_{ma} < 54.91\%$ (average is 36.88%)	$41.19\% < V_{cl} < 61.27\%$ (average is 57.38%)	$3.89\% < \phi < 7.43\%$ (average is 5.72%)

chose JY1 well 2330 m–2352 m for the test and the iterate index is 30.

Table 6 reveals that V_{ma}, V_{cl}, and ϕ can be compared with the core, and Figure 10 presents that the physical parameters $v(\Delta t_{ma}, \rho_{ma}, \Phi_{Nma}, \Delta t_{cl}, \rho_{cl}, \Phi_{Ncl}, \Delta t_{\phi}, \rho_{\phi}, \Phi_{N\phi})$ conform to the actual situation.

4. Discussions and Conclusions

4.1. Mixed-Matrix Model from the Original Model. Tight reservoirs study is complex; they are composed of many kinds of minerals, which are difficult to be determined. Various factors should be considered in the establishment of the reservoir model, so the multicomponent model should be modified to mixed-model, in order to obtain more accurate reservoir parameters.

Mixed-matrix model is based on the ideas of Fuzzy Clustering. Comparing the classic multicomponent or multimineral models, the brittleness minerals are fuzzy as mixed-brittle matrix, the clay minerals are fuzzy as mixed-clay matrix, and the oil and gas and water porosity are fuzzy as the pore space volume. This improvement makes the approach easier and the user did not require all types of mineral's physical parameters.

4.2. Simulated Annealing Algorithm and the Normal Distribution. The Simulated Annealing Algorithm can provide a high reliability in the minimization of incoherence function.

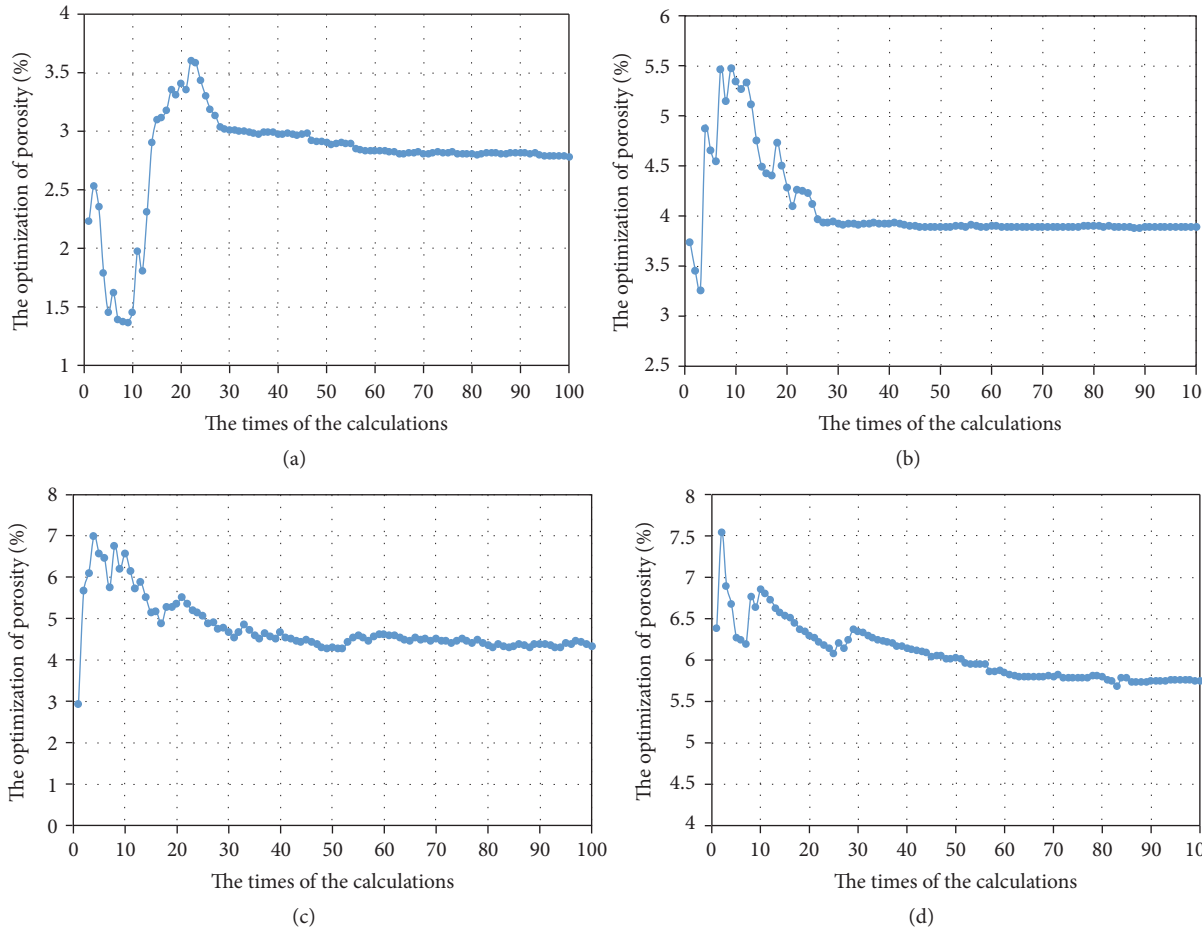


Figure 7: The 100 times calculations convergence. (a) The convergence of point a. (b) The convergence of point b. (c) The convergence of point c. (d) The convergence of point d.

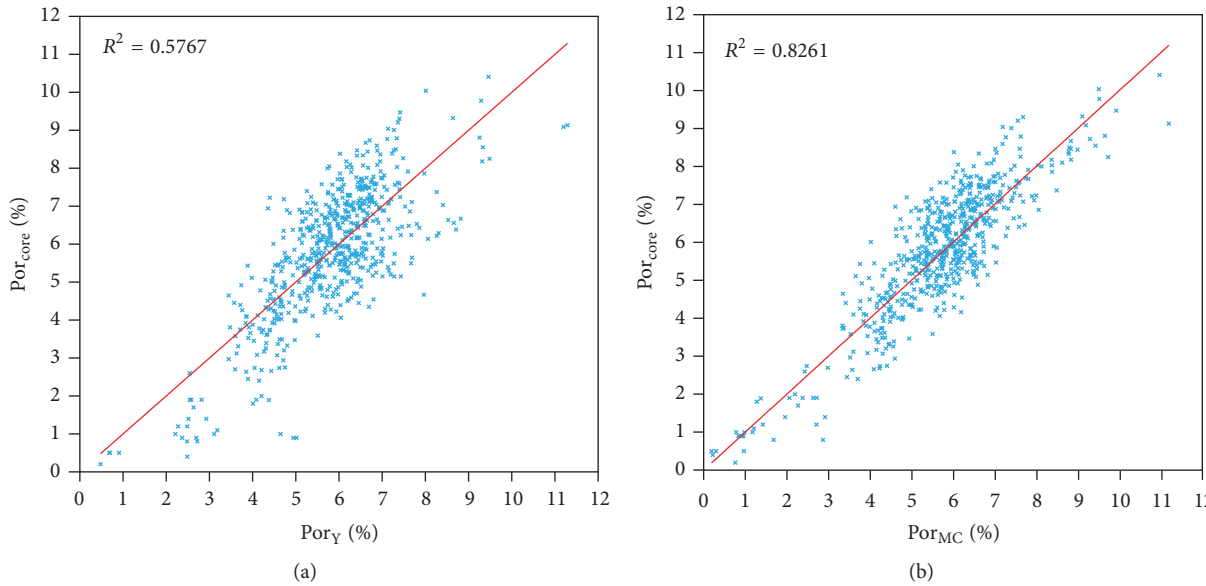


Figure 8: The contrast with Yong's initial value method. (a) The crossplot of core porosity and calculated porosity by Yong's initial value method. (b) The crossplot of core porosity and calculated porosity by Monte Carlo random initial value method.

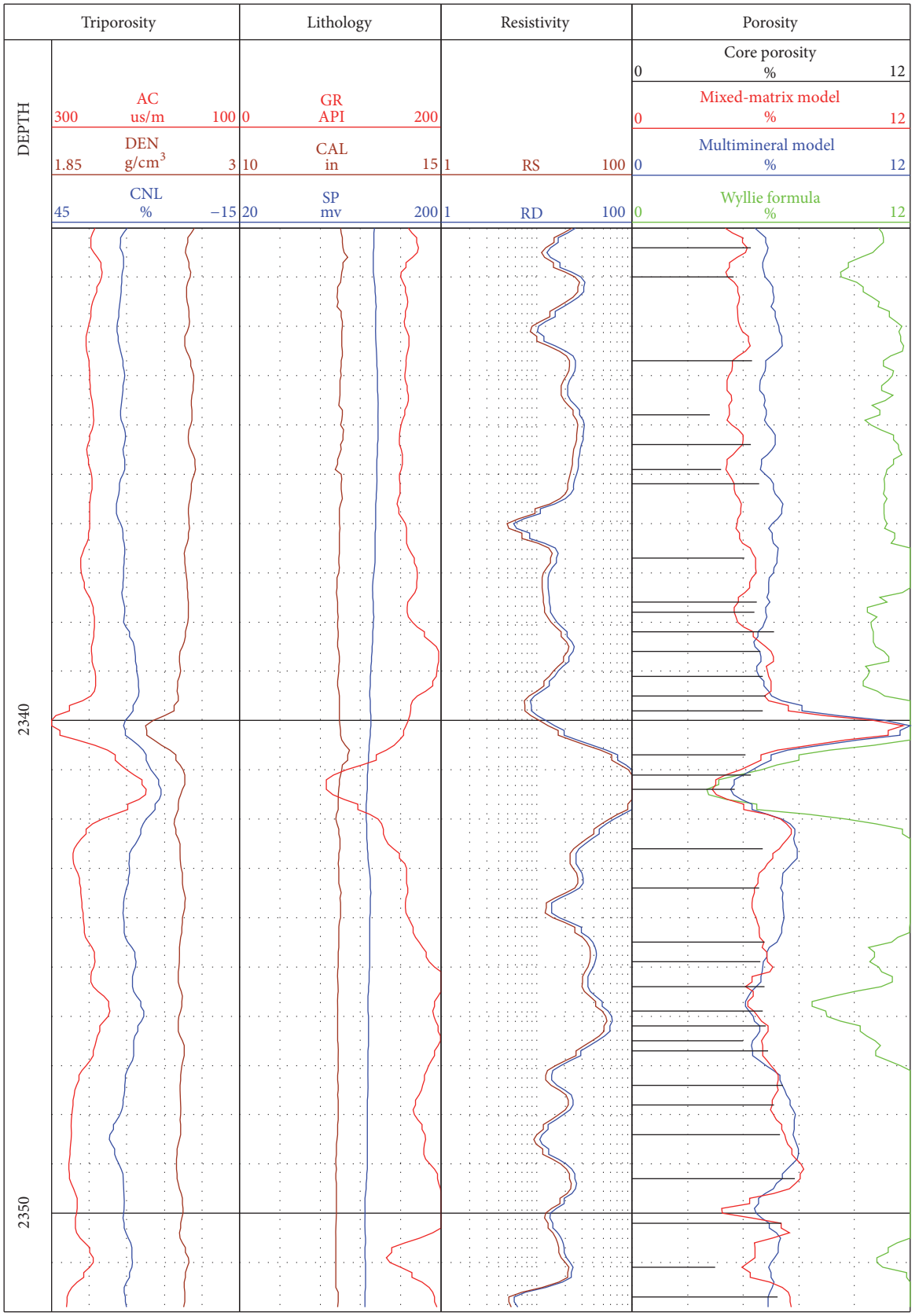


Figure 9: The contrast among Wyllie formula, mixed-matrix model, and the multimineral model.

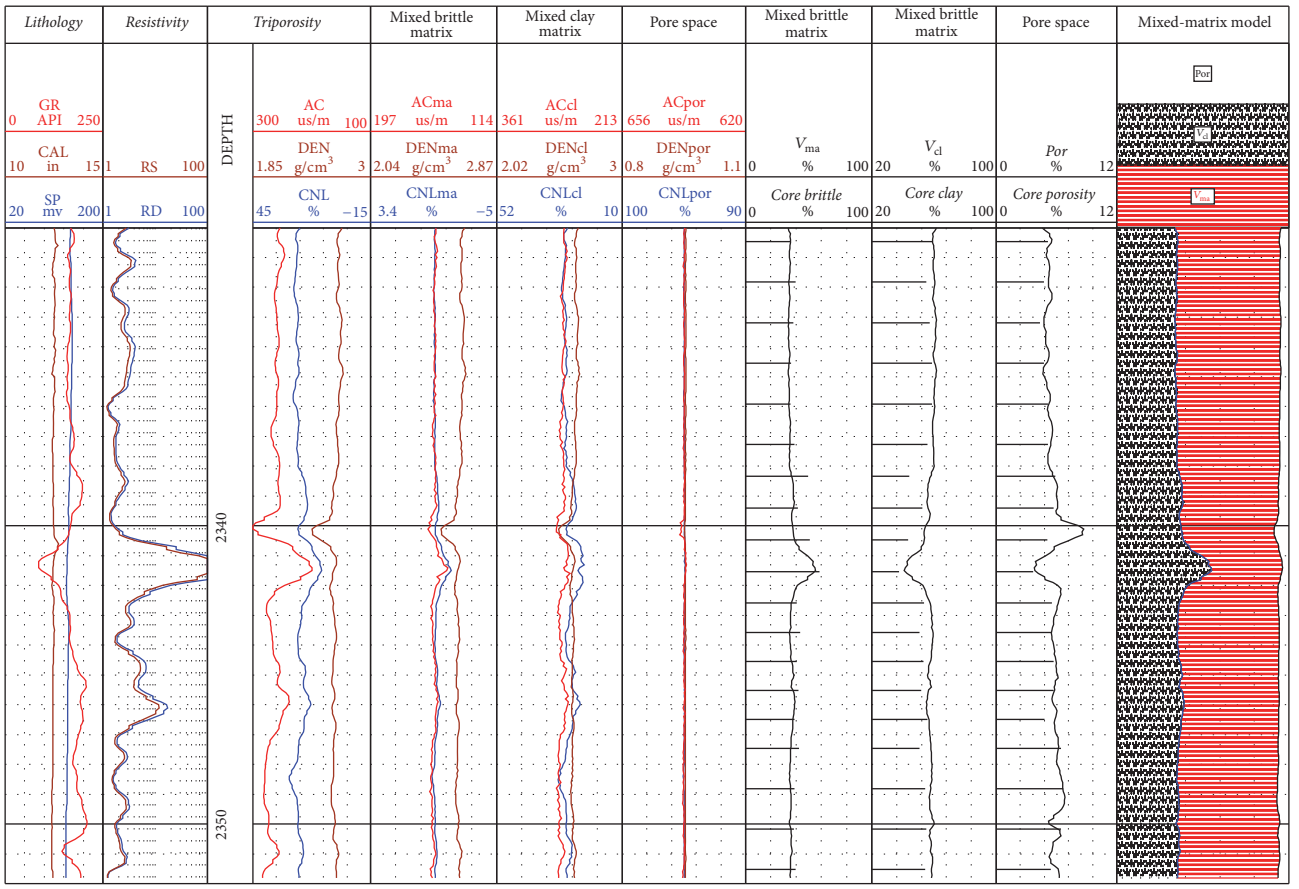


Figure 10: The physical parameters of the method.

It does not guarantee, of course, finding the global minimum of the incoherence function, but if the function has many good near-optimal solutions, it should find one. It is worth noting that the results of the calculation are slightly different each time, so we should use the normal-like distribution to eliminate those differences and achieve computational stability.

4.3. The Advantages and Limitations. All methods provide advantages as well as limitations which will be discussed next. The advantages of the method are the following: (a) the porosity can be evaluated well even if there is a lack of the regional geological parameters; (b) matrix physical parameters can be accurately obtained; (c) the result of calculation is different for each time, but it has consistent trend in the interval of computation; (d) the user is not required to compute all types of mineral, for instance, physical parameters, so the approach is easier.

On the other hand, the limitations are as follows: for individual well, iterating index of 30 may be not enough to meet the requirement. If the case happened, it will take more time to find the final answer.

Nomenclature

V_{ma}: Volume fraction of the mixed-brittle matrix
V_{cl}: Volume fraction of the mixed-clay matrix
ϕ: Porosity or the pore space volume
V_{QUA}: Volume fraction of quartz
V_{FEL}: Volume fraction of feldspar
V_{CAL}: Volume fraction of calcite
V_{DOL}: Volume fraction of dolomite
V_{KAO}: Volume fraction of kaolinite
V_{MON}: Volume fraction of montmorillonite
V_{CHL}: Volume fraction of chlorite
V_{OG}: Volume fraction of oil and gas
V_{W}: Volume fraction of water
Δt: Acoustic
ρ: Density
Φ_N: Neutron
Δt_{ma}: Acoustic of the mixed-brittle matrix
Δt_{cl}: Acoustic of the mixed-clay matrix
Δt_{ϕ}: Acoustic of fluid
ρ_{ma}: Density of matrix
ρ_{cl}: Density of clay
ρ_{ϕ}: Density of fluid

Φ_{Nma}: Neutron of matrix
Φ_{Ncl}: Neutron of clay
$\Phi_{N\phi}$: Neutron of fluid
v: A vector, $v = (V_{ma}, V_{cl}, \phi, \Delta t_{ma}, \Delta t_{cl}, \Delta t_f, \rho_{ma}, \rho_{cl}, \rho_f, \Phi_{Nma}, \Phi_{Ncl}, \Phi_{Nf})$
$F'(v)$: The incoherence function
$F(v)$: The normalized incoherence function
c_i: ith logging curve which is Δt, ρ and Φ_N
$f_i(v)$: ith tool response function
ΔF: The difference between $f(v')$ and $f(v_i)$
T: A parameter called temperature except 0
N: The termination cycle
N_T: Temperature reduction
f': The error tolerance for function
Por_{core}: The core porosity
Por_{ave}: The average porosity
Por_{sd}: The standard deviation porosity
Por_Y: The calculation porosity by Yong's method
Por_{MC}: The calculation porosity by Monte Carlo random initial value method
ma_{min}: The minimum of volume fraction of the mixed-brittle matrix
ma_{max}: The maximum of volume fraction of the mixed-brittle matrix
cl_{min}: The minimum of volume fraction of the mixed-clay matrix
cl_{max}: The maximum of volume fraction of the mixed-clay matrix
ϕ_{min}: The minimum of porosity
ϕ_{max}: The maximum of porosity.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was financially supported by (1) the National Natural Science Foundation of China, (Grant no. 41302107); (2) the Ministry of Land and Resources Special Funds for Scientific Research on Public Cause (Grant no. 201311107); (3) the Fundamental Research Funds for the Central Universities (Grant no. 2652011282); and (4) the CNPC Innovation Foundation (Grant no. 2012D-5006-0103).

References

[1] C. Mayer, "Global, a new approach to computer-processed log interpretation," in *Proceedings of the SPE Annual Technical Conference and Exhibition*, SPE-9341-MS, pp. 2–14, Society of Petroleum Engineers, Dallas, Tex, USA, September 1980.

[2] W. K. Mitchell and R. J. Nelson, "A practical approach to statistical log analysis," in *Proceedings of the SPWLA 27th Annual Logging Symposium*, vol. 19, June 1988.

[3] Y. N. Duan, B. Z. Pan, X. Han, H. T. Zhang, and X. M. Yang, "Porosity calculation of tight sand gas reservoirs with GA-CM hybrid optimization log interpretation method," *Journal of Geoscience and Environment Protection*, vol. 2, no. 3, pp. 92–98, 2014.

[4] G. Xie, Z. P. Hu, L. Luo, and S. D. Zhang, "The method for complex lithology logging identification based on the theory of constrained least squares," *Well Logging Technology*, vol. 31, no. 4, pp. 354–356, 2007 (Chinese).

[5] Z. L. Bao, *Optimization Theory and Algorithm*, Tsinghua University Press, 2005 (Chinese).

[6] A. Mezzatesta, E. Rodriguey, and E. Frost, *OPTIMA—A Statistical Approach to Well Log Analysis*, Dresser Atlas. Dresser Industries Inc, 1986.

[7] J. Ouyang, Y. W. Wang, and S. F. Hu, "The digital processing optimization method for Complex lithology," *Oil & Gas Geology*, vol. 4, no. 1, pp. 76–87, 1983 (Chinese).

[8] J. Ouyang, Y. W. Wang, and S. F. Hu, "The logging interpretation for Lamprophyre," *ACTA Petrolei Sinica*, vol. 5, no. 3, pp. 37–42, 1984 (Chinese).

[9] J. Ouyang, "The well log interpretation study on oily-igneous rock by optimization method," *Well Logging Technology*, vol. 9, no. 1, pp. 1–16, 1985 (Chinese).

[10] X. S. Yang, A. H. Gandomi, S. Talatahari, and A. H. Alavi, "Optimization and metaheuristic algorithms in engineering," *Metaheursitics in Water, Geotechnical and Transport Engineering*, vol. 11, pp. 1–23, 2013.

[11] P. L. Stoffa and M. K. Sen, "Nonlinear multiparameter optimization using genetic algorithms," *Geophysics*, vol. 56, no. 11, pp. 1794–1810, 1991.

[12] C. C. Zou, Z. L. Wei, X. Y. Cai, C. Han, and H. Y. Zhang, "The optimization log interpretation by the genetic algorithm," *Well Logging Technology*, vol. 23, no. 3, pp. 361–365, 1999 (Chinese).

[13] S. Kirkpatrick, J. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[14] A. Corana, M. Marchesi, C. Martini, and S. Ridella, "Minimizing multimodal functions of continuous variables with the "simulated annealing" algorithm," *ACM Transactions on Mathematical Software*, vol. 13, no. 3, pp. 262–280, 1987.

[15] S. H. Yong, J. M. Sun, and Z. X. Bi, "Using optimization method to evaluate complex lithologic reservoir," *Well Logging Technology*, vol. 12, no. 4, pp. 18–28, 1988 (Chinese).

[16] Y. N. Duan, B. Z. Pan, and X. Han, "Glutenite reservoir of Multi-Component Model and the Optimization Log Interpretation," *World Well Logging Technology*, vol. 192, pp. 20–23, 2012 (Chinese).

[17] P. Szucs and F. Civan, "Multi-layer well log interpretation using the simulated annealing method," *Journal of Petroleum Science and Engineering*, vol. 14, no. 3-4, pp. 209–220, 1996.

[18] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[19] M. Steinbrunn, G. Moerkotte, and A. Kemper, "Heuristic and randomized optimization for the join ordering problem," *The VLDB Journal*, vol. 6, no. 3, pp. 191–208, 1997.

[20] S. H. Yong and J. M. Sun, "The choice of optimization method in the digital procesing," *Journal of the University of Petroleum, China*, vol. 12, no. 4-5, pp. 11–25, 1988 (Chinese).

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 8306732, 13 pages
https://doi.org/10.1155/2017/8306732

Research Article

Application of Heuristic and Metaheuristic Algorithms in Solving Constrained Weber Problem with Feasible Region Bounded by Arcs

Igor Stojanović,[1] Ivona Brajević,[2] Predrag S. Stanimirović,[2] Lev A. Kazakovtsev,[3] and Zoran Zdravev[1]

[1]*Faculty of Computer Science, Goce Delčev University, Goce Delčev 89, 2000 Štip, Macedonia*
[2]*Department of Mathematics and Informatics, Faculty of Science and Mathematics, University of Niš, Višegradska 33, 18000 Niš, Serbia*
[3]*Department of Systems Analysis and Operations Research, Reshetnev University, Prosp. Krasnoyarskiy Rabochiy 31, Krasnoyarsk 660037, Russia*

Correspondence should be addressed to Predrag S. Stanimirović; pecko@pmf.ni.ac.rs

Received 26 February 2017; Accepted 15 May 2017; Published 14 June 2017

Academic Editor: Domenico Quagliarella

The continuous planar facility location problem with the connected region of feasible solutions bounded by arcs is a particular case of the constrained Weber problem. This problem is a continuous optimization problem which has a nonconvex feasible set of constraints. This paper suggests appropriate modifications of four metaheuristic algorithms which are defined with the aim of solving this type of nonconvex optimization problems. Also, a comparison of these algorithms to each other as well as to the heuristic algorithm is presented. The artificial bee colony algorithm, firefly algorithm, and their recently proposed improved versions for constrained optimization are appropriately modified and applied to the case study. The heuristic algorithm based on modified Weiszfeld procedure is also implemented for the purpose of comparison with the metaheuristic approaches. Obtained numerical results show that metaheuristic algorithms can be successfully applied to solve the instances of this problem of up to 500 constraints. Among these four algorithms, the improved version of artificial bee algorithm is the most efficient with respect to the quality of the solution, robustness, and the computational efficiency.

1. Introduction

The Weber problem is one of the most studied problems in location theory [1–3]. This optimization problem searches for an optimal facility location $X^* \in \mathbb{R}^2$ on a plane, which satisfies

$$X^* = \arg\min_{X\in\mathbb{R}^2} f(X) = \arg\min_{X\in\mathbb{R}^2} \sum_{i=1}^{N} w_i \left\| A_i - X \right\|. \quad (1)$$

In (1), it is assumed that $A_i \in \mathbb{R}^2$, $i \in \{1, \ldots, N\}$ are known demand points, $w_i \in \mathbb{R}$ and $w_i \geq 0$ are weight coefficients, and $\|\cdot\|$ is a matrix norm, used as the distance function.

The basic Weber problem is stated with the Euclidean norm underlying the definition of the distance function. Also, many other types of distances have been used in the facility location problems [3–5]. In general, a lot of extensions and modifications of the Weber location problem are known. Detailed reviews of these problems can be found in [3, 6].

The most popular method for solving the Weber problem with Euclidean distances is given by a one-point iterative procedure which was first proposed by Weiszfeld [7]. Later, Vardi and Zhang developed a different extension of Weiszfeld's algorithm [8], while Szegedy partially extended Weiszfeld's algorithm to a more general problem [9]. In particular, some variants of the continuous Weber problem represent nonconvex optimization problems which are hard to be solved exactly [10]. A nonconvex optimization problem may have multiple feasible regions and multiple locally optimal points

within each region [11]. Consequently, finding the global solution of a nonconvex optimization problem is very difficult.

Heuristics and metaheuristics represent the main types of stochastic methods [12]. Both types of algorithms can be used to speed up the process of finding a high-quality solution in the cases where finding an optimal solution is very hard. The distinctions between heuristic and metaheuristic methods are inappreciable [12]. Heuristics are algorithms developed to solve a specific problem without the possibility of generalization or application to other similar problems [13]. On the other hand, a metaheuristic method represents a higher-level heuristic in the sense that they guide their design. In such a way we can use any of these methods to design a specific method for computing an approximate solution for an optimization problem.

In the last several decades, there is a trend in the scientific community to solve complex optimization problems by using metaheuristic optimization algorithms. Some applications of metaheuristic algorithms include neural networks, data mining, industrial, mechanical, electrical, and software engineering, as well as certain problems from location theory [14–21]. The most interesting and most widely used metaheuristic algorithms are swarm-intelligence algorithms which are based on a collective intelligence of colonies of ants, termites, bees, flock of birds, and so forth [22]. The reason of their success lies in the fact that they use commonly shared information among multiple agents, so that self-organization, coevolution, and learning during cycles may help in creating the highest quality results. Although not all of the swarm-intelligence algorithms are successful, a few techniques have proved to be very efficient and thus have become prominent tools for solving real-world problems [23]. Some of the most efficient and the most widely studied examples are ant colony optimization (ACO) [24–26], particle swarm optimization (PSO) [15, 27–29], artificial bee colony (ABC) [19, 30–35], and recently proposed firefly algorithm (FA) [18, 36–38] and cuckoo search (CS) [17, 39–41].

Different heuristic methods are proposed in order to provide encouraging results for challenging continuous Weber problem with regard to solution quality and computational effort [42–46]. Also, some variants of the Weber problem have been successfully solved by different metaheuristic approaches [47–52]. In [52], the authors studied a capacitated multisource Weber problem as an extended facility location problem that involves both facility locations and service allocations simultaneously. The method proposed in [52] is based on the integration of two genetic algorithms. The problem of locating one new facility with respect to a given set of existing facilities in the plane and in the presence of convex polyhedral barriers was considered in [47]. The general strategy in [47] arises from the iterative application of a genetic algorithm for the subproblems selection. A hybrid particle swarm optimization approach was applied in solving the incapacitated continuous location-allocation problem in [48]. In [49], the authors compared performances of four metaheuristic algorithms, modified to solve the single-facility location problem with barriers. The method for solving a kind of Weber problem from [50] was developed using an evolutionary algorithm enhanced with variable neighborhood search.

The aim of this paper is to investigate the performances of some prominent swarm-intelligence metaheuristic approaches to solve the constrained Weber problem with feasible region bounded by arcs. This variant of Weber problem has a nonconvex feasible set given by the constraints that make it much harder to find the global optimum using any deterministic algorithms. Hence, metaheuristic optimization algorithms can be employed in order to provide promising results.

In this paper, four swarm-intelligence techniques are applied to solve this version of the constrained Weber problem: the artificial bee colony for constrained optimization [53], the crossover-based artificial bee colony (CB-ABC) algorithm [54], the firefly algorithm for constrained optimization [37], and the enhanced firefly algorithm (E-FA) [55]. The CB-ABC and the E-FA are two of the most recently proposed improved variants of the ABC and FA for solving constrained problems, respectively. Also, a heuristic algorithm is proposed in [44] with the aim of solving this version of the constrained Weber problem. Hence, it is also implemented for the purpose of comparison with the metaheuristic approaches. These five techniques are tested to solve randomly generated test instances of constrained Weber problem with feasible region bounded by arcs of up to 500 constraints.

The rest of the paper is organized as follows. A formulation of the constrained Weber problem with feasible region bounded by arcs and the heuristic approach developed to solve this variant of the constrained Weber problem are presented in Section 2. Section 3 presents the four metaheuristic optimization techniques used to solve this variant of the Weber problem. Description of the generated benchmark functions and comparative results of the four implemented metaheuristic techniques are given in Section 4. Concluding remarks are provided in Section 5.

2. The Heuristic Method for Solving a Constrained Weber Problem

The constrained Weber problem with feasible region bounded by arcs in the continuous space was introduced in [44]. In order to complete our presentation, we briefly restate the method. It can be formulated by the goal function defined in (1) and by the feasible region which is defined on the basis of constraints of two opposite types:

$$\begin{aligned} \mathcal{S}_{<} &= \left\{ i \in \{1, \ldots, N\} \mid \left\| X - A_i \right\| \leq 1 \right\}, \\ \mathcal{S}_{>} &= \left\{ i \in \{1, \ldots, N\} \mid \left\| X - A_i \right\| \geq 1 \right\}, \end{aligned} \tag{2}$$

where N is the total number of demand points and $\{1, \ldots, N\}\mathcal{S}_{<}$ and $\mathcal{S}_{>}$ are subsets of the set of demand point indices satisfying $\{1, \ldots, N\}\mathcal{S}_{<}$, $\mathcal{S}_{>} \subseteq \{1, \ldots, N\}$, and $\mathcal{S}_{<} \cap \mathcal{S}_{>} = \emptyset$. For the sake of simplicity, the optimization problem given by (1) with constraints (2) is denoted as the CWP problem.

Such a problem may occur if some demand points coincide with locations of some important facilities and the

searched optimal location X^* must be close to them. Other demand points may coincide with dangerous facilities and the facility X^* must be located far from them.

The metric used in practically important location problems depends on various factors, including properties of the transportation means [44]. In the case of public transportation systems, the price usually depends on a distance. However, some minimum price is usually defined. For example, the initial fare of the taxi cab may include some distance, usually 1–5 km. Having rescaled the distances so that this distance included in the initial price is equal to 1, we can define the price function d_P as

$$d_P(X,Y) = \max\{\|X - Y\|, 1\} \quad \forall X, Y \in \mathbb{R}^2, \tag{3}$$

where $\|\cdot\|$ is a matrix norm.

In the case of distance function defined by (3), the problem can be decomposed into series of constrained location problems with the Euclidean metric where the area of the feasible solutions is bounded by arcs. Each of the problems has the feasible region equal to the same intersection of the discs with centers in the demand points. For more details, see [44, 56].

The Weiszfeld procedure for solving the Weber problem with a given tolerance ε, based on the results from [57], is presented as Algorithm 2.1 in [44].

An algorithm based on the Weiszfeld procedure for solving the CWP defined by objective (1) and constraints (2) was proposed [44]. The feasible set of our constrained optimization problems is generally nonconvex, while the objective function $f(X)$ given by (1) is convex [58]. A solution of constrained optimization problems with convex objective functions coincides with the solution of the unconstrained problem or lies on the border of the forbidden region [59]. Thus, if X^* is a solution of the constrained problem given by (1) with constraints (2) then it is the solution of the unconstrained problem (1) or $\exists i \in \{\overline{1,N} : \|A_i - X^*\|_2 = 1\}$.

Step 2.2 of Algorithm 2.1 from [44] can lead to generating a new point X^{**} outside the feasible region determined by constraints (2). Let us denote this region $\mathscr{R}_f$. It is assumed that $\mathscr{R}_f \neq \emptyset$.

For an arbitrary point $X \in \mathbb{R}^2$, let us denote the closest point in $\mathscr{R}_f$ by $\mathscr{C}(X)$. It can be computed using

$$\mathscr{C}(X) = \arg\min_{X' \in \mathscr{R}_f} \left\|X - X'\right\| = \begin{cases} X, & X \in \mathscr{R}_f, \\ \arg\min_{X' \in \mathscr{R}_f} \left\|X - X'\right\|, & X \notin \mathscr{R}_f. \end{cases} \tag{4}$$

Algorithm 1 was proposed as Algorithm 2.2 in [44], and it is based on the substitution of the point X^{**} generated in Step 2.2 of Algorithm 2.1 from [44] with its closest point $\mathscr{C}(X^{**})$ in the feasible region.

3. Review of the Metaheuristic Optimization Techniques

The four metaheuristics used to solve constrained Weber problem with feasible region bounded by arcs are described in the following subsections.

3.1. Artificial Bee Colony Algorithm for Solving the CWP. A numerical variant of the ABC algorithm for constrained optimization problems (COPs) proposed in [60] is applied to solve the CWP. In the ABC the population is iteratively refined through employed, onlooker, and scout bee phases.

The update process used in the employed and onlooker bee phase is the same and it is determined by

$$v_{ij} = \begin{cases} x_{ij} + \varphi_j \cdot \left(x_{ij} - x_{kj}\right), & \text{if } R_j < \text{MR} \\ x_{ij}, & \text{otherwise,} \end{cases} \tag{5}$$

where φ_j is a uniform random number in the range $[-1, 1]$, x_k represents another solution selected randomly from the population, MR is the modification rate control parameter, R_j is a randomly chosen real number in the range $[0, 1)$, and $j = 1, 2$. The update process is completed when the selection between x_i and v_i is carried out.

The ABC uses Deb's rules in order to decide which solution will be kept for the next iteration. This constraint handling method consists of a set of three feasibility rules introduced by Deb [61]. They are the following: (1) any feasible solution is preferred to any infeasible solution, (2) between two feasible solutions, the one having a better fitness value is preferred, and (3) if both solutions are infeasible, the one with the lowest sum of constraint violations is preferred.

In the employed bee phase, every solution involves the update process. On the other hand, in the onlooker bee phase only the solutions selected probabilistically proportional to their fitness values have the chance to be upgraded [60].

In the scout phase solutions that do not improve over a certain number of cases are replaced by new randomly generated solutions. The control parameters *limit* and the scout production period SPP are used in this phase. The parameter *limit* is used to signify exhausted food source, while SPP parameter is employed in order to denote a predetermined period of cycles for producing scout bees.

The pseudocode of the ABC is given as Algorithm 2.

3.2. Crossover-Based Artificial Bee Colony Algorithm for Solving the CWP. Recent improved variant of the ABC for COPs, called crossover-based artificial bee colony, is also used to solve the constrained Weber problem [54]. The main modifications introduced in the CB-ABC are related to the search operators used in each bee phase in order to improve the distribution of good information between solutions [54]. The differences between the CB-ABC and the ABC for COPs are given as follows.

In the employed bee phase, the CB-ABC algorithm uses modified search equation (5), in which φ is the same random number of each parameter j which will be changed. Also, the CB-ABC does not use the fixed value of MR control parameter. Value of MR linearly increases from 0.1 to the predefined

Require: Coordinates and weights of the demand points $A_i = (a_1^i, a_2^i), w_i,\ i = \overline{1, N}$, pre-specified tolerance ε, constraints (2) specified by sets $\mathcal{S}_<$ and $\mathcal{S}_>$.

Step 1. Calculate the initial point $X^* \in \mathcal{R}_f$ (here, $\mathcal{R}_f$ is the feasible set bounded by constraints);

$$X^* = \mathcal{C}(X^*);\ \Delta = +\infty.$$

Step 2. While $\Delta > \varepsilon$ do:

Step 2.1. $n_{\text{iter}} = n_{\text{iter}} + 1$;

$$d_{\text{denom}} = \sum_{i=1}^{N} \frac{w_i}{\|A_i - X^*\|_2}.$$

Step 2.2. $x_r^{**} = \sum_{i=1}^{N} (x_r^* w_i / (\|X^* - A_i\|_2 \cdot d_{\text{denom}}))\ \forall r \in \{1, 2\}$.

Step 2.3. If $X^{**} \notin \mathcal{R}_f$ then $X^{**} = \mathcal{C}(X^{**})$.

Step 2.4. $\Delta = \|X^* - X^{**}\|$; $X^* = X^{**}$.

Step 2.5. Continue Step 2.

Step 3. STOP, return X^{**}.

Algorithm 1: Solving the CWP problem.

Initial parameters of the ABC including maximum cycle number (MCN), SN, MR, *limit*, SPP;
Generate initial population x_i ($i = 1, 2, \ldots, \text{SN}$) randomly in the search space and evaluate each x_i;
$t = 0$;
while ($t <$ MCN) **do**
  for $i = 1$ to SN **do**
    Generate a solution v_i with x_i by Eq. (5), evaluate it and apply selection process based on Deb's method between v_i with x_i;
  end for
  for $i = 1$ to SN **do**
    Select food source x_l based on fitness proportionate selection;
    Generate a solution v_l with x_l by Eq. (5), evaluate it and perform selection process based on Deb's method between v_l with x_l;
  end for
  if (t mod SPP = 0) **then**
    Every solution which did not enhance at least *limit* number of times is replaced, each with a randomly produced solution.
  end if
  Memorize the best solution reached so far.
  $t = t + 1$
end while

Algorithm 2: Pseudocode of the ABC.

Table 1: The values of specific control parameters of the algorithms.

FA		E-FA		ABC		CB-ABC	
α	0.25	α	0.25	MR	0.8	$\text{MR}_{\max}$	0.9
β	1	β	1.5	*limit*	SN	*limit*	1
γ	1	γ	1	SPP	SN	SPP	50
						P	0.3

value $\text{MR}_{\max}$ in the first $P * \text{MCN}$ iterations, while the value $\text{MR} = \text{MR}_{\max}$ is used in the remaining iterations. The value of P is defined in Table 1.

In the onlooker bee phase, the CB-ABC proposes a new search equation with the aim of enabling better exploration of the neighborhood of the high-quality solution. This equation is given by

$$v_{ij} = x_{ij} + \varphi \cdot \left(x_{lj} - x_{kj}\right), \tag{6}$$

where φ is a uniform random number in range $[-1, 1]$, x_l and x_k represent the other two solutions selected randomly from the population, R_j is a randomly chosen real number in the range $[0, 1)$, and $j = 1, 2$.

In the scout bee phase, the CB-ABC uses uniform crossover operator to generate new solutions in a promising region of the search space. Therefore, after each SPPth iteration, each solution x_i which did not improve *limit* number of times is replaced with a new solution which is created by

$$v_{ij} = \begin{cases} y_j, & \text{if } R_j < 0.5 \\ x_{ij}, & \text{otherwise,} \end{cases} \tag{7}$$

where y_j is the jth element of the global best solution found so far, R_j is a randomly chosen real number in range $[0, 1)$, and $j = 1, 2$.

Initial parameters of the FA including SN, MCN, $\alpha_0, \beta_0, \gamma$.
Generate initial population of fireflies x_i ($i = 1, 2, \ldots, \text{SN}$) randomly distributed in the solution space.
Assume that $\varphi(x_i)$ is the expanded objective function of x_i ($i = 1, 2, \ldots, \text{SN}$) calculated by (10).
$t = 0$.
while $t <$ MCN **do**
 for $i = 1$ to SN **do**
  for $j = 1$ to SN **do**
   if $(\varphi(x_j) < \varphi(x_i))$ **then**
    Generate a new x_i according to Eq. (8) and evaluate it.
   end if
  end for
 end for
 $t = t + 1$.
 $\alpha(t) = \alpha(t-1) \cdot (10.0^{-4.0}/0.9^{1/\text{MCN}})$.
 Rank the fireflies and memorize the best solution achieved so far.
end while

Algorithm 3: Pseudocode of the FA.

3.3. Firefly Algorithm for Solving the CWP. In order to solve the CWP we have employed a numerical optimization version of the FA for COPs, introduced in [37]. In the FA, a colony of artificial fireflies searches for good solutions in every iteration.

The search operator represents the movement of a firefly i to another more attractive or brighter firefly j and it is given by

$$x_{ik} = x_{ik} + \beta \cdot \left(x_{jk} - x_{ik}\right) + \alpha \cdot S_k \cdot \left(\text{rand}_k - \frac{1}{2}\right), \quad (8)$$

where the second term is due to the attraction and the third term is a randomization term.

In the second term of (8), the parameter β is the attractiveness of fireflies which is calculated according to the following monotonically decreasing function [62]:

$$\beta = \beta_0 \cdot e^{-\gamma \cdot r_{ij}^2}, \quad (9)$$

where r_{ij} denotes the distance between firefly x_i and firefly x_j, while β_0 and γ are predetermined algorithm parameters: maximum attractiveness value and absorption coefficient, respectively. Distance between fireflies is calculated by the Euclidean distance.

In the third term of (8), $\alpha \in [0, 1]$ is a randomization parameter, S_k are the scaling parameters, and rand_k is a random number uniformly distributed between 0 and 1. The scaling parameters S_k ($k = 1, 2$) are calculated by $S_k = |u_k - l_k|$, where l_k and u_k are the lower and upper bound of the parameter x_{ik}. Diversity of solutions is controlled by the randomization parameter α which needs to be reduced gradually during iterations so that it can vary with the iteration counter t [63].

In the FA for solving CWP, penalty functions approach is used in order to handle the constraints. In this way, a constrained problem is solved as an unconstrained one. A general formula of calculation penalty functions is given in [64] by

$$\varphi(X) = f(X) + \sum_{j=1}^{q} r_j \cdot \max\left(0, g_j(X)\right)^2 + \sum_{j=q+1}^{m} c_j \cdot \left|h_j(X)\right|, \quad (10)$$

where $\varphi(X)$ is the new (expanded) objective function to be optimized, r_j and c_j are positive constants normally called "penalty factors," q is the number of inequality constraints, and $m - q$ is the number of equality constraints for a given problem. We found it suitable to set each r_i to the value $r_i = 10^8$. The penalty factors for equality constraints were not used, since these problems have only inequality constraints.

The pseudocode of the FA is given as Algorithm 3.

3.4. An Enhanced Firefly Algorithm for Solving the CWP. An enhanced firefly algorithm for COPs is presented in [55] and it is also applied to solve the CWP. Two modifications are incorporated in the E-FA in order to improve the performance of the firefly algorithm for COPs.

The first modification is related to using Deb's rules instead of the penalty approach. Three feasibility rules are employed instead of the greedy selection in order to decide which firefly is brighter. These rules are also used each time after (8) is applied in order to decide whether the solution will be updated. Evaluation of solution population is given as Algorithm 4.

The second modification is employing the geometric progression reduction scheme to reduce the scaling factors S_k at the end of each cycle, by the rule

$$S_k(t) = S_k(t-1) \cdot \theta^{1/\text{MCN}}, \quad (11)$$

where MCN is the maximum cycle number, t is the current iteration number, and $\theta = 10.0^{-4.0}/0.9$.

```
for i = 1 to SN do
  for j = 1 to SN do
    if (x_j is better than x_i based on Deb's rules) then
      for k = 1 to D (dimension of the problem) do
        g_k = x_ik + β · (x_jk − x_ik) + α · S_k · (rand_k − 1/2)
      end for
      if (g is better than x_i based on Deb's rules) then
        x_i = g {the solution is updated}
      end if
    end if
  end for
end for
```

Algorithm 4: Evaluation of a new population in the E-FA.

4. Experimental Study

The ABC, CB-ABC, FA, and E-FA are implemented in the Java programming language on a PC Intel Core i5-3300@3 GHz with 4 GB of RAM. The heuristic algorithm based on the modified Weiszfeld procedure is also implemented for the purpose of comparison with the metaheuristic approaches.

4.1. Benchmark Functions. The performance of the four metaheuristics techniques and behavior of the heuristic algorithm are evaluated through eighteen test instances of the single-facility constrained Weber problems with the connected feasible region bounded by arcs with equal radius.

The benchmark problems with the increasing number of input points are randomly generated according to the algorithm given in [44]. These problems have 5, 10, 50, 100, 250, and 500 input points. Three different random test problems are generated for each number of input points. Hence, these test instances have a nonconvex feasible set given from 5 up to 500 constraints.

Four example problems, named P1, P4, P7, and P10 with 5, 10, 50, and 100 input points, respectively, are shown in Figure 1. In each test image, the feasible region is represented by a gray surface area and the final solution obtained by the heuristic algorithm [44] is represented by a red cross.

4.2. Parameter Settings. The solution number (SN) in the four metaheuristic algorithms was set to 20. The maximum number of fitness function evaluations (FEs) was used as the stopping criterion. The allowed FEs were set to 8000. In addition, the metaheuristic algorithms presented in Section 2 have several other control parameters that considerably influence their performance. The values of these control parameters are presented in Table 1.

In order to calculate FEs researchers usually use the rule SN∗MCN, where MCN is the maximum number of iterations [65, 66]. Hence, the FA and E-FA were terminated after 400 iterations. The number of consumed fitness evaluations in each iteration of the ABC and CB-ABC algorithms is 2 ∗ SN, since it calculates the solutions both in the employed bee and in onlooker bee phase [65]. Therefore, to ensure a fair comparison, the ABC and CB-ABC algorithms were terminated after 200 iterations.

For the FA, it is widely reported in the literature that the light absorption coefficient $\gamma = O(1)$, the initial attractiveness $\beta_0 = 1$, and the initial randomness factor $\alpha_0 \in [0, 1]$ can be used for most applications [36, 62]. It can be seen from Table 1 that the value of the parameter γ was set to 1 and the initial value of α was set to 0.25 for both FA and E-FA. A typical value of $\beta_0 = 1$ is used in the FA. It was empirically determined that slightly higher value of the parameter β_0 is more suitable for the E-FA. Hence $\beta_0 = 1.5$ was adapted. For the ABC and CB-ABC algorithms, the values of the specific control parameters were taken from [53, 54], where these algorithms were proposed to solve COPs. Especially for the CB-ABC, it was empirically determined that a lower value of the scout production period SPP is more appropriate for solving the CWP. Therefore, it was set to 50. Each of the experiments was repeated for 30 runs.

4.3. Analysis of Solution Quality and Robustness. The coordinates of the solution, corresponding objective function value, and the CPU time (in seconds) obtained by the heuristic algorithm are arranged in Table 2. To analyze the solution quality of the tested four metaheuristic algorithms, the best values, mean values, and standard deviations have been obtained by the ABC, CB-ABC, FA, and E-FA algorithms over 30 runs. Significance tests are used to achieve reliable comparisons. According to [67], two-sample 95%-confidence t-test was conducted between each pair of compared metaheuristics on every benchmark function. The calculated best results are presented in Table 3, while the mean values and standard deviations are arranged in Table 4. Results of two-sample t-tests are reported in Table 5. The sign "+" indicates that the associated comparative algorithm is significantly better than the other one, while the sign "−" indicates it is significantly worse than the opposite one. If both algorithms show similar performance, they are both marked by "+."

Kazakovtsev in [44] experimentally proved the convergence of the heuristic algorithm on randomly generated test problems. Hence, the calculated best values of the metaheuristics can be compared to the results found by the heuristic approach in order to show the ability of the metaheuristic algorithm to reach the near-optimal result. The obtained mean and standard deviation values indicate the robustness of the metaheuristic approaches.

It can be seen from Table 3 that each of the metaheuristic algorithms found the best results which are very close to the results obtained by the heuristic algorithm. More precisely, the FA obtained 10 better best results (P2, P4, P5, P6, P8, P9, P11, P16, P17, and P18) and 8 worse best results with respect to the heuristic approach. The E-FA obtained 11 better best results (P1, P2, P4, P5, P6, P8, P9, P11, P16, P17, and P18), one equal best result (P3) and 6 slightly worse best results in comparison with the heuristic approach. The ABC algorithm achieved 12 better best results (P2, P4, P5, P6, P8, P9, P10, P11, P12, P16, P17, and P18), one equal best result (P3), and 5 worse best results with respect to the heuristic approach. The algorithm CB-ABC was able to find better or the same best solution for all problems with respect to the heuristic

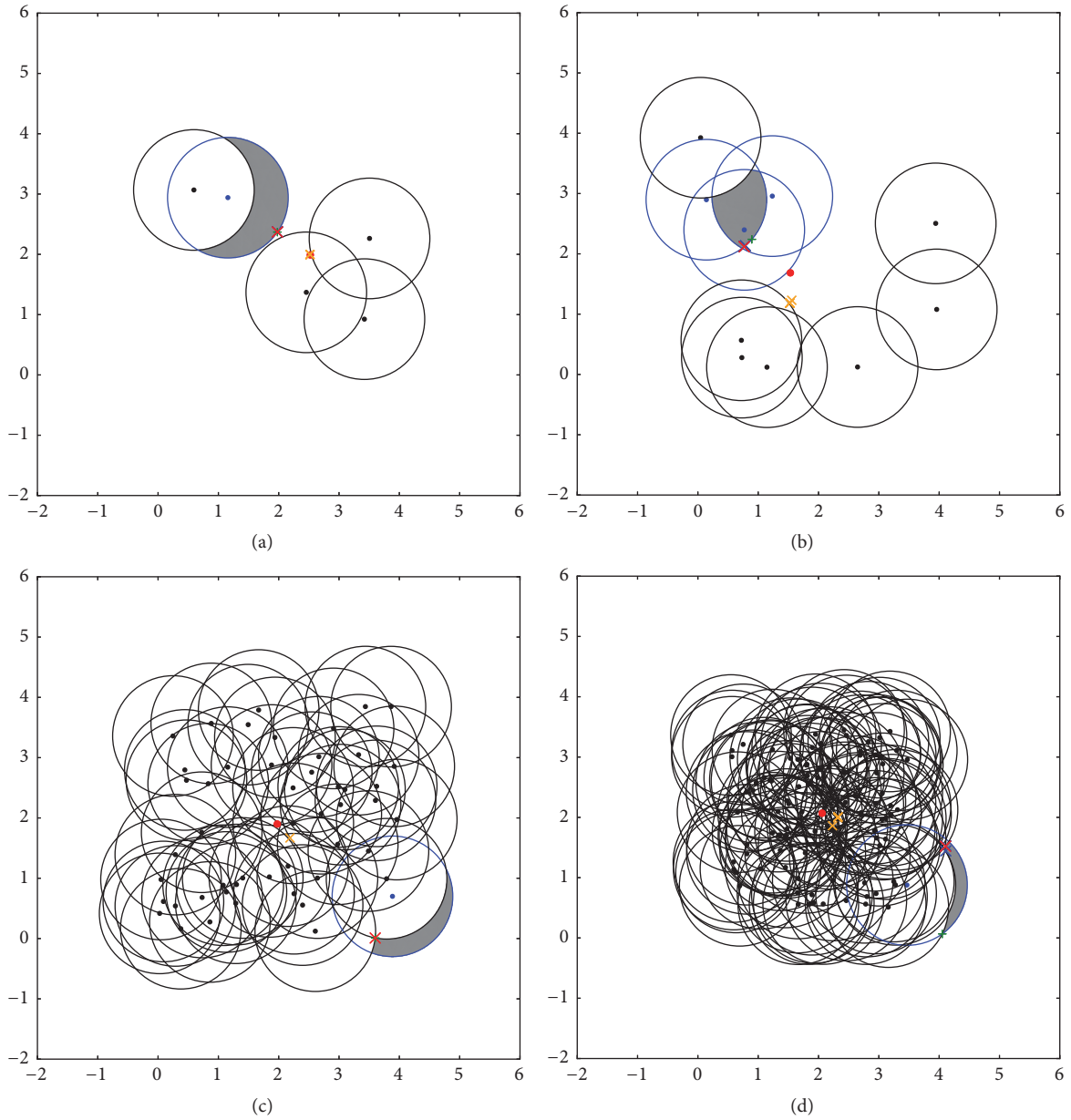


Figure 1: Example problems with (a) 5 points (P1), (b) 10 points (P4), (c) 50 points (P7), and (d) 100 points (P10).

algorithm, with the exception of the problem P7, where the CB-ABC obtained slightly worse best result.

In terms of best results from Table 3, it can be noticed that the CB-ABC achieved better and in several cases the same values in comparison with each considered metaheuristic approach. Further, each of the improved metaheuristics, the E-FA and CB-ABC, obtained better best results with respect to both original metaheuristic algorithms for the majority of test problems. If we compare the performance of the original ABC to that of the original FA, it can be seen that both algorithms show similar ability to reach the near-optimal result; that is, the ABC has found 9 slightly better best results and 9 slightly worse ones compared to the FA.

From Table 4, it can be seen that mean and standard deviation results obtained by the CB-ABC are much better than the results obtained by the other metaheuristic algorithms. The CB-ABC converged consistently to the same solution with the same objective function value and very lower standard deviation. If we compare the robustness of the remaining three metaheuristics, it can be noticed that the E-FA outperformed the FA and ABC. Compared with the ABC, the FA obtained 9 better mean results and standard deviation values (P1, P2, P4, P6, P11, P13, P14, P17, and P18). The remaining mean and standard deviation results are better in the case of the ABC algorithm, with the exception of P5 and P15 where the FA and ABC show similar performances.

Table 2: The solution point, corresponding objective function, and time in seconds provided by the heuristic algorithm.

Prob	D	Solution point	Objective function value	Time (sec)
P1	5	{1.97900484015, 2.37038768628}	38.6308975914	$4.61E-05$
P2	5	{2.511104376, 2.36822367838}	35.6524439425	$3.34E-05$
P3	5	{3.29440267812, 1.91712629136}	32.8210021279	$3.45E-05$
P4	10	{0.768148485664, 2.11871223533}	18.9069002076	0.0001
P5	10	{1.35331974755, 1.96480428579}	18.4514161651	0.0001
P6	10	{2.40607643306, 1.99679256415}	106.1627295736	0.0001
P7	50	{3.60036240333, 0.00927504530255}	723.0047301353	0.0002
P8	50	{2.14432492473, 0.0678576078541}	114.6249767471	0.0007
P9	50	{1.61875017517, 0.463308206334}	455.1344852622	0.0002
P10	100	{4.10439880225, 1.51968295711}	231.0855570171	0.0003
P11	100	{2.85555816286, 4.68224060226}	297.6045429275	0.0009
P12	100	{2.81772843636, 4.5680021125}	286.7386225628	0.0005
P13	250	{0.861049499813, 0.377164213194}	549.9465893405	0.0008
P14	250	{0.794900743781, 0.459653174322}	543.7610648263	0.0007
P15	250	{0.844305238343, 0.460727996564}	536.8203916176	0.0015
P16	500	{0.816903951623, 0.164582180338}	6194.3817254403	0.0032
P17	500	{0.478124601239, 0.957209048476}	1019.3526027819	0.0041
P18	500	{0.95138220166, 0.698523640367}	4667.4820432420	0.0036

Table 3: Comparison of the best solutions obtained from the FA, E-FA, ABC, and CB-ABC algorithms for 18 test instances over 30 runs.

Prob	D	FA	E-FA	ABC	CB-ABC
P1	5	38.6308975997	**38.6308975913**	38.6309142364	**38.6308975913**
P2	5	32.6409719983	**32.6409719926**	32.6409728972	**32.6409719926**
P3	5	32.8210027975	**32.8210021279**	**32.8210021279**	**32.8210021279**
P4	10	18.7528484406	**18.7528484372**	18.7528484934	**18.7528484372**
P5	10	18.3972444320	**18.3972444317**	18.3972444324	**18.3972444317**
P6	10	105.9917830548	**105.9917830153**	105.9917835005	**105.9917830153**
P7	50	723.0047799356	723.0047449044	**723.0047448967**	**723.0047448967**
P8	50	108.9894160829	108.9894138822	**108.9894138815**	**108.9894138815**
P9	50	455.1344837962	455.1344639665	**455.1344639631**	**455.1344639631**
P10	100	231.0855604428	231.0855570172	231.0855570166	**231.0855570165**
P11	100	297.2586347001	297.2586211173	297.2586217396	**297.2586211143**
P12	100	286.7386294462	286.7386225632	**286.7386225626**	**286.7386225626**
P13	250	549.9466005355	549.9465893424	549.9476076576	**549.9465893411**
P14	250	543.7610916668	543.7610648272	543.7611199289	**543.7610648263**
P15	250	536.8204201801	536.8203916186	536.8203922596	**536.8203916167**
P16	500	6193.7933019446	6193.7927947779	**6193.7927947450**	**6193.7927947450**
P17	500	1017.8088547714	1017.8088230476	1017.8091304405	**1017.8088230326**
P18	500	4667.0497043757	4667.0492697127	4667.0515567517	**4667.0492696773**

Results of two-sample t-tests are given in Table 5, and they show that the CB-ABC is significantly better than the FA, E-FA, and ABC on 18, 14, and 12 test problems, respectively. It is similar to the FA, E-FA, and ABC on 0, 4, and 6 problems, respectively. It is worth noting that the FA, E-FA, and ABC can not outperform the CB-ABC on any problem. Further, it can be observed that the E-FA is significantly better than the FA on each test problem. In comparison with the ABC, the E-FA is superior on 11 test problems, inferior on 4 problems, and similar on 3 benchmarks. When comparing the performances of the FA and ABC it can be noticed that the FA is significantly better than the ABC on 7 problems, while it is inferior to it on 7 problems. The FA and ABC show similar performances on 4 benchmarks.

According to the results reported in Tables 3, 4, and 5, we can conclude that the CB-ABC and E-FA exhibit superior performances compared to both original versions, ABC and FA, in solving constrained Weber problems with

Table 4: Comparison of the mean values and standard deviations obtained from the FA, E-FA, ABC, and CB-ABC algorithms for 18 test instances over 30 runs.

Prob	D	Stats	FA	E-FA	ABC	CB-ABC
P1	5	Mean	38.6308978682	38.6308975914	38.6310183806	**38.6308975913**
		Std	$2.40E-7$	$4.15E-11$	$1.56E-4$	**2.08E − 14**
P2	5	Mean	32.6409720879	**32.6409719926**	32.6409910473	**32.6409719926**
		Std	$8.97E-8$	$1.57E-11$	$2.80E-5$	**1.21E − 14**
P3	5	Mean	32.8210056512	32.8210021284	**32.8210021279**	**32.8210021279**
		Std	$1.90E-6$	$3.29E-10$	$4.49E-11$	**1.20E − 12**
P4	10	Mean	18.7528484769	**18.7528484372**	18.7528528692	**18.7528484372**
		Std	$1.14E-8$	$6.66E-12$	$5.62E-6$	**1.02E − 14**
P5	10	Mean	18.3972444465	**18.3972444317**	18.3972444546	**18.3972444317**
		Std	$3.40E-8$	$3.77E-12$	$1.91E-8$	**3.04E − 15**
P6	10	Mean	105.9917835910	105.9917830154	105.9917985650	**105.9917830153**
		Std	$4.27E-7$	$9.71E-11$	$1.47E-5$	**1.66E − 14**
P7	50	Mean	723.0049108305	723.0047449232	723.0047449186	**723.0047448968**
		Std	$8.68E-5$	$1.33E-8$	$6.16E-8$	**3.38E − 10**
P8	50	Mean	108.9894309395	108.9894138838	**108.9894138815**	**108.9894138815**
		Std	$8.15E-6$	$1.04E-9$	$3.06E-12$	**2.32E − 12**
P9	50	Mean	455.1346382073	455.1344639826	**455.1344639631**	**455.1344639631**
		Std	$8.83E-5$	$5.60E-8$	**1.49E − 13**	$8.79E-13$
P10	100	Mean	231.0855873314	231.0855570223	231.0855570526	**231.0855570166**
		Std	$1.36E-5$	$2.94E-9$	$1.07E-8$	**1.30E − 11**
P11	100	Mean	297.2586695071	297.2586211205	297.2587315275	**297.2586211143**
		Std	$2.23E-5$	$2.39E-9$	$3.19E-4$	**1.03E − 11**
P12	100	Mean	286.7386559609	286.73862256805	**286.7386225626**	**286.7386225626**
		Std	$2.03E-5$	$3.15E-9$	$1.02E-11$	**1.10E − 12**
P13	250	Mean	549.9466493305	549.9465893483	549.9913513828	**549.94658934110**
		Std	$2.40E-5$	$3.27E-9$	0.0527	**3.91E − 11**
P14	250	Mean	543.7611473895	543.7610648386	543.7655305827	**543.7610648263**
		Std	$3.45E-5$	$5.68E-9$	0.00540	**2.51E − 12**
P15	250	Mean	536.8205045637	536.8203916312	536.8204579818	**536.8203916167**
		Std	$7.09E-5$	$8.00E-9$	$1.08E-4$	**2.83E − 12**
P16	500	Mean	6193.7944120365	6193.7927949375	6193.7927947582	**6193.7927947450**
		Std	$7.68E-4$	$1.13E-7$	$2.26E-8$	**1.42E − 11**
P17	500	Mean	1017.8090988841	1017.8088230685	1017.8110427370	**1017.8088230326**
		Std	$1.63E-4$	$1.55E-8$	0.0043	**2.35E − 11**
P18	500	Mean	4667.0511842616	4667.0492697730	4667.0890166945	**4667.0492696773**
		Std	$9.17E-4$	$4.63E-8$	0.0588	**1.33E − 11**

the connected feasible region bounded by arcs. Further, from these results and according to the results from Table 2, it is clear that the CB-ABC outperformed all other three metaheuristic algorithms as well as the heuristic algorithm with respect to the quality of the obtained results. Although the CB-ABC has more accurate and more stable results than the remaining three metaheuristics, all four metaheuristic approaches perform better than or equal to the heuristic approach with respect to the quality of the obtained results for most of the tested problems.

4.4. Computational Time Analysis. In order to compare the computational cost of the four metaheuristic algorithms, we computed the mean of the CPU times over 30 runs taken by each metaheuristic algorithm. These results are reported in Table 6. The results from Table 6 show that the execution time for each of the metaheuristics approaches linearly increases when the number of the constraints or input points increases.

By comparing computational times for the ABC and CB-ABC algorithms with respect to the FA and E-FA, it is observable that ABC and CB-ABC algorithms are about 4 times faster than the FA and about 20 times faster than the E-FA for the majority of test problems. The computational times of the ABC and CB-ABC algorithms are not significantly different. In addition, when the number of constraints is 500 that time is less than 0.1 seconds. The computational time requirements for the E-FA algorithm are about five times greater compared to the FA and when the number

Table 5: The results of 95%-confidence two-sample t-test over each test problem.

Prb	FA versus E-FA		FA versus ABC		FA versus CB-ABC		E-FA versus ABC		E-FA versus CB-ABC		ABC versus CB-ABC	
	FA	E-FA	FA	ABC	FA	CB-ABC	E-FA	ABC	E-FA	CB-ABC	ABC	CB-ABC
P1	−	+	+	−	−	+	+	−	−	+	−	+
P2	−	+	+	−	−	+	+	−	+	+	−	+
P3	−	+	−	+	−	+	−	+	−	+	+	+
P4	−	+	+	−	−	+	+	−	+	+	−	+
P5	−	+	+	+	−	+	+	−	+	+	−	+
P6	−	+	+	−	−	+	+	−	−	+	−	+
P7	−	+	−	+	−	+	+	+	−	+	+	+
P8	−	+	−	+	−	+	−	+	−	+	+	+
P9	−	+	−	+	−	+	+	+	+	+	+	+
P10	−	+	−	+	−	+	+	−	−	+	−	+
P11	−	+	+	+	−	+	+	+	−	+	+	+
P12	−	+	+	+	−	+	−	+	−	+	+	+
P13	−	+	−	+	−	+	+	−	−	+	−	+
P14	−	+	+	−	−	+	+	−	−	+	−	+
P15	−	+	+	+	−	+	+	−	−	+	−	+
P16	−	+	−	+	−	+	−	+	−	+	−	+
P17	−	+	+	−	−	+	+	−	−	+	−	+
P18	−	+	+	−	−	+	+	−	−	+	−	+
Total	0	18	11	11	0	18	14	7	4	18	6	18

Table 6: Mean of the CPU times (in seconds) obtained from the FA, E-FA, ABC, and CB-ABC algorithms for 18 test instances over 30 runs.

Prob	D	FA	E-FA	ABC	CB-ABC
P1	5	0.022	0.036	0.004	0.004
P2	5	0.020	0.044	0.005	0.004
P3	5	0.022	0.040	0.004	0.004
P4	10	0.025	0.060	0.005	0.006
P5	10	0.022	0.060	0.006	0.006
P6	10	0.025	0.062	0.004	0.005
P7	50	0.056	0.226	0.013	0.016
P8	50	0.052	0.224	0.014	0.014
P9	50	0.057	0.252	0.014	0.010
P10	100	0.090	0.690	0.035	0.043
P11	100	0.090	0.738	0.034	0.043
P12	100	0.086	0.544	0.022	0.025
P13	250	0.200	0.990	0.044	0.044
P14	250	0.216	1.316	0.048	0.051
P15	250	0.194	1.046	0.045	0.042
P16	500	0.376	2.182	0.082	0.093
P17	500	0.360	1.990	0.080	0.091
P18	500	0.374	2.128	0.081	0.094

of constraints or input points is 500 that time is about two seconds.

Compared with the computational time results of the heuristic approach, which are presented in Table 2, it can be seen that the heuristic algorithm requires less computational time than the four metaheuristic algorithms. However, the computational time of the four metaheuristics is reasonable and it can be considered as negligible, since it is less than one second in most cases.

5. Conclusion

The constrained Weber problem with feasible region bounded by arcs represents a problem of a nonconvex optimization.

Finding a global optimum of such a problem is difficult considering the fact that it has multiple locally optimal points within the feasible region. Metaheuristic approaches for solving this problem are suitable choice, since these techniques can obtain quality results in a reasonable amount of time.

The performances of two prominent swarm-intelligence algorithms (the artificial bee colony and firefly algorithm) and their recently proposed improved versions for constrained optimization (the crossover-based artificial bee colony and enhanced firefly algorithm) are compared. The heuristic algorithm based on modified Weiszfeld procedure is also implemented for the purpose of the comparison with the metaheuristic approaches.

The four metaheuristic algorithms are compared on eighteen randomly generated test instances in which the number of input points or constraints increases up to 500. Numerical results indicate that all four metaheuristic algorithms are superior compared to the heuristic approach with respect to the precision of the results, with the notable ascendancy of the CB-ABC algorithm. In terms of the execution time, the ABC and CB-ABC are more efficient than the FA and E-FA. Although these four algorithms require somewhat higher computational cost than the heuristic approach, the CPU times for all these algorithms are reasonable and grow at a linear rate as the number of input points or constraints increases. Finally, it turns out that the CB-ABC algorithm is superior compared to other metaheuristics with respect to the quality of the results, robustness, and computational efficiency.

From this research it can be concluded that metaheuristic approaches can be successfully used for problems with maximum and minimum distance limits. Further, this research encourages the application of the metaheuristic algorithms for solving some other complex constrained optimization problems of practical importance.

Conflicts of Interest

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

Acknowledgments

The second and third authors gratefully acknowledge support from the Project supported by Ministry of Education and Science of Republic of Serbia, Grant no. 174013. The first, third, and fifth authors gratefully acknowledge support from the Project Applying Direct Methods for Digital Image Restoring of the Goce Delčev University. The fourth author gratefully acknowledges support from the Ministry of Education and Science of Russian Federation (Project 2.5527.2017/8.9).

References

[1] R. Z. Farahani and M. Hekmatfar, *Facility Location: Concepts, Models, Algorithms and Case Studies*, Springer-Verlag, Berlin, Germany, 2009.

[2] A. Weber, *Ueber den Standort der Industrien, Erster Teil: Reine Theorie des Standortes*, J.C.B. Mohr, Tübingen, Germany, 1909.

[3] G. Wesolowsky, "The Weber problem: history and perspectives," *Location Science*, vol. 1, pp. 5–23, 1993.

[4] Z. Drezner, K. Klamroth, A. Schöbel, and G. O. Wesolowsky, "The Weber problem," in *Facility Location*, pp. 1–36, Springer, Berlin, Germany, 2002.

[5] P. S. Stanimirović, M. S. Ćirić, L. A. Kazakovtsev, and I. A. Osinuga, "Single-facility Weber location problem based on the lift metric," *Facta Universitatis. Series: Mathematics and Informatics*, vol. 27, no. 2, pp. 175–190, 2012.

[6] R. F. Love, W. G. Truscott, and J. Walker, "Facilities Location: Models and Methods," *International Journal of Machine Learning and Cybernetics, North-Holland, New York*, 1988.

[7] E. Weiszfeld, "Sur le point sur lequel la somme des distances de n points donnes est minimum," *Tohoku Mathematical Journal*, vol. 43, no. 1, pp. 335–386, 1937.

[8] Y. Vardi and C.-H. Zhang, "A modified Weiszfeld algorithm for the Fermat-WEBer location problem," *Mathematical Programming*, vol. 90, no. 3, Ser. A, pp. 559–566, 2001.

[9] C. Szegedy, *Some Applications of the Combinatorial Laplacian*, University of Bonn, 2005.

[10] P. Hansen, N. Mladenović, and E. Taillard, "Heuristic solution of the multisource Weber problem as a p-median problem," *Operations Research Letters*, vol. 22, no. 2-3, pp. 55–62, 1998.

[11] S. Gonzalez-Martin, A. Ferrer, A. A. Juan, and D. Riera, "Solving non-smooth arc routing problems throughout biased-randomized heuristics," *Advances in Intelligent Systems and Computing*, vol. 262, pp. 451–462, 2014.

[12] A. H. Gandomi, X.-S. Yang, S. Talatahari, and A. H. Alavi, "Metaheuristic algorithms in modeling and optimization," in *Metaheuristic Applications in Structures and Infrastructures*, A. H. Gandomi, X.-S. Yang, S. Talatahari, and A. H. Alavi, Eds., pp. 1–24, Elsevier, 2013.

[13] R. Martí and G. Reinelt, *The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization*, vol. 175, Springer, New York, NY, USA, 2011.

[14] A. Afshar, F. Massoumi, A. Afshar, and M. A. Mariño, "State of the art review of ant colony optimization applications in water resource management," *Water Resources Management*, vol. 29, no. 11, pp. 3891–3904, 2015.

[15] A. Banks, J. Vincent, and C. Anyakoha, "A review of particle swarm optimization. I. Background and development," *Natural Computing. An International Journal*, vol. 6, no. 4, pp. 467–484, 2007.

[16] I. Fister Jr., M. Perc, S. M. Kamal, and I. Fister, "A review of chaos-based firefly algorithms: perspectives and research challenges," *Applied Mathematics and Computation*, vol. 252, pp. 155–165, 2015.

[17] I. Fister, X. S. Yang, and D. Fister, "Cuckoo search: a brief literature review," in *Cuckoo Search and Firefly Algorithm: Theory and Applications*, Xin-She Yang, Ed., pp. 49–62, Springer International Publishing.

[18] I. Fister, I. Fister Jr., X.-S. Yang, and J. Brest, "A comprehensive review of firefly algorithms," *Swarm and Evolutionary Computation*, vol. 13, no. 1, pp. 34–46, 2013.

[19] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, pp. 21–57, 2014.

[20] M. Bilal, M. Shams-ur-Rehman, and M. A. Jaffar, "Reconstruction: Image restoration for space variant degradation," *Smart Computing Review*, vol. 3, no. 4, pp. 220–232, 2013.

[21] P. Siarry, *Optimisation in Signal and Image Processing*, Wiley-ISTE, 2009.

[22] X. Yang, S. F. Chien, and T. O. Ting, "Computational Intelligence and Metaheuristic Algorithms with Applications," *The Scientific World Journal*, vol. 2014, Article ID 425853, 4 pages, 2014.

[23] I. Fister Jr., X.-S. Yang, I. Fister, J. Brest, and D. Fister, "A brief review of nature-inspired algorithms for optimization," *Elektrotehniški Vestnik*, vol. 80, no. 3, pp. 1–7, 2013.

[24] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.

[25] B. C. Mohan and R. Baskaran, "A survey: ant colony optimization based recent research and implementation on several engineering domain," *Expert Systems with Applications*, vol. 39, no. 4, pp. 4618–4627, 2012.

[26] R. F. Tavares Neto and M. Godinho Filho, "Literature review regarding ant colony optimization applied to scheduling problems: guidelines for implementation and directions for future research," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 150–161, 2013.

[27] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks (ICNN '95)*, vol. 4, pp. 1942–1948, Perth, Western Australia, November-December 1995.

[28] A. Banks, J. Vincent, and C. Anyakoha, "A review of particle swarm optimization. II. Hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications," *Natural Computing. An International Journal*, vol. 7, no. 1, pp. 109–124, 2008.

[29] A. Khare and S. Rangnekar, "A review of particle swarm optimization and its applications in Solar Photovoltaic system," *Journal Applied Soft Computing*, vol. 13, no. 5, pp. 2997–3006, 2013.

[30] B. Akay and D. Karaboga, "A modified artificial bee colony algorithm for real-parameter optimization," *Information Sciences*, vol. 192, pp. 120–142, 2012.

[31] D. Karaboga, "An Idea Based on Honey Bee Swarm for Numerical Optimization," Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[32] A. Baykasoglu, L. Ozbakir, and P. Tapkan, "Artificial bee colony algorithm and its application to generalized assignment problem," in *Swarm Intelligence, Focus on Ant and Particle Swarm Optimization*, F. T. S. Chan and M. K. Tiwari, Eds., pp. 113–144, I-Tech Education and Publishing, Vienna, Austria, 2007.

[33] S. Zhang and S. Liu, "A novel artificial bee colony algorithm for function optimization," *Mathematical Problems in Engineering*, vol. 2015, Article ID 129271, 10 pages, 2015.

[34] Y. Liang, Z. Wan, and D. Fang, "An improved artificial bee colony algorithm for solving constrained optimization problems," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 3, pp. 739–754, 2017.

[35] T. K. Sharma and M. Pant, "Shuffled Artificial Bee Colony Algorithm," *Soft Computing*, pp. 1–20, 2016.

[36] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications*, vol. 5792 of *Lecture Notes in Computer Science*, pp. 169–178, Springer, Berlin, 2009.

[37] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Mixed variable structural optimization using firefly algorithm," *Computers and Structures*, vol. 89, no. 23-24, pp. 2325–2336, 2011.

[38] M. Huang, J. Yuan, and J. Xiao, "An adapted firefly algorithm for product development project scheduling with fuzzy activity duration," *Mathematical Problems in Engineering*, vol. 2015, Article ID 973291, 11 pages, 2015.

[39] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NABIC '09)*, pp. 210–214, Coimbatore, India, December 2009.

[40] Y. Lin, C. Zhang, and Z. Liang, "Cuckoo search algorithm with hybrid factor using dimensional distance," *Mathematical Problems in Engineering*, vol. 2016, Article ID 4839763, 11 pages, 2016.

[41] A. R. Yildiz, "Cuckoo search algorithm for the selection of optimal machining parameters in milling operations," *International Journal of Advanced Manufacturing Technology*, vol. 64, no. 1–4, pp. 55–61, 2013.

[42] M. Hakan Akyüz, T. Öncan, and I. Kuban Altinel, "Beam search heuristics for the single and multi-commodity capacitated multi-facility weber problems," *Computers and Operations Research*, vol. 40, no. 12, pp. 3056–3068, 2013.

[43] J.-L. Jiang and X.-M. Yuan, "A heuristic algorithm for constrained multi-source Weber problem—the variational inequality approach," *European Journal of Operational Research*, vol. 187, no. 2, pp. 357–370, 2008.

[44] L. A. Kazakovtsev, "Algorithm for constrained Weber problem with feasible region bounded by arcs," *Facta Universitatis. Series: Mathematics and Informatics*, vol. 28, no. 3, pp. 271–284, 2013.

[45] M. Luis, S. Salhi, and G. Nagy, "Region-rejection based heuristics for the capacitated multi-source Weber problem," *Computers and Operations Research*, vol. 36, no. 6, pp. 2007–2017, 2009.

[46] S. M. H. Manzour-Al-Ajdad, S. A. Torabi, and K. Eshghi, "Single-source capacitated multi-facility weber problem—an iterative two phase heuristic algorithm," *Computers and Operations Research*, vol. 39, no. 7, pp. 1465–1476, 2012.

[47] M. Bischoff and K. Klamroth, "An efficient solution method for Weber problems with barriers based on genetic algorithms," *European Journal of Operational Research*, vol. 177, no. 1, pp. 22–41, 2007.

[48] A. Ghaderi, M. S. Jabalameli, F. Barzinpour, and R. Rahmaniani, "An efficient hybrid particle swarm optimization algorithm for solving the uncapacitated continuous location-allocation problem," *Networks and Spatial Economics*, vol. 12, no. 3, pp. 421–439, 2012.

[49] H. G. Gharravi and M. S. Farham, "Applying metaheuristic approaches on the single facility location problem with polygonal barriers," *International Journal of Metaheuristics*, vol. 3, no. 4, p. 348, 2014.

[50] M. Saleh Farham, H. Süral, and C. Iyigun, "The Weber problem in congested regions with entry and exit points," *Computers and Operations Research*, vol. 62, pp. 177–183, 2015.

[51] N. Javadian, R. Tavakkoli-Moghaddam, M. Amiri-Aref, and S. Shiripour, "Two meta-heuristics for a multi-period minisum location-relocation problem with line restriction," *International Journal of Advanced Manufacturing Technology*, vol. 71, no. 5-8, pp. 1033–1048, 2014.

[52] N. Mohammadi, M. R. Malek, and A. A. Alesheikh, "A new GA based solution for capacitated multi source Weber problem," *International Journal of Computational Intelligence Systems*, vol. 3, no. 5, pp. 514–521, 2010.

[53] D. Karaboga and B. Akay, "A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems," *Applied Soft Computing Journal*, vol. 11, no. 3, pp. 3021–3031, 2011.

[54] I. Brajevic, "Crossover-based artificial bee colony algorithm for constrained optimization problems," *Neural Computing and Applications*, vol. 26, no. 7, pp. 1587–1601, 2015.

[55] I. Brajevic and J. Ignjatović, "An enhanced firefly algorithm for mixed variable structural optimization problems," *Facta Universitatis. Series: Mathematics and Informatics*, vol. 30, no. 4, pp. 401–418, 2015.

[56] L. A. Kazakovtsev and P. S. Stanimirovic, "Algorithm for Weber problem with a metric based on the initial fare," *Journal of Applied Mathematics & Informatics*, vol. 33, no. 1-2, pp. 157–172, 2015.

[57] Z. Drezner, C. Scott, and J.-S. Song, "The central warehouse location problem revisited," *IMA Journal of Management Mathematics*, vol. 14, no. 4, pp. 321–336 (2004), 2003.

[58] P. Hansen, D. Peeters, and J. F. Thisse, "Algorithm for a constrained weber problem," *Management Science*, vol. 28, no. 11, pp. 1285–1295, 1982.

[59] P. Hansen, D. Peeters, and J.-F. Thisse, "Constrained Location and the Weber-Rawls Problem," *North-Holland Mathematics Studies*, vol. 59, no. C, pp. 147–166, 1981.

[60] D. Karaboga and B. Basturk, *Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems*, Springer-Verlag, Berlin, Germany, LNAI 4529: IFSA'07, 2007.

[61] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2–4, pp. 311–338, 2000.

[62] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, UK, 2010.

[63] X.-S. Yang, S. Deb, M. Loomes, and M. Karamanoglu, "A framework for self-tuning optimization algorithm," *Neural Computing and Applications*, vol. 23, no. 7-8, pp. 2051–2057, 2013.

[64] E. Mezura-Montes and C. A. Coello Coello, "Constraint-handling in nature-inspired numerical optimization: past, present and future," *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 173–194, 2011.

[65] M. Mernik, S.-H. Liu, D. Karaboga, and M. Crepinšek, "On clarifying misconceptions when comparing variants of the artificial bee colony algorithm by offering a new implementation," *Information Sciences*, vol. 291, pp. 115–127, 2015.

[66] I. Fister, X. Yang, J. Brest, and I. Fister Jr., "Modified firefly algorithm using quaternion representation," *Expert Systems with Applications*, vol. 40, no. 18, pp. 7220–7230, 2013.

[67] E. Mezura-Montes and O. Cetina-Domínguez, "Empirical analysis of a modified artificial bee colony for constrained numerical optimization," *Applied Mathematics and Computation*, vol. 218, no. 22, pp. 10943–10973, 2012.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 2324316, 14 pages
https://doi.org/10.1155/2017/2324316

Research Article

A Hybrid Method for Truss Mass Minimization considering Uncertainties

Herbert M. Gomes[1] and Leandro L. Corso[2]

[1]*Federal University of Rio Grande do Sul, Av. Sarmento Leite 425, Sala 202, 2° Andar, 90050-170 Porto Alegre, RS, Brazil*
[2]*University of Caxias do Sul, R. Francisco Getúlio Vargas 1130, 95070-560 Caxias do Sul, RS, Brazil*

Correspondence should be addressed to Herbert M. Gomes; herbert@mecanica.ufrgs.br

Received 10 November 2016; Revised 2 March 2017; Accepted 16 May 2017; Published 13 June 2017

Academic Editor: David Greiner

In real-world structural problems, a number of factors may cause geometric imperfections, load variability, or even uncertainties in material properties. Therefore, a deterministic optimization procedure may fail to account such uncertainties present in the actual system leading to optimum designs that are not reliable; the designed system may show excessive safety or sometimes not sufficient reliability to carry applied load due to uncertainties. In this paper, we introduce a hybrid reliability-based design optimization (RBDO) algorithm based on the genetic operations of Genetic Algorithm, the position and velocity update of the Particle Swarm Algorithm (for global exploration), and the sequential quadratic programming, for local search. The First-Order Reliability Method is used to account uncertainty in design and parameter variables and to evaluate the associated reliability. The hybrid method is analyzed based on RBDO benchmark examples that range from simple to complex truss parametric sizing optimizations with stress, displacements, and frequency deterministic and probabilistic constraints. The proposed final problem, which cannot be handled by single loop RBDO algorithms, highlights the importance of the proposed approach in cases where the discrete design variables are also random variables.

1. Introduction

In the engineering science, the cost reduction in manufacturing is pursued in order to obtain efficiency and save time in production. Although the use of optimization methods is increasing, in practical terms, the design parameters and/or design variables used in deterministic optimization procedures may present uncertainties. It means that analyzing the responses obtained by deterministic optimization procedures in terms of failure probability, nonacceptable levels may be present in the engineering point of view. According to [1], uncertainties in real-world optimization problems include factors such as data incompleteness, mathematical model inaccuracies, and environmental condition variation, just to name a few. This directly affects the optimization problem since in, a structural engineering design, economy and safety are competing goals [2]. One way to link such uncertainties in an optimization problem is using a reliability index (β), a measure of the degree of reliability in the design, according to [1]. This procedure is referred to as reliability-based design optimization (RBDO); in this case, the reliability index becomes an extra constraint to the optimization problem. According to [3], in the last decade, the RBDO significance and conceptual and mathematical complexity have been intensively studied. The optimization procedure may require a high computational cost [4]. For this reason, several authors have studied methods that use single loop RBDO [3, 5–7]. These authors have applied reliability-based design optimization strategies like SAP (Sequential Approximate Programming) and SORA (Sequential Optimization and Reliability Assessment), which are capable of solving the majority of practical cases, when multiple modes of failure are present, but for linear and smooth functions. According to [8], the use of approximation procedures in the sequential optimization and the evaluation of the reliability index in a single step may result in spurious optimal points. That is evident since these methods are deterministic and they show a tendency of convergence to local minima points

due to possible nonlinear function behavior in probabilistic problems. In order to increase robustness, authors [9] proposed to construct a response surface that is the product of individual performance functions and be sued as surrogate to get optimal design solutions. Despite the large amount of literature in the theme, a good introduction to the RBDO subject can be found in [10].

In this article, a hybrid RBDO methodology is proposed and applied to a range of spatial trusses in order to find the optimum parameters for minimum mass, considering maximum allowable deflections and limited stress. The First-Order Reliability Method (FORM) is used to evaluate the reliability index (β) along the optimization steps. Due to the high computational cost of the traditional algorithms to solve nonconvex optimization problems, we propose a hybrid optimization method. Comparisons about the computational cost and quality of the solution are performed using as baseline standard structural problems. The proposed hybrid global optimization method based on existent global search (metaheuristic) and deterministic algorithms is detailed. This new method is presented and discussed by using reliability-based optimization on structural truss examples.

2. Reliability Index Evaluation

As stated by [11], reliability analysis can be applied to many engineering fields such as in aeronautical, mechanical, and civil problems. By definition, the reliability is the complement of the probability of failure, that is, the likelihood of failure of a specific event or set of events from a complex system. A limit state function that relates the failure event (violation of specific set of constraints) as function of several variables is stated by the mathematical expression:

$$g\left(X_1, \ldots, X_n\right) = 0, \tag{1}$$

where g means the limit state function that defines a constraint. Some design variables may present random components. In case X_i is a set of n random variables that affects that constraint, this limit state function becomes also random and some probability of violation is implicit [11, 12]. $g(\cdot) \leq 0$ means that the system is in the failure domain, D, and $g(\cdot) > 0$ means that the system is in the safe domain (constraint was not violated). The probability of failure can be evaluated by the integration of the joint probability density function $f_X(X_1, \ldots, X_n)$ as indicated in [11]:

$$P_f = \int \cdots \int_D f_X\left(X_1, \ldots, X_n\right) dX_1 \cdots dX_n \cong \Phi\left(-\beta\right), \tag{2}$$

where the failure domain is defined by D means that $g(\cdot) \leq 0$, Φ is the cumulative standard distribution function, and β is a safety index metric. Equation (2) presents a close solution in some particular cases where $f_X(\cdot)$ is Gaussian and for linear and quadratic $g(\cdot)$. However, it is difficult to be handled in case of several random variables n. In this case, Monte Carlo (MC) Simulation can be used to get approximate asymptotic solutions. Moreover, statistic values for function $f_X(X)$ are not known a priori and the number of MC samples frequently is not enough to ensure confidence.

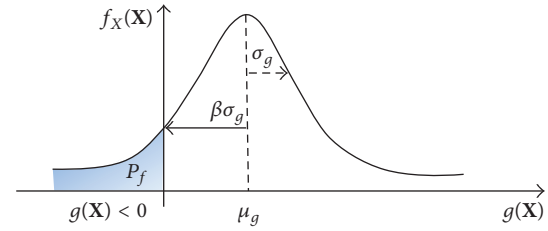


Figure 1: Concept of probability of failure and the reliability index β for the first-order approximation.

So, a very simple but not so robust way to get and estimate for the reliability index (that is related to the probability of survival) β is using the probability density function first and second moments (mean and variance) for the limit state function $g(X_1, \ldots, X_n)$. When the limit state function $g(X)$ is linear and the random variables are normally distributed and uncorrelated, the reliability index β can be approximated by the following (see Figure 1):

$$\beta = \frac{\mu_g}{\sigma_g}, \tag{3}$$

where μ_g and σ_g represent, respectively, the mean value and the standard deviation (square root of variance) for function $g(X)$.

Let σ_i be the mechanical stress that can be measured in a loaded component i, assuming a failure situation where this value exceeds the imposed material strength limit value (σ_{lim}). Equation (1) can be rewritten, for the limit state function, as follows:

$$g_i\left(\sigma_i\right) = 1 - \frac{\sigma_i}{\sigma_{\text{lim}}}. \tag{4}$$

Linearization of function $g(X)$ by a Taylor expansion up to linear terms can be used to get approximated values for μ_g and σ_g when the limit state functions are nonlinear. The point around the linearization performed affects μ_g and σ_g values. A method to obtain the reliability index β that is independent of the limit state function formulation is known as AFOSM (Advanced First-Order Second Moment) and was first proposed by [13]. For uncorrelated Gaussian random variables X_i, they are transformed into normalized ones U_i by the transformation:

$$U_i = \Phi^{-1}\left[F_{X_i}\left(x_i\right)\right] = T\left(X_i\right), \tag{5}$$

where $F_{X_i}(x_i)$ and $\Phi^{-1}(\cdot)$ are the cumulative distribution function of the random variable X_i and the inverse of the cumulative standard Gaussian distribution, respectively. In this way, the limit state function in the real space X is transformed to the uncorrelated normalized space U, so

$$h\left(U\right) \cong g\left(X\right). \tag{6}$$

The linearization of the limit state function $h(U)$ is performed at the U^* point that presents the shorter distance to the origin of the uncorrelated space U and that ensures

$h(U) = 0$. The U^* point is called the design point and the reliability index β is evaluated as mentioned previously as

$$\begin{aligned} \beta &= \min \quad (\|U\|) \\ \text{subject to} \quad & h(U) \leq 0. \end{aligned} \tag{7}$$

2.1. Rackwitz-Fiessler Algorithm. In order to solve (7), the efficient algorithm proposed by Rackwitz and Fiessler [13] is used. It can be described in the following steps.

Step 1. Define the limit state function for the problem $g(\mathbf{X}) = 0$.

Step 2. Assume initial values for the design point in the real space $\mathbf{X}^* = (X_1, \ldots, X_n)^T$ and evaluate the corresponding values for the limit state function $g(\mathbf{X})$ (e.g., assume an initial design point as the mean values of random variables).

Step 3. Evaluate the equivalent Gaussian mean value and standard deviation for the random variables:

$$\begin{aligned} \sigma_{X_i}^N &= \frac{\phi\left(\Phi^{-1}\left[F_{X_i}(x_i)\right]\right)}{f_{X_i}(x_i)}, \\ \mu_{X_i}^N &= X_i - \sigma_{X_i}^N \Phi^{-1}\left[F_{X_i}(x_i)\right]. \end{aligned} \tag{8}$$

Step 4. Transform random variables from real space $\mathbf{X}$ to normal uncorrelated $\mathbf{U}$. The design variables values at the design point will be evaluated as follows:

$$U_i = \frac{X_i - \mu_{X_i}^N}{\sigma_{X_i}^N}. \tag{9}$$

Step 5. Evaluate the sensitivities $\partial g(X)/\partial X_i$ at the design point $\mathbf{X}^*$.

Step 6. Evaluate the partial derivatives $\partial g(X)/\partial U_i$ in the normal uncorrelated space using the chain rule:

$$\frac{\partial g(X)}{\partial U_i} = \frac{\partial g(X)}{\partial X_i}\frac{\partial X_i}{\partial U_i}. \tag{10}$$

Step 7. Evaluate the new value for the design point U_i^* in the normal uncorrelated space using the recurrent equation:

$$U_{i,k+1}^* = \left[\nabla g\left(U_{i,k}^*\right)^T U_{i,k}^* - g\left(U_{i,k}^*\right)\right]\frac{\nabla g\left(U_{i,k}^*\right)}{\left|\nabla g\left(U_{i,k}^*\right)\right|}. \tag{11}$$

Step 8. Evaluate the distance from the origin to this new point and estimate the new reliability index:

$$\beta = \|U\| = \sqrt{\sum_{i=1}^{n}\left(U_i^*\right)^2}. \tag{12}$$

Step 9. Check convergence of β along iterations using a predefined tolerance.

Step 10. Evaluate the random variables at the new design point using

$$X_i = \mu_{X_i}^N + \sigma_{X_i}^N U_i^*. \tag{13}$$

Step 11. Evaluate $g(\mathbf{X})$ value for the new random variables and verify a final convergence criterion, for instance, $\Delta g(\mathbf{X}) <$ tolerance and $\Delta \mathbf{X} <$ tolerance.

Step 12. If both criteria are met, stop iterating; otherwise, repeat Steps 3–11.

This algorithm assumes all the random variables as noncorrelated in the original actual space. If there exists correlation between random variables, using Cholesky decomposition of the covariance matrix, the correlated variables can be transformed to uncorrelated ones and the algorithm is still valid [8, 11, 14]. This transformation is presented in

$$\mathbf{Z} = \mathbf{L}^{-1}(\mathbf{X} - \boldsymbol{\mu}) \quad \text{with } \operatorname{cov}(\mathbf{X}, \mathbf{X}) = \mathbf{C} = \mathbf{L}\mathbf{L}^T. \tag{14}$$

3. Reliability-Based Design Optimization (RBDO)

In the RBDO, the objective function should satisfy predefined probabilistic constraints, which are set as new constraints to the problem. Probability failure analyses are performed along the optimization process in order to check if the probabilistic constraints are met. This is used to guide the optimization to the minimum weight that complies with the target reliability levels. The easier formulation for RBDO implements the algorithm with a double loop where the optimization is split into two stages: (a) on a first stage, the objective function optimization is performed focusing on the design variables; (b) on a second stage, the optimization is performed focusing on the random variables starting from the design variables from the outer loop. More details can be found in [15]. A deterministic model for the minimization can be defined generally as follows [16]:

$$\begin{aligned} \text{Minimize} \quad & f(v_d, \mathbf{p}) \\ \text{Subject to:} \quad & h_i(v_d, \mathbf{p}) = 0 \quad i = 1, \ldots, m \\ & g_j(v_d, \mathbf{p}) < 0 \quad j = 1, \ldots, n \\ & v_{dl} \leq v_d \leq v_{du}, \end{aligned} \tag{15}$$

where v_d is the vector of design variables, $\mathbf{p}$ is the vector of parameters for the optimization problem, $h_i(\cdot) = 0$ is ith model's equality constraint from a total of m equality constraints, g_j are the n inequality constraints, and v_{du} and v_{dl} are the vectors that contains upper and lower values for the design variables. However, a deterministic optimization does not consider the uncertainties in the design variables nor fixed design parameters. In RBDO, the probabilistic constraints are added increasing the set of constraint equations. Since the reliability index can be defined in terms of the

accumulated probability function for the limit state function (and vice versa), the following holds:

$$P_f(v_d, \mathbf{p}) = \Phi(-\beta) \quad \text{or } \beta = -\Phi^{-1}\left[P_f(v_d, \mathbf{p})\right], \tag{16}$$

where Φ is the cumulative standard distribution function. In this article, the reliability constraint is formulated as follows:

$$g_j(v_d, \mathbf{p}) = 1 - \frac{\beta}{\beta_{\text{target}}} < 0 \quad j = nr + 1, \ldots, np, \tag{17}$$

where $g_j(\cdot)$ is the probabilistic constraint defined by the dimensionless ratio between the evaluated reliability index β and the target reliability index β_{target}. This means that if the evaluated reliability index β during the optimization is larger than the reliability target index β_{target}, then $g_j(\cdot) \leq 0$ and the probabilistic criterion is met. Otherwise, a penalization for the objective function will take place. Numerically, during the optimization process, the failure function needs to be evaluated a number of times so one can find the probability of failure value (or conversely the reliability index). In a RBDO, both parameter vector $\mathbf{p}$ and design variables v_d can be random variables.

Figure 2 shows a geometric interpretation for the difference between a deterministic and a reliability-based optimization. The implementation of the optimization may be performed using two different approaches: RIA (Reliability Index Approach) or PMA (Performance Measure Approach).

3.1. Reliability Index Approach (RIA). This approximation for the reliability constraint is treated as an extra constraint that is formulated by the reliability index β (for the sake of simplicity, in the uncorrelated design space). So, the following can be written:

$$\begin{aligned}
&\text{Minimize} && f(\mathbf{u}, \mathbf{p}) \\
&\text{Subject to} && \mathbf{p}^L < \mathbf{p} < \mathbf{p}^U \\
&&& g_i(E[\mathbf{u}, \mathbf{p}]) < 0 \quad i = 1, \ldots, m \\
&&& h_j(E[\mathbf{u}, \mathbf{p}]) = 0 \quad j = 1, \ldots, n \\
&&& \beta_{\text{target}} - \beta\left(f_k(\mathbf{u}, \mathbf{p}) = 0\right) < 0 \\
&&&\qquad\qquad k = 1, \ldots, p \\
&&& \beta_{\text{target}} - \beta\left(f_l(\mathbf{u}, \mathbf{p}) = 0\right) = 0 \\
&&&\qquad\qquad l = 1, \ldots, q,
\end{aligned} \tag{18}$$

where $\mathbf{u}$ is the standard uncorrelated random variables and g_i and h_j are the m inequality and n equality deterministic constraints with corresponding p probabilistic equalities and q probabilistic inequalities constraints. $E[\cdot]$ means the expected values. In order to find β, the reliability problem is defined as $\beta = \min(\|U\|)$, subject to $g(U) \leq 0$.

3.2. Performance Measure Approach (PMA). This formulation is performed with the inverse of the previous analysis by RIA, in such a way that the following can be written:

$$\begin{aligned}
&\text{Minimize} && f(\mathbf{u}, \mathbf{p}) \\
&\text{Subject to} && \mathbf{p}^L < \mathbf{p} < \mathbf{p}^U \\
&&& g_i(E[\mathbf{u}, \mathbf{p}]) < 0 \quad i = 1, \ldots, m \\
&&& h_j(E[\mathbf{u}, \mathbf{p}]) = 0 \quad j = 1, \ldots, n \\
&&& g_k\left(\|\mathbf{u}\| = \beta_{\text{target}}, \mathbf{p}\right) - g_k(\mathbf{u}, \mathbf{p}) < 0 \\
&&&\qquad\qquad k = 1, \ldots, p \\
&&& h_l\left(\|\mathbf{u}\| = \beta_{\text{target}}, \mathbf{p}\right) - h_l(\mathbf{u}, \mathbf{p}) = 0 \\
&&&\qquad\qquad l = 1, \ldots, q,
\end{aligned} \tag{19}$$

where $\mathbf{u}$ is the vector the normalized uncorrelated random variables, g_i and h_j are the m and n inequality and equality constraints, and g_k and h_l are the p and q probabilistic inequalities constraints, respectively. So, differently from RIA, for a fixed reliability index $\|\mathbf{u}\| = \beta_{\text{target}}$, the equality and inequality constraints are ensured beforehand, so a line search for the point where hypersphere $\|\mathbf{u}\| = \beta_{\text{target}}$ cuts the constraints should be performed. The advantages and disadvantages in using this or the previous PMA formulation can be found in [17].

4. Proposed Hybrid Optimization Method

Evolutionary algorithms are widely considered powerful and robust techniques for global optimization and can be used to solve full-scale problems which contain several local optima [18]. Although the implementation of such methods is easy, they can demand a high computational cost due to high number of function calls. Therefore, they are not recommended for problems in which the objective function presents high computational cost [19]. Besides, it is important that the algorithm parameters be tuned adequately in order to avoid premature convergence of the algorithm.

Pioneering works with hybrid algorithms for optimization have been performed by [20] with the so-called Augmented Lagrangian Genetic Algorithm that could deal with constraints by penalization functions. The algorithm has an outer loop to update penalization parameter based on constraints values and an inner loop for traditional GA using the penalized fitness function. The algorithm can also avoid trial and error selection and manually increase penalty constants in the optimization problem. In a different way, the hybrid method proposed here is based on the use of genetic operators of GA like mutation, crossover, and elitism associated with a gradient search by SQP algorithm performed on best individual of the GA population. Besides, 20% of the best individuals will pass by a hybrid PSO operation, which allows position update based on velocity and global and local cognitive coefficients. These operators are inherited from the traditional PSO algorithm [21]. Using

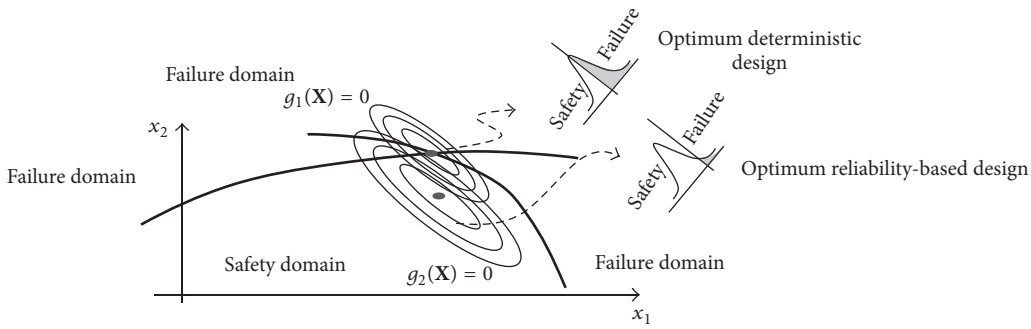


Figure 2: Geometric definition for the reliability-based design optimization.

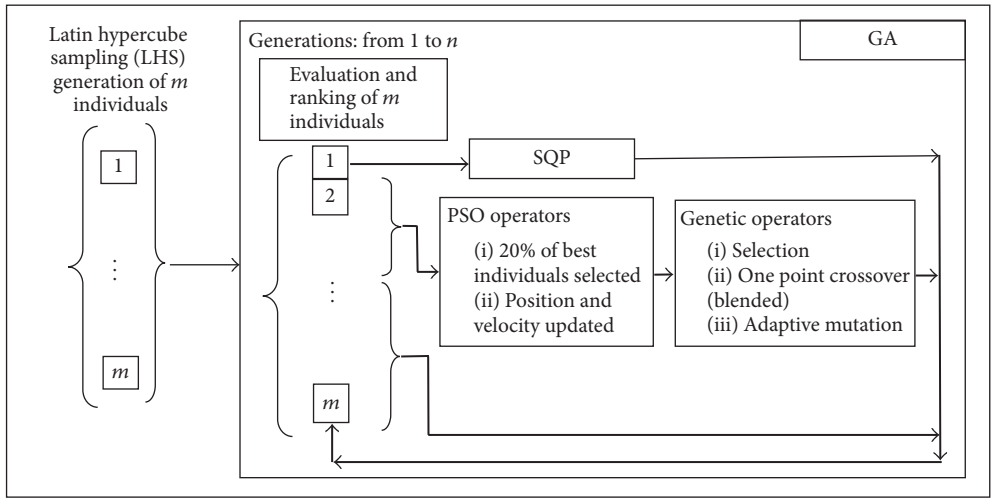


Figure 3: Flowchart schematic for the HGPS.

elitism, the best individual of the group passes to a cost function enhancement by sequential quadratic programming (SQP) and then is sent to the next generation. This method is hereafter called HGPS. The main goal of hybrid method is to accelerate the convergence rate of a global search while maintaining the exploration capability. The SQP method is justified by the intrinsic speed in finding local optima, while GA and PSO preserve the diversity in the exploration of new regions of the search space.

Although the convergence rate can be accelerated, getting stuck in local optima still remains possible. For this reason, the adaptive mutation is inherited from GA. PSO operators (like momentum) are also used in order to help the escape from local minima. Initial tests carried out by [22, 23] showed that loss in diversity may still occur, resulting in premature convergence; thus adaptive mutation operation is recommended mainly in cases where the problem has a complex solution (e.g., in nonsmooth or discontinuous functions). Their use is also based on the good results found in preliminary tests in literature [24].

A simple sketch illustrating the idea of mixing best features from several algorithms is shown in Figure 3, considering m individuals and n generations. Algorithm 1 presents the corresponding pseudocode with the steps followed by the proposed algorithm. It is important to notice that the codification for the design variables (in a GA sense) is the real one; that is, each individual i for the generation t is represented by $\mathbf{b}_{i,t} = (x_1, x_2, \ldots, x_{nc})$ where nc is the number of genes (design variables).

In the pseudocode in Algorithm 1, $x_{i,j}^k$ is the current value of the design variable j of the particle i of the generation k of the GA. The $v_{i,j}^{k+1}$ is the updated velocity of the design variable j of the particle i of the generation k of the GA. The $xlbest_{i,j}^k$ is the best design variable j from the selected 20% of best individuals, and $xgbest_j^k$ is the best design variable j of generation k. Related to the stopping criteria, a criterion is chosen that is based on the coefficient of variation (σ/μ) of the objective function of the elite individuals within a defined number of generations. For the following problems, it has been considered that, within five consecutive generations, if the change in the coefficient of variation of the objective function of the elite individuals is lower than a defined tolerance, the convergence criterion is met.

This hybridized methodology (HGPS) was proposed aiming at convergence acceleration and maintaining the diversity of individuals to avoid premature convergence. Therefore, this approach takes advantage from the use of adaptive mutation, PSO operator performed on individuals from consecutive generations, and SQP improvements on the

(1) Initialize generations, Initialize Population size: "m", mutation probability: "P_m", crossover probability: "P_c", number of genes per individual: "nc", upper and lower bounds values for each gene: "$x_{\max}(nc)$", "$x_{\min}(nc)$".
Main Loop (while stopping criteria are not met)
(2) Evaluate and rank the population of individuals by the objective function vector.
(3) Generation of new population
(3.1) *SQP method:* The best individual of the population is the starting point of search SQP method, which replaces the same point in the next generation.
(3.2) *PSO method:* 20% of the best individuals suffers PSO operations, constrained to $x_{\max}$ and $x_{\min}$, According to:

$$v_{i,j}^{k+1} = \chi[\varpi v_i^k + c_1 r_1 (xlbest_{i,j}^k - x_{i,j}^k) + c_2 r_2 (xgbest_j^k - x_{i,j}^k)]$$
$$x_{i,j}^{k+1} = x_{i,j}^k + v_{i,j}^{k+1}$$

This work assumes $c_1 = c_2 = 1{,}5$ and $\varpi = 0.4(1 + \min * (\text{cov}, 0.6)$, according to [25]).
(3.3) *GA Method (genetic operators)*
(3.3.1) *Crossover* It is assumed $P_c = 0.8$ according to [26]
"One point Recombination"
Loop $i = 1$ to $m - 1$ Step 2
If random$(0, 1) < P_c$ do
$\alpha = 0.5$
$\Delta = \max[b_{i,t}(k), b_{i+1,t}(k)] - \min[b_{i,t}(k), b_{i+1,t}(k)]$
$b_{i,t}(k) = \text{random}(\min[b_{i,t}(k), b_{i+1,t}(k)] - \alpha\Delta, \max[b_{i,t}(k), b_{i+1,t}(k)] + \alpha\Delta)$
End If
End Loop i
(3.3.2) *Adaptive Mutation* [22]
Loop $i = 1$ to m
If $f_i > f_{\min}$ then $P_m = 0.2$
Else

$$P_m = \frac{0{,}5(f_i - f_{\min})}{(\overline{f} - f_{\min})}$$

End If
End Loop i
If random$(0, 1) < P_m$ do
$k = \text{random}(0, 1) * nc$
$b_{i,t}(k) = \text{random}(x_{\max}(k), x_{\min}(k))$
End If
End of the Main Loop

Algorithm 1: Pseudocode for the HGPS.

solution of the best individual found so far. The probability of mutation is not a constant value but varies according to the fitness function used by the GA, that is, the mean of individual objective function value $\overline{f}$. According to [22], the value of P_m should depend not only on $\overline{f} - f_{\min}$ (a measure of convergence) but also on the fitness function value f_i of the solution. The closer f_i is to $f_{\min}$, the smaller P_m should be, assuming zero at $f_i = f_{\min}$. To prevent the overshooting of P_m beyond 1.0, the constraint for P_m is achieved by setting $P_m = 0.2$ whenever $f_i > f_{\min}$ (since such solutions need to be completely disrupted); otherwise, it is set to $P_m = 0.5(f_i - f_{\min})/(\overline{f} - f_{\min})$.

5. Results

The comparisons are performed using only simple optimization algorithms like GA (Genetic Algorithm), DE (Differential Evolution), PSO (Particle Swarm Optimization), FMA (Firefly Metaheuristic Algorithm), and the proposed HGPS algorithm. When possible, the deterministic SQP (sequential quadratic programming) algorithm will be used for comparisons. Specifically, in example 3, where discrete design variables are present, this comparison will not be possible. Since there is no theorem that gives the best parameters values for the algorithms that may serve to any problem, based on literature recommendations and some trial and error, some effort was made in finding those parameters for each metaheuristic algorithm. In all studied cases, in order to have fair comparisons, the same set of optimization parameters was applied that were previously tuned for best performance in each algorithm. Table 1 shows such parameters for the tested algorithms.

5.1. Case 1: 10 Bar Truss Problem. This classical model was presented by [6] that performed the minimization of the truss mass with stress and displacement constraints taking into account reliability index for stress and displacements. The analyzed truss is presented in Figure 4. The bar length $a = 9.144$ m (360 in). The ten bar cross sections are the design variables. The lower and upper limit for the design variable

Table 1: Summary of parameters for each optimization method.

GA		DE		PSO		FMA		HGPS	
Number of individuals	200	Number of individuals	50	Number of particles	50	Number of particles	50	Number of individuals	50
Crossover	one point	Weighting ratio	1.5	Social constant	2.01	Beta	1.0	Social constant/cognitive constant	2.01/2.01
Probability of crossover	80%	Crossover prob.	0.8	Cognitive constant	2.01	$\text{Beta}_{\min}$	0.1	Probability of crossover	80%
Probability of mutation	1%	Search Starting point	$(x_{\max} + x_{\min})/2$	Inertia moment	0.8	Gamma	15	Probability of mutation	1%
Tolerance (successive Obj. Func., design variable)	10^{-6}	Tolerance (successive Obj. Func., design variable or mesh size)	10^{-6}	Tolerance (successive Obj. Func., design variable)	10^{-6}	Tolerance (successive Obj. Func., design variable)	10^{-6}	Tolerance (successive Obj. Func., design variable)	10^{-6}
Maximum number of generations	200	Maximum number of iterations	800	Maximum number of iterations	800	Maximum number of iterations	800	Maximum number of iterations	800

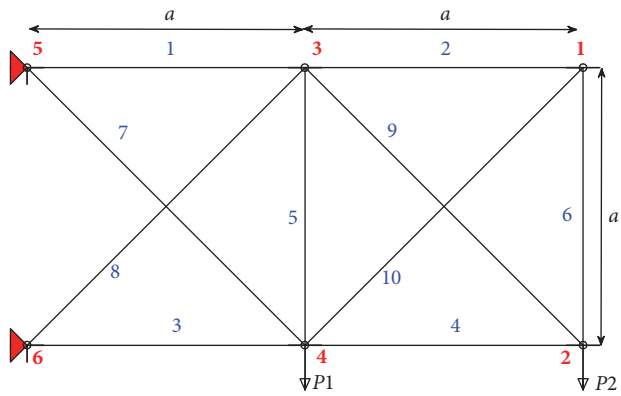


Figure 4: 10 bar truss geometry, nodes, and connectivity.

are $6.45 \times 10^{-6}\,\text{m}^2$ ($0.01\,\text{in}^2$) and $1.61 \times 10^{-2}\,\text{m}^2$ ($25\,\text{in}^2$), respectively. The member's elastic modulus are $E = 6.895 \times 10^{10}$ Pa (10^7 Psi) and assumed deterministic. The applied loads are assumed random and uncorrelated, following a lognormal distribution, with mean $\mu = 4.448 \times 10^5$ N (1.0×10^5 lbf) and standard deviation $\sigma = 2.224 \times 10^4$ N (5.0×10^3 lbf). Material strength is assumed as Gaussian random variable with mean and standard deviation being equal to 1.724×10^8 Pa (2.5×10^4 Psi) and 1.724×10^7 Pa (2.5×10^3 Psi), respectively. The vertical displacement has a constraint of ± 0.1143 m (± 4.5 in).

The deterministic constraints for stress and displacements are treated as limit state functions from the probabilistic point of view. The reliability indexes for the stress values in any member (tension or compression) and for the maximum vertical displacement at any node are set to 3.0. Therefore, the mathematical RBDO problem can be stated as follows:

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{A}) = \rho \sum_{i=1}^{n} A_i L_i \\ \text{Subject to} \quad & |\sigma_i| \le 1.724 \times 10^8 \text{ Pa} \quad i = 1, \ldots, n \\ & |d_j| \le 0.1143 \text{ m} \quad j = 1, \ldots, nn \\ & \beta_s(\mathbf{u}, \mathbf{p}) \ge \beta_{Ls} \\ & \beta_d(\mathbf{u}, \mathbf{p}) \ge \beta_{Ld} \\ & \mathbf{p}_L \le \mathbf{p} \le \mathbf{p}_U, \end{aligned} \tag{20}$$

where $\mathbf{p}$ is the vector of design variable (member area), β_{Ld}, β_{Ls} are the target reliability indexes, and $\beta_s(\mathbf{u}, \mathbf{p})$ and $\beta_d(\mathbf{u}, \mathbf{p})$ are the reliability indexes for stress and displacement, respectively that are function of vector of design variables $\mathbf{p}$ and vector of random variables $\mathbf{u}$ (two loads and the material strength). The parameters $\mathbf{p}_L$ and $\mathbf{p}_U$ represent lower and upper values for design variables, respectively. nn is the total number of nodes.

Table 2 shows the obtained results with the optimization methods averaged for 20 independent runs. In the same table, the efficiency parameter (ratio between total number of objective function (deterministic) and limit state function (probabilistic) evaluations to the corresponding value for the best valid solution found so far, i.e., less weight and not constrained) and the quality parameter (best objective function values ratio) for each result are presented.

Results from GA, FMA, and SQP show to be heavier then that reported by [6] using SORA/SQP method, but both PSO and HGPS (particularly HGPS) achieved a lighter truss solution. The author [8] reported that the methods based on SORA may fall into local optima points; that is, local optimum originated mainly due to the used gradient-based methods involved in optimization of the objective function and constraints. So, the best results are attributed to the use of metaheuristic capabilities in the proposed HGPS algorithm.

In addition, it is possible to note that the GA and FMA solutions were not better (objective function) than SORA solution, meaning that the reported configuration seems not to be the best for this problem. Apparently, GA and FMA get stuck into local minima. It was also observed that the PSO, SQP, and HGPS methods found slightly different design variable solutions even with reliability index constraints being satisfied and having similar objective function values. This seems to represent a flat design space near the optimum solution. It is also possible to notice the not so good HGPS efficiency when compared to PSO algorithm or SPQ (the most efficient), although HGPS have a superior solution (quality). It should the emphasized that any of the deterministic and probabilistic constraints are violated for the HGPS solution. For the 20 independent runs, the HGPS presented the best mass value of 1252.31 kg, mean value of 1262.15 kg (median 1264.97 kg), the worst mass value of 1269.68 kg, and a coefficient of variation of 0.012.

5.2. Case 2: 37 Bar Truss Problem. In this example, a Pratt type truss with 37 members is analyzed. The goal of the deterministic optimization problem is to minimize the mass. There are nonstructural masses of $m = 10$ kg attached to each of the bottom node; see Figure 5. The lower chord is modelled as bar elements with fixed cross-sectional area $A = 4 \times 10^{-3}\,\text{m}^2$. The remaining symmetric bars are design variables also modelled as bar elements with initial cross section of $A_0 = 1 \times 10^{-4}\,\text{m}^2$. This problem is considered a truss optimization on size and geometry since all nodes of the upper chord are allowed to vary along the y-axis in a symmetric way and all the diagonal and upper bars are allowed to vary the cross-sectional area within upper and lower values. Figure 5 describes the structure. In this example, the design variables are $(y_3, y_5, y_7, y_9, y_{11}, A_1, A_2, \ldots, A_{14})$. The lower limit for the design variables are $(0, 0, 0, 0, 0, 1 \times 10^{-4}, 1 \times 10^{-4}, \ldots, 1 \times 10^{-4})$ in m and m^2, respectively. The upper limit is $(5, 5, 5, 5, 5, 5 \times 10^{-4}, 5 \times 10^{-4}, \ldots, 5 \times 10^{-4})$ in m and m^2. A deterministic constraint for the 3 first natural frequencies is imposed to the original problem; see Table 3. The optimum mass found for the deterministic problem is 360.56 kg with the design variables being (0.93911 1.3327 1.5211 1.6656 1.7584 2.9941×10^{-4} 1.0019×10^{-4} 1.0033×10^{-4} 2.4837×10^{-4} 1.1804×10^{-4} 1.2643×10^{-4} 2.5903×10^{-4} 1.5983×10^{-4} 1.5209×10^{-4} 2.5021×10^{-4} 1.2443×10^{-4} 1.3579×10^{-4} 2.3928×10^{-4} 1.0014×10^{-4}) m and m^2 and the natural frequencies constraints are satisfied ($f_1 \ge 20$, $f_2 \ge 40$ and $f_3 \ge 60$).

Table 2: Results obtained using the optimization methods (10 bar truss).

Parameters	GA	FMA	PSO	SQP	SORA/SQP [6]	HGPS
Best mass value (kg)	1447.04	1667.75	1253.75	1256.10	1253.91	1252.31
β_s	3.004	3.181	3.004	3.000	3.0**	3.000
β_d	3.001	3.000	3.000	3.000	3.0**	3.000
*Mean number of objective function evaluations	9031	10005	2012	3207	4030	2454
Mean number of LSF evaluations	15630	16028	12121	4580	1775	29640
Efficiency	4.24	4.48	2.43	1.34	1.00	5.53
Quality	1.155	1.332	1.001	1.003	1.001	1.000
A_1 (m^2)	7.2529×10^{-3}	8.3642×10^{-3}	7.4611×10^{-3}	7.4563×10^{-3}	7.4580×10^{-3}	7.4782×10^{-3}
A_2 (m^2)	5.1958×10^{-3}	1.5359×10^{-3}	4.9060×10^{-3}	5.5108×10^{-3}	4.9032×10^{-3}	4.8697×10^{-3}
A_3 (m^2)	1.1237×10^{-2}	1.0675×10^{-2}	9.9456×10^{-3}	9.3988×10^{-3}	9.9483×10^{-3}	1.0026×10^{-2}
A_4 (m^2)	1.2833×10^{-3}	5.0408×10^{-3}	7.8759×10^{-6}	7.4939×10^{-6}	6.4516×10^{-6}	6.9707×10^{-6}
A_5 (m^2)	3.5329×10^{-3}	3.7096×10^{-3}	6.5592×10^{-6}	2.0610×10^{-5}	6.4516×10^{-6}	6.4516×10^{-6}
A_6 (m^2)	3.0688×10^{-3}	8.5805×10^{-3}	6.4516×10^{-6}	7.6501×10^{-6}	6.4516×10^{-6}	6.9707×10^{-6}
A_7 (m^2)	6.1501×10^{-3}	5.6797×10^{-3}	6.9485×10^{-3}	6.6513×10^{-3}	6.9548×10^{-3}	6.9358×10^{-3}
A_8 (m^2)	4.5216×10^{-3}	5.9744×10^{-3}	5.3252×10^{-3}	5.2952×10^{-3}	5.3354×10^{-3}	5.2899×10^{-3}
A_9 (m^2)	9.8911×10^{-4}	5.9404×10^{-3}	6.4516×10^{-6}	7.8343×10^{-6}	6.4516×10^{-6}	6.9707×10^{-6}
A_{10} (m^2)	6.4405×10^{-3}	2.1937×10^{-3}	6.9428×10^{-3}	7.2971×10^{-3}	6.9419×10^{-3}	6.9180×10^{-3}

*Mean number of LSF evaluations: the mean number of times the probabilistic constraints are evaluated in order to get the reliability index (20 independent runs).
**It is assumed that the constraints become active. The original paper did not mention this value.

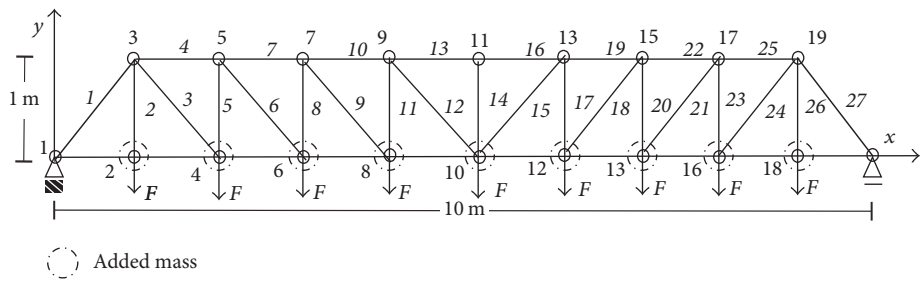


Figure 5: 37 bar truss problem.

Table 3: Structural properties and constraints for the 37 bar Pratt truss.

Parameter	Value	Unit
E (modulus of elasticity)	2.1×10^{11}	Pa
F (applied load)	50	N
ρ (material density)	7800	kg/m^3
Natural frequency constraints	$f_1 \geq 20$, $f_2 \geq 40$ and $f_3 \geq 60$	Hz
Material strength	3.0×10^8	Pa
Displacement limit	5.0×10^{-4}	m

The probabilistic problem is then adapted from this deterministic example, described in [27, 28], following the same geometry and deterministic frequency constraint. Displacement and stress constraints as well as concentrated loads are added to the original problem in order to result in extra probabilistic constraints for stress and displacements.

In this problem, the first three natural frequencies (f_1, f_2 and f_3) are of interest and a reliability constraint is considered in the probabilistic problem. Therefore, the goal of the optimization problem is to minimize the mass of the truss taking into account constraints for stress, displacement, and natural frequency reliability indexes. All remaining symmetric member areas are considered random variables with lognormal distribution with coefficient of variation of 0.01. Therefore, in this problem, the mean value of cross-sectional areas is either of the design variables for the optimization problem as random variables for the probabilistic problem as well. The modulus of elasticity is considered following a normal distribution with mean 2.1×10^{11} Pa and coefficient of variation of 0.05 (as reported in [29]). The areas of the lower chord bars are equal to 4×10^{-3} m^2 and assumed deterministic and they are not design variables. The physical properties and the constraints are shown in Table 3. The violation of any of these constraints will represent a failure mode. In this particular model, as previously described, three reliability

Table 4: Final objective function, constraints, and parameters by optimization method (objective function values do not take into account added masses).

Parameters	GA	FMA	DE	PSO	HGPS
Best mass value (kg)	448.85	413.29	394.75	397.10	374.187
β_f	3.084	3.803	3.65	3.038	3.275
β_s	10.00	10.00	10.0	10.00	10.00
β_d	10.00	10.00	10.0	10.00	10.00
Mean number of objective function evaluations	1982	3084	10007	4003	1839
*Mean number of LSF evaluations	767715	392793	1443497	591936	805459
Efficiency	1.94	1.0	3.67	1.51	2.00
Quality	1.20	1.10	1.05	1.06	1.00
Maximum absolute member stress (Pa)	7.79×10^5	1.26×10^6	1.11×10^6	2.11×10^6	1.13×10^6
Maximum absolute node displacement (m)	6.09×10^{-5}	6.32×10^{-5}	5.37×10^{-5}	6.79×10^{-5}	6.58×10^{-5}
f_1 (Hz)	21.76	21.01	23.27	21.78	21.85
f_2 (Hz)	53.64	46.53	44.33	48.84	43.82
f_3 (Hz)	70.69	66.99	66.30	64.93	65.63

*Mean number of LSF evaluations: the mean number of times the probabilistic constraints are evaluated in order to get the reliability index (20 independent runs).

indexes are considered as extra constraints for the problem: β_f, β_s, and β_d. They represent the reliability for frequency, stress, and displacement constraints.

Equation (21), in this case, can represent the RBDO mathematical model, where $\beta_{Lf} = 3.0$, $\beta_{Ls} = 2.0$, and $\beta_{Ld} = 2.0$ represent the target reliability indexes for frequency, stress, and displacement constraints:

$$\begin{aligned}
&\text{Minimize} \quad f(\mathbf{A}) = \rho \sum_{i=1}^{n} A_i L_i \\
&\text{Subject to} \quad f_1 \geq 20\,\text{Hz}, \\
&\qquad f_2 \geq 40\,\text{Hz}, \\
&\qquad f_3 \geq 60\,\text{Hz} \\
&\qquad |\sigma_i| \leq 3.0 \times 10^8\,\text{Pa} \quad i = 1, \ldots, n \\
&\qquad |d_j| \leq 5.0 \times 10^{-4}\,\text{m} \quad j = 1, \ldots, nn \\
&\qquad \beta_f(\mathbf{u}, \mathbf{p}) \geq \beta_{Lf} \\
&\qquad \beta_s(\mathbf{u}, \mathbf{p}) \geq \beta_{Ls} \\
&\qquad \beta_d(\mathbf{u}, \mathbf{p}) \geq \beta_{Ld} \\
&\qquad \mathbf{p}_L \leq \mathbf{p} \leq \mathbf{p}_U,
\end{aligned} \tag{21}$$

where $\mathbf{p}$ represents the vector of design variables (five y coordinates and 14 member areas) and $\mathbf{u}$ represents the random variable vector (14 member areas and the Young modulus). nn is the total number of nodes.

Table 4 presents a summary of the results and Table 5 shows the obtained values for each design variable for the tested optimization algorithms. For the 20 independent runs, the HGPS presented the best mass value of 374.187 kg, mean value of 396.680 kg (median 398.13 kg), worst mass value of 405.097 kg, and a coefficient of variation of 0.037.

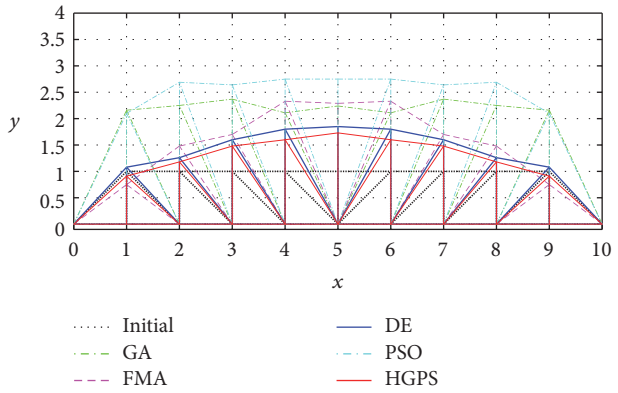


Figure 6: Superimposed geometries obtained with the optimization methods.

Figure 6 presents the resulting superimposed geometric configurations obtained with the optimization methods. HGPS in this case give the best mass value (quality) although with not so good efficiency as the PSO algorithm. The reliability for displacements and stress constraints resulted in high values ($\beta = 10$ means negligible failure of probability), indicating that the frequency constraint, in this case, is a dominant mode of failure that prevails over the two other constraints. As expected, the absolute value for displacement, stress, and frequencies in the optimum designs are far below the corresponding limits.

5.3. Case 3: 25 Bar Space Truss Problem. In this example, the mass minimization of a 25 bar truss is performed, subject to constraints in the reliability indexes for stress, displacement, and first natural frequency. The deterministic optimization problem was previously presented by [30] where the mass minimization was performed not taking into account uncertainties; see Figure 7. The problem is described as a discrete

Table 5: Design variables for each optimization method.

Variable	GA	FMA	DE	PSO	HGPS
Y_3, Y_{19} (m)	2.16	0.75	1.08	2.10	0.91
Y_5, Y_{17} (m)	2.25	1.48	1.26	2.69	1.18
Y_7, Y_{15} (m)	2.37	1.70	1.60	2.64	1.48
Y_9, Y_{13} (m)	2.11	2.33	1.80	2.75	1.60
Y_{11} (m)	2.24	2.29	1.85	2.75	1.73
A_1, A_{27} (m^2)	3.34×10^{-4}	3.08×10^{-4}	4.20×10^{-4}	2.78×10^{-4}	3.34×10^{-4}
A_2, A_{26} (m^2)	4.24×10^{-4}	2.98×10^{-4}	1.00×10^{-4}	2.84×10^{-4}	1.30×10^{-4}
A_3, A_{24} (m^2)	4.33×10^{-4}	1.52×10^{-4}	1.66×10^{-4}	1.23×10^{-4}	1.18×10^{-4}
A_4, A_{25} (m^2)	3.17×10^{-4}	3.37×10^{-4}	2.88×10^{-4}	2.77×10^{-4}	3.83×10^{-4}
A_5, A_{23} (m^2)	3.45×10^{-4}	2.57×10^{-4}	1.00×10^{-4}	2.07×10^{-4}	1.44×10^{-4}
A_6, A_{21} (m^2)	1.39×10^{-4}	1.10×10^{-4}	1.00×10^{-4}	2.38×10^{-4}	1.24×10^{-4}
A_7, A_{22} (m^2)	3.44×10^{-4}	3.66×10^{-4}	5.00×10^{-4}	2.75×10^{-4}	3.25×10^{-4}
A_8, A_{20} (m^2)	1.84×10^{-4}	3.09×10^{-4}	4.36×10^{-4}	2.89×10^{-4}	1.75×10^{-4}
A_9, A_{18} (m^2)	3.94×10^{-4}	4.01×10^{-4}	2.99×10^{-4}	2.70×10^{-4}	1.88×10^{-4}
A_{10}, A_{19} (m^2)	4.76×10^{-4}	2.96×10^{-4}	3.92×10^{-4}	2.95×10^{-4}	4.56×10^{-4}
A_{11}, A_{17} (m^2)	4.41×10^{-4}	4.48×10^{-4}	1.00×10^{-4}	2.41×10^{-4}	2.11×10^{-4}
A_{12}, A_{15} (m^2)	2.71×10^{-4}	3.57×10^{-4}	3.80×10^{-4}	2.59×10^{-4}	1.58×10^{-4}
A_{13}, A_{16} (m^2)	4.08×10^{-4}	2.16×10^{-4}	5.00×10^{-4}	2.78×10^{-4}	3.63×10^{-4}
A_{14} (m^2)	2.64×10^{-4}	4.20×10^{-4}	1.02×10^{-4}	2.63×10^{-4}	1.00×10^{-4}

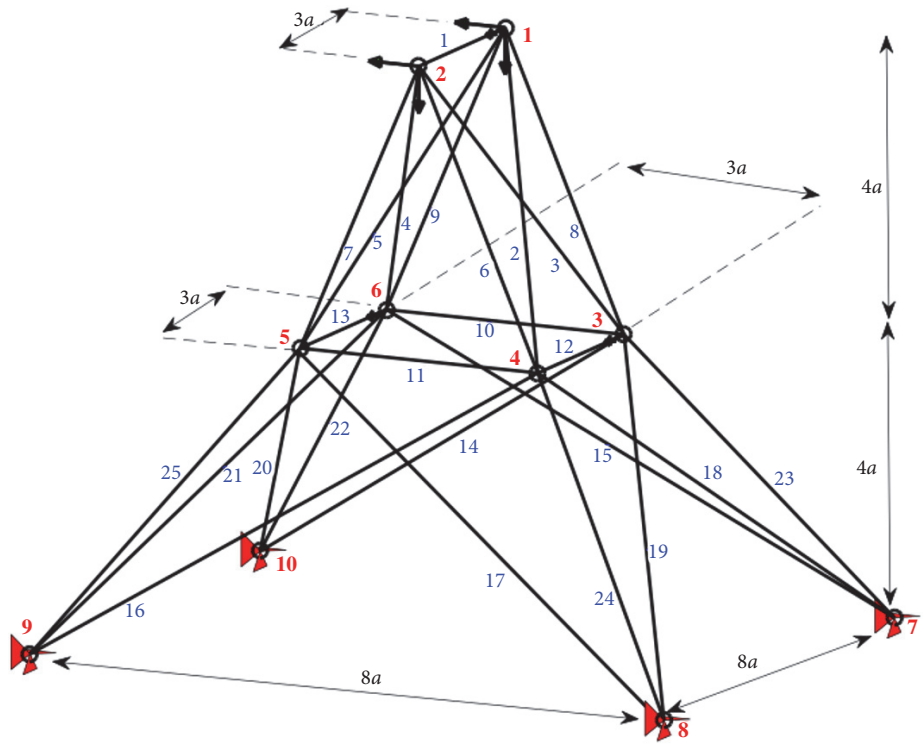


Figure 7: 25 bar space truss. Dimensions, numbering, and boundary conditions.

optimization problem since the member areas can vary following a table of discrete values that ranges from 6.4516×10^{-5} m^2 (0.1 in^2) to 2.258×10^{-3} m^2 (3.5 in^2) by increments of 6.4516×10^{-5} m^2 (0.1 in^2). There are stress constraints in all members such that a strength limit of $\pm 2.7579 \times 10^8$ Pa (± 40 ksi) should be verified. A displacement constraint of $\pm 8.89 \times 10^{-3}$ m (± 0.35 in) is imposed to nodes 1 and 2 in x and y direction. The fundamental frequency should be constrained to $f_1 \geq 55$ Hz. Figure 7 shows the dimensions of the 25 bar spatial truss example. The physical properties of

the material, upper and lower limits for the design variables, and the deterministic constraints are listed in Table 6.

In this example, eight sets of grouped bars are assumed, resulting in eight design variables, namely, set 1, nodes 1-2; set 2, nodes 1–4, 2-3, 1–5, and 2–6; set 3, nodes 2–5, 2–4, 1–3, and 1–6; set 4, nodes 3–6 and 4-5; set 5, nodes 3-4 and 5-6; set 6, nodes 3–10, 6-7, 4–9, and 5–8; set 7, nodes 3–8, 4–7, 6–9, and 5–10; set 8, nodes 3–7, 4–8, 5–9, and 6–10. Table 7 presents the load direction and magnitude. The spatial truss is fixed at the lower nodes and the loads are applied as indicated in Figure 7 (where dimension $a = 9.144$ m). The obtained final mass by the deterministic optimization is 217.60 kg (479.72 lbm) with the optimum design variable vector (discrete areas) as (6.4516×10^{-5} 1.9354×10^{-4} 2.2580×10^{-3} 6.4516×10^{-5} 1.0322×10^{-3} 5.8064×10^{-4} 3.2258×10^{-4} 2.2580×10^{-3}) m^2.

For the RBDO problem, the areas are either design variables (deterministic) or random (probabilistic). They are assumed uncorrelated, following a lognormal distribution with coefficient of variation of 0.02; the elastic modulus follows a Gaussian distribution with coefficient of variation of 0.05. Thus, there are, in total, nine random variables. The reliability index levels for each constraint (frequency, stress, and displacement) are set to $\beta_{Lf} \geq 1.5$, $\beta_{Ls} \geq 2.0$, and $\beta_{Ld} \geq 1.5$. Therefore, the reliability-based optimization problem is written as indicated by

$$\begin{aligned}
&\text{Minimize} \quad f(\mathbf{A}) = \rho \sum_{i=1}^{n} A_i L_i \\
&\text{Subject to} \quad f_1 \geq 55\,\text{Hz} \\
&\qquad |\sigma_i| \leq 2.7579 \times 10^8\,\text{Pa} \quad i = 1, \ldots, n \\
&\qquad |d_j| \leq 8.89 \times 10^{-3}\,\text{m} \\
&\qquad\qquad j = 1, 2 \text{ in } x \text{ and } y \text{ direction} \\
&\qquad \beta_f(\mathbf{u}, \mathbf{p}) \geq \beta_{Lf} \\
&\qquad \beta_s(\mathbf{u}, \mathbf{p}) \geq \beta_{Ls} \\
&\qquad \beta_d(\mathbf{u}, \mathbf{p}) \geq \beta_{Ld} \\
&\qquad \mathbf{p}_L \leq \mathbf{p} \leq \mathbf{p}_U,
\end{aligned} \tag{22}$$

where $\mathbf{p}$ represents the vector of design variables (eight discrete member areas) and $\mathbf{u}$ represents the random variable vector (8 discrete member areas and the elastic modulus). $\mathbf{p}_L$ and $\mathbf{p}_U$ are the lower and upper limits for the design variables. nn is the total number of nodes.

Table 8 presents a summary of the results and Table 9 shows the obtained design variables for each optimization method. For the 20 independent runs, the HGPS presented the best mass value of 253.80 kg, mean value of 255.76 kg (median 254.01 kg), the worst mass value of 265.73 kg, and a coefficient of variation of 0.014. Particularly in this example, PSO and HGPS found the same close results (best quality indexes), although HGPS resulted in a better efficiency index. All reliability constraints were satisfied and the displacement constraint presented as the most important and dominant mode of failure. Stress and frequency constraints resulted in high reliability indexes (negligible probability of failure).

Table 6: Structural properties and constraints for the 25 bar space truss.

Properties	Values
E (Young's modulus)	6.895×10^{10} Pa (1×10^4 ksi)
ρ (density)	2767.99 kg/m^3 (0.1 lbm/in^3)
Area lower limit	6.4516×10^{-5} m^2 (0.1 in^2)
Area upper limit	2.258×10^{-3} m^2 (3.5 in^2)
Material Strength	2.7579×10^8 Pa (40 ksi)
Displacement limit	8.89×10^{-3} m (0.35 in)
1st natural frequency constraint	$f_1 \geq 55$ Hz

Table 7: Applied loads.

Node	Load component		
	F_x	F_y	F_z
1	4448.22 N (1000 lbf)	−44482.22 N (−10000 lbf)	−44482.22 N (−10000 lbf)
2	0.0 N	−44482.22 N (−10000 lbf)	−44482.22 N (−10000 lbf)
3	2224.11 N (500 lbf)	0.0 N	0.0 N
6	2668.93 N (600 lbf)	0.0 N	0.0 N

6. Final Remarks and Conclusions

In this article, a hybridized method for RBDO is proposed. It takes advantage of the main features presented in some heuristic algorithms like PSO, GA, and gradient-based methods, like SQP. Simple to gradually more complex truss examples are analyzed using the proposed methodology, compared with literature examples. The best result is obtained with the hybrid method HGPS in the analyzed examples, albeit in the last example PSO and HGPS, presented the same result. Moreover, it is relevant to mention that, due to complexity of the optimization problems with uncertainties, in most of situations, the heuristic methods solely were not able to find the optimal design, although they resulted in objective function with same order of magnitude. This suggests that a different convergence criterion may be used for each algorithm in order to avoid premature convergence.

In all problems, the resulting final mass obtained using RBDO is larger than that obtained by deterministic optimization; however, the corresponding reliability levels in stress, displacements, or frequency have been met. These reliability levels are not attained with the original deterministic optimization.

Even with a large number of design variables in the optimization analysis, all presented global search methods were able to find feasible solutions. The optimization methods found different optimum design variables for similar objective function. It can be argued that those design points do not represent the best quality converged design points and may

Table 8: Results for the optimization methods in 25 bar truss example.

Parameters	GA	FMA	DE	PSO	HGPS
Best mass value (kg)	272.69	261.38	254.45	253.80	253.80
β_f	10.000	9.031	9.014	9.803	9.803
β_s	10.000	10.000	10.000	10.000	10.000
β_d	1.973	1.996	2.021	1.963	1.963
*Mean number of objective function evaluations	3452	2002	5012	3002	1002
*Mean number of LSF evaluations	499001	281583	185877	333634	74143
Efficiency	6.69	3.77	2.54	4.48	1.00
Quality	1.074	1.029	1.002	1.000	1.000
Maximum absolute stress (Pa)	5.2752×10^7	3.504×10^7	4.1136×10^7	3.904×10^7	3.904×10^7
Maximum absolute displacement (m)	7.992×10^{-3}	7.998×10^{-3}	8.882×10^{-3}	7.998×10^{-3}	7.998×10^{-3}
f_1 (Hz)	75.42	77.38	59.68	73.81	73.81

*Mean number of LSF evaluations: the mean number of times the probabilistic constraints are evaluated in order to get the reliability index (20 independent runs).

Table 9: Best design variables for tested optimization methods.

Design variable	GA	FMA	DE	PSO	HGPS
A_1 (m^2)	6.45×10^{-5}	1.93×10^{-4}	6.45×10^{-5}	6.45×10^{-5}	6.45×10^{-5}
$A_{2\text{ to }5}$ (m^2)	1.29×10^{-3}	7.74×10^{-4}	6.45×10^{-4}	5.16×10^{-4}	5.16×10^{-4}
$A_{6\text{ to }9}$ (m^2)	1.80×10^{-3}	2.06×10^{-3}	2.25×10^{-3}	2.25×10^{-3}	2.25×10^{-3}
$A_{10\text{ and }11}$ (m^2)	6.45×10^{-4}	1.29×10^{-4}	6.45×10^{-5}	6.45×10^{-5}	6.45×10^{-5}
$A_{12\text{ and }13}$ (m^2)	2.58×10^{-4}	3.87×10^{-4}	7.74×10^{-4}	1.16×10^{-3}	1.16×10^{-3}
$A_{14\text{ and }17}$ (m^2)	9.03×10^{-4}	7.09×10^{-4}	7.09×10^{-4}	7.09×10^{-4}	7.09×10^{-4}
$A_{18\text{ and }21}$ (m^2)	6.45×10^{-4}	9.03×10^{-4}	6.45×10^{-4}	5.80×10^{-4}	5.80×10^{-4}
$A_{22\text{ and }25}$ (m^2)	2.19×10^{-3}	2.19×10^{-3}	2.25×10^{-3}	2.26×10^{-3}	2.26×10^{-3}

represent a local minimum; however, for nonexplicit limit state function problems, this can only be guaranteed checking the results with different algorithms.

The hybrid method HGPS, when compared to the GA in the conventional form, presented a lower mass value (quality in result) with a low number of function evaluations (efficiency in the optimization process). The better efficiency was attained with the gradient- based methods at the expense of worse quality.

The proposed final problem, which cannot be handled by single loop RBDO algorithms (design variables are also random variables), highlights the importance of the proposed approach in cases where the discrete design variables are also random variables. Some new improvements in the HGPS method are being planned in future research like the parallelization of the code. More challenging problems are foreseen to test the proposed method.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors acknowledge Brazilian Agencies CNPq and CAPES for the partial financial support for this research.

References

[1] H. Agarwal, *Reliability Based Design Optimization: Formulation and Methodologies*, Aerospace and Mechanical Engineering. University of Notre Dame, formulation and methodologies, 2004.

[2] A. T. Beck and C. C. Verzenhassi, "Risk optimization of a steel frame communications tower subject to tornado winds," *Latin American Journal of Solids and Structures*, vol. 5, no. 3, pp. 187–203, 2008.

[3] S. Shan and G. G. Wang, "Reliable design space and complete single-loop reliability-based design optimization," *Reliability Engineering System Safety*, vol. 93, no. 8, pp. 1218–1230, 2008.

[4] F. Li, T. Wu, A. Badiru, M. Hu, and S. Soni, "A single-loop deterministic method for reliability-based design optimization," *Engineering Optimization*, vol. 45, no. 4, pp. 435–458, 2013.

[5] X. Du and W. Chen, "Sequential optimization and reliability assessment method for efficient probabilistic design," *Journal of Mechanical Design*, vol. 126, no. 2, pp. 225–233, 2004.

[6] P. Yi, G. Cheng, and L. Jiang, "A sequential approximate programming strategy for performance-measure-based probabilistic structural design optimization," *Structural Safety*, vol. 30, no. 2, pp. 91–109, 2008.

[7] R. H. Lopez and A. T. Beck, "Reliability-based design optimization strategies based on FORM: a review," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 34, no. 4, pp. 506–514, 2012.

[8] H. M. Gomes, *Reliability analysis of reinforced concrete structures [Ph.D. Thesis]*, Brazil, Porto Alegre, 2001.

[9] X. Zhuang, R. Pan, and X. Du, "Enhancing product robustness in reliability-based design optimization," *Reliability Engineering & System Safety*, vol. 138, pp. 145–153, 2015.

[10] H. Bae, R. V. Grandhi, and R. A. Canfield, "Structural design optimization based on reliability analysis using evidence theory," SAE Technical Paper 2003-01-0877, 2007.

[11] D. M. Frangopol, "Probability concepts in engineering: emphasis on applications to civil and environmental engineering," *Structure and Infrastructure Engineering*, vol. 4, no. 5, pp. 413-414, 2008.

[12] I. Gondres Torné, R. Báez Prieto, S. Lajes Choy, and A. del Castillo Serpa, "Determinación de la confiabilidad en interruptores de potencia: caso de estudio," *Ingeniare. Revista Chilena de Ingeniería*, vol. 21, no. 2, pp. 271–278, 2013.

[13] A. M. Hasofer and D. Lind, "An exact and invariant first-order reliability format," *Journal of Engineering Mechanics. ASCE*, vol. 100, pp. 111–121, 1974.

[14] A. Haldar and S. Mahadevan, "Probability, reliability and statistical methods in engineering design," in *Engineering Design*, J. W. Sons, Ed., vol. 77, p. 379, 2000.

[15] R. E. Melchers, *Structural Reliability Analysis and Prediction*, C. John Wiley & Sons Ltd, New York, NY, USA, 2nd edition, 1999.

[16] A. D. Belegundu and T. R. Chandrupatla, *Optimization Concepts and Applications in Engineering*, Cambridge University Press, UK, 2nd edition, 1999.

[17] J. Tu, K. K. Choi, and Y. H. Park, "A new study on reliability-based design optimization," *Journal of Mechanical Design*, vol. 121, no. 4, pp. 557–564, 1999.

[18] V. Kelner, F. Capitanescu, O. Léonard, and L. Wehenkel, "A hybrid optimization technique coupling an evolutionary and a local search algorithm," *Journal of Computational and Applied Mathematics*, vol. 215, no. 2, pp. 448–456, 2008.

[19] I. E. Grossmann and L. T. Biegler, "Part II. Future perspective on optimization," *Computers and Chemical Engineering*, vol. 28, no. 8, pp. 1193–1218, 2004.

[20] H. Adeli and N.-T. Cheng, "Augmented lagrangian genetic algorithm for structural optimization," *Journal of Aerospace Engineering*, vol. 7, no. 1, pp. 104–118, 1994.

[21] F. van den Bergh and A. P. Engelbrecht, "A study of particle swarm optimization particle trajectories," *Information Sciences*, vol. 176, no. 8, pp. 937–971, 2006.

[22] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 4, pp. 656–667, 1994.

[23] S. Marsili Libelli and P. Alba, "Adaptive mutation in genetic algorithms," *Soft Computing*, vol. 4, no. 2, pp. 76–80, 2000.

[24] R. Hassan, B. Cohanim, O. Weck, and G. Venter, "A comparison of particle swarm optimization and the genetic algorithm," in *Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics Material Conference*, Austin, Tex, USA, 2005.

[25] L. J. Li, Z. B. Huang, F. Liu, and Q. H. Wu, "A heuristic particle swarm optimizer for optimization of pin connected structures," *Computers and Structures*, vol. 85, no. 7-8, pp. 340–349, 2007.

[26] R. Riolo, U.-M. O'Reilly, and T. McConaghy, "genetic programming theory and practice VII," in *Genetic & Evolutionary Computation*, Series, Ed., Springer, 2010.

[27] D. Wang, W. H. Zhang, and J. S. Jiang, "Truss optimization on shape and sizing with frequency constraints," *AIAA Journal*, vol. 42, no. 3, pp. 622–630, 2004.

[28] W. Lingyun, Z. Mei, W. Guangming, and M. Guang, "Truss optimization on shape and sizing with frequency constraints based on genetic algorithm," *Computational Mechanics*, vol. 35, no. 5, pp. 361–368, 2005.

[29] G. D. Cheng, L. Xu, and L. Jiang, "A sequential approximate programming strategy for reliability-based structural optimization," *Computers and Structures*, vol. 84, no. 21, pp. 1353–1367, 2006.

[30] W. H. Tong and G. R. Liu, "Optimization procedure for truss structures with discrete design variables and dynamic constraints," *Computers and Structures*, vol. 79, no. 2, pp. 155–162, 2001.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 6710929, 15 pages
https://doi.org/10.1155/2017/6710929

Research Article

Solving a Two-Stage Stochastic Capacitated Location-Allocation Problem with an Improved PSO in Emergency Logistics

Ye Deng, Wanhong Zhu, Jian Tang, and Jianfei Qin

College of Field Engineering, The PLA University of Science and Technology, Nanjing 210000, China

Correspondence should be addressed to Wanhong Zhu; zhuwanhong2016@gmail.com

Received 12 November 2016; Revised 12 March 2017; Accepted 23 March 2017; Published 31 May 2017

Academic Editor: Jorge Magalhaes-Mendes

A stochastic expected value model and its deterministic conversion are developed to formulate a two-stage stochastic capacitated location-allocation (LA) problem in emergency logistics; that is, the number and capacities of supply centers are both decision variables. To solve these models, an improved particle swarm optimization algorithm with the Gaussian cloud operator, the Restart strategy, and the adaptive parameter strategy is developed. The algorithm is integrated with the interior point method to solve the second-stage model. The numerical example proves the effectiveness and efficiency of the conversion method for the stochastic model and the proposed strategies that improve the algorithm.

1. Introduction

A location-allocation (LA) problem, also known as facility location problem (FLP), involves locating a number of facilities to which customers are allocated to minimize the cost of satisfying customer demands. It is an important problem in supply chain or logistics management and greatly affects long-term transportation and storage decisions. Many enterprises and government departments focus on this problem to reduce cost and to improve efficiency, especially in emergency logistics given the frequent occurrence of disasters, epidemic, security incidents, and other emergencies nowadays. This problem has complex and uncertain features, such as the changing demands, allocations, and locations of customers or facilities.

Many studies have been conducted on the basic LA problem since the proposal of Cooper [1] in 1963. However, most of the studies were for deterministic cases and not for uncertain occurrences. In the recent two decades, several models for uncertain occurrences have been proposed and solved by different algorithms. Logendran and Terrell [2] first considered a stochastic uncapacitated LA problem and proposed an expected value model (EVM) to maximize the net profits. Carrizosa et al. [3, 4] proposed a LA problem that considers the locations of both customers and facilities, which may be regions that have several probability distributions. Liu [5, 6] contributed to the uncertainty theory by proposing three stochastic models [7] and three fuzzy programing models [8] for the capacitated LA problem. A hybrid intelligent algorithm that consists of a network simplex algorithm, a simulation, and a genetic algorithm was developed to solve the stochastic and fuzzy models above. Silva and De La Figuera [9] studied the capacitated facility location problem with constrained backlogging probabilities and solved it using a heuristic method based on a reactive greedy adaptive search procedure. Wang and Shi-Wei [10] proposed a robust optimization model for a logistics center LA problem and compared it with stochastic and deterministic optimization models. Two algorithms, namely, an enumeration method and a genetic algorithm, were adopted to solve the problem. Yao et al. [11] considered a joint facility LA and inventory problem with stochastic demands. The problem involves identifying the best locations of warehouses and the inventory levels and allocating customers. A heuristic integrated approximation and transformation technique was developed to solve the problem. Wen and Kang [12] considered a facility LA problem with random fuzzy demands. They proposed a hybrid intelligent algorithm, similar to the method of Zhou and Liu [8], which consists of the simplex algorithm, a random fuzzy simulation, and a genetic algorithm. A similar

method was also adopted by Mousavi and Niaki [13] in solving a LA problem with a fuzzy variable and customer location demands, which were normal random variables. Vidyarthi and Jayaswal [14] proposed a nonlinear integer programing model to solve a LA problem with immobile servers, stochastic demands, and congestions. Pereira et al. [15] presented a probabilistic maximal covering LA problem and proposed a hybrid algorithm to solve it. They formulated a linear programing model to efficiently solve small and medium problems and a flexible adaptive large neighborhood search heuristic to solve large problems. Alizadeh et al. [16] considered a capacitated multifacility LA problem with stochastic demands. The capacitated subsources of each facility could be utilized when the number of demand points, that is, the planned total requirements, are exceeded. Alizadeh et al. [17] transformed the mixed-integer nonlinear programing model to a simple formulation model and proposed a genetic algorithm (GA) and a colonial competitive algorithm (CCA) to solve medium and large problems.

In this study, we consider a two-stage stochastic capacitated LA problem (SCLAP) in the context of emergency logistics. In the management of emergency logistics, the core problem is utilizing the limited relief supplies rapidly and efficiently. Hence, predesigned supply centers, that is, emergency logistics distribution centers, are important. The "appropriate" number, size, and location of supply centers have become a comprehensive decision problem that should not be addressed separately. The demands of different customers are mostly uncertain and depend largely on different scenarios. Hence, we focus on the uncertainty in demand quantity, which is assumed as the only independent stochastic variable in this paper that follows a given regular stochastic distribution. To solve SCLAP two conditions must be satisfied: the constraint of stochastic quantity demands and the minimization of the generalized cost. The generalized cost consists of two parts, namely, the sum of the costs of building and maintaining supply centers and the stochastic costs of transportation to each customer from each supply center. Therefore, we determine the appropriate number, capacities, and locations of supply centers in this paper.

In a traditional capacitated LA problem (CLAP), each customer can be supplied by existing supply centers and can be supplied by more than one center at the same time. Hence, the problem becomes NP-hard and difficult to solve [7]. Moreover, in the two-stage SCLAP, the number, the capacities, and the locations of supply centers are considered decision variables. The research on this model is extremely weak, and few relevant papers have been found [18–23]. Furthermore, most of the research considered the number and the capacities of supply centers separately, and no research that considered both variables together has been found.

To efficiently solve this model, an improved particle swarm optimization (PSO) algorithm is proposed. The algorithm consists of three improvement strategies: the Gaussian cloud operator, the Restart strategy, and the adaptive parameter strategy. The second stage of the problem is modeled as a linear program. Hence, we adopt the interior point method of the time-consuming simplex method. We convert the initial stochastic programing model to a crisp model, thus reducing the computing time dramatically based on the assumption of the demands with independent regular distributions and the uncertainty theory proposed by Liu [6].

The remainder of this paper is organized as follows. Section 2 describes the random EVM and the crisp model for the two-stage SCLAP. Section 3 presents the details of the hybrid algorithm solution to the model. Section 4 introduces a case study of the new model and verifies the algorithm efficiency with the improvement strategies. Section 5 concludes with the contributions and innovations of this paper and presents the future research directions.

2. Model Formulation

2.1. Problem Description and Theoretical Foundation. To model the two-stage SCLAP, the following assumptions should be considered: the graph of all the nodes is a complete graph, each customer node can be connected with all supply nodes but cannot be connected with another customer node, the weight of each edge between two nodes is measured by the Euclidian distance plus the transportation volume, the locations of customer nodes are fixed and the demand quantities are stochastic, and the capacity constraint is only imposed on supply nodes. The notation and variables used in the following formulations are defined in Descriptions of Notations and Variables.

To model the two-stage SCLAP, we first apply the EVM introduced by Zhou and Liu [7] to the SCLAP. Then, we extend the classic one-stage EVM to a two-stage model and provide a deterministic equivalent form. We introduce several basic definitions, theorems of probability, and uncertainty theories.

Definition 1 (see [5]). Let Ω be a nonempty set and $\mathscr{A}$ the σ-algebra of the subsets (called events) of Ω. The set function Pr is called a probability measure if it satisfies the following conditions.

Axiom 1 (normality).

$$\Pr\{\Omega\} = 1. \tag{1}$$

Axiom 2 (nonnegativity).

$$\Pr\{A\} \geq 0 \quad \text{for any } A \in \mathscr{A}. \tag{2}$$

Axiom 3 (countable additivity). For every countable sequence of mutually disjoint events $\{A_i\}$, we obtain

$$\Pr\left\{\bigcup_{i=1}^{\infty} A_i\right\} = \sum_{i=1}^{\infty} \Pr\{A_i\}. \tag{3}$$

Definition 2 (see [5]). Let Ω be a nonempty set, $\mathscr{A}$ the σ-algebra of the subsets of Ω, and Pr the probability measure. Then, the triplet $(\Omega, \mathscr{A}, \Pr)$ is called a probability space.

Definition 3 (see [5]). A random variable is a measurable function from the probability space $(\Omega, \mathscr{A}, \Pr)$ to the set of

real numbers; that is, for any Borel set B of real numbers, the set is

$$\{\xi \in B\} = \{\omega \in \Omega \mid \xi(\omega) \in B\}. \tag{4}$$

Liu [6] also proposed the definitions for *measure inversion theorem*, *regular uncertainty distributions,* and *inverse uncertainty distribution* as supplements to the uncertainty theory. The uncertainty distributions are specifically presented as stochastic distributions, and the uncertainty measure $\mathcal{M}$ can also be replaced by the probability measure Pr. He proposed several theorems that can help in the conversion of the stochastic model into a crisp model based on these definitions. The related definitions and theorems are listed as follows.

Definition 4 (see [6] measure inversion theorem). Let ξ be an uncertain variable with an uncertainty distribution Φ. Then, for any real number x, we have

$$\begin{aligned} \mathcal{M}\{\xi \le x\} &= \Phi(x), \\ \mathcal{M}\{\xi > x\} &= 1 - \Phi(x). \end{aligned} \tag{5}$$

Definition 5 (see [6] regular uncertainty distribution). An uncertainty distribution $\Phi(x)$ is regular if it is a continuous and strictly increasing function with respect to x at which $0 < \Phi(x) < 1$ and

$$\begin{aligned} \lim_{x \to -\infty} \Phi(x) &= 0, \\ \lim_{x \to +\infty} \Phi(x) &= 1. \end{aligned} \tag{6}$$

Definition 6 (see [6] inverse uncertainty distribution). Let ξ be an uncertain variable with a regular uncertainty distribution $\Phi(x)$. Then, the inverse function $\Phi^{-1}(x)$ is the inverse uncertainty distribution of ξ.

Theorem 7 (see [6]). *Let $\xi_1, \xi_2, \ldots, \xi_n$ be independent uncertain variables with regular uncertainty distributions $\Phi_1, \Phi_2, \ldots, \Phi_n$, respectively. If f is a strictly increasing function, then $\xi = f(\xi_1, \xi_2, \ldots, \xi_n)$ has an inverse uncertainty distribution*

$$\Psi^{-1}(\alpha) = f\left(\Phi_1^{-1}(\alpha), \Phi_2^{-1}(\alpha), \ldots, \Phi_n^{-1}(\alpha)\right). \tag{7}$$

Theorem 8 (see [6]). *Let ξ_1 and ξ_2 be independent normal uncertain variables $\mathcal{N}(e_1, \sigma_1)$ and $\mathcal{N}(e_2, \sigma_2)$, respectively. Then, the sum of $\xi_1 + \xi_2$ is also a normal uncertain variable $\mathcal{N}(e_1 + e_2, \sigma_1 + \sigma_2)$; that is,*

$$\mathcal{N}(e_1, \sigma_1) + \mathcal{N}(e_2, \sigma_2) = \mathcal{N}(e_1 + e_2, \sigma_1 + \sigma_2). \tag{8}$$

2.2. Initial Expected Value Model. The mathematical formulation, which is an expected value model (EVM), proposed by Zhou and Liu for the initial SCLAP can be defined as follows [7]:

$$F_s = \min_{\mathbf{x},\mathbf{y}} E\left(\min_{z \in Z(\omega)} \sum_{i=1}^{n} \sum_{j=1}^{m} z_{ij} \sqrt{\left(x_i - a_j\right)^2 + \left(y_i - b_j\right)^2}\right), \tag{9}$$

which is subject to

$$\sum_{i=1}^{n} z_{ij} = \xi_j(\omega), \quad \forall j \in M,\ \omega \in \Omega; \tag{10}$$

$$\sum_{j=1}^{m} z_{ij} \le s_i, \quad \forall i \in N; \tag{11}$$

$$z_{ij} \ge 0, \quad \forall i \in N,\ j \in M; \tag{12}$$

$$(x_i, y_i) \in \mathbf{R}^2, \quad \forall i \in N; \tag{13}$$

$$(a_j, b_j) \in \mathbf{R}^2, \quad \forall j \in M, \tag{14}$$

where $E(X(\omega))$ denotes the expected value of uncertain variable X. If

$$C(\mathbf{x}, \mathbf{y} \mid \omega) = \min_{z \in Z(\omega)} \sum_{i=1}^{n} \sum_{j=1}^{m} z_{ij} \sqrt{\left(x_i - a_j\right)^2 + \left(y_i - b_j\right)^2}, \tag{15}$$

(9) will be reformulated as follows:

$$F_s = \min_{\mathbf{x},\mathbf{y}} \int_0^{\infty} \Pr\{\omega \in \Omega \mid C(\mathbf{x}, \mathbf{y} \mid \omega) \ge r\}\, dr. \tag{16}$$

In this mathematical model, the decision variable is the location of the supply centers $(\mathbf{x}, \mathbf{y})$, and the objective function is the minimum value of the sum of the demand and distance from each supply center to the customer in (9) or (16). The first constraint (see (10)) states that a customer's stochastic demand must be satisfied. The second constraint (see (11)) states that the capacity of each supply center must be sufficient to supply all customers with demands. Equations (12), (13), and (14) are used to keep the variables within a certain range.

According to Zhou and Liu [7], the EVM can be easily solved by a hybrid intelligent algorithm that consists of the network simplex algorithm, stochastic simulations, and the GA. However, in emergency logistics management, in addition to making a decision on supply center locations, the number and capacities of supply centers should also be considered. Therefore, the SCLAP should be extended into a two-stage model. Variables n and s_i are both fixed in the initial EVM, whereas, in the two-stage EVM, these variables are decision variables.

2.3. Two-Stage Expected Value Model. In the two-stage EVM for SCLAP, the decision variables extend to the number p, the capacities $\mathbf{S}$, and the locations of supply centers $(\mathbf{x}, \mathbf{y})$. The first stage aims to determine the minimum value of the generalized cost and generates the number and capacity values for the second stage. The second stage utilizes the number and capacity values from the first stage and determines the minimum value of the stochastic demand cost of transportation. In addition, to provide a comprehensive description of the optimization objective in emergency logistics, we introduce a nonlinear function based on the objective function in the initial model to calculate the generalized cost. The two-stage EVM for SCLAP can be formulated as follows.

First stage

$$\min_{p,\mathbf{S},\mathbf{x},\mathbf{y}} E\left(\sum_{i=1}^{p}\left(A+\delta\cdot s_i\right)+F_s\left(p,\mathbf{S}\right)\right) \tag{17}$$

is subject to

$$S_{\min}\le s_i\le S_{\max},\quad \forall i\in N; \tag{18}$$

$$\sum_{i=1}^{p}s_i\ge E\left(\sum_{j=1}^{m}\xi_j\left(\omega\right)\right),\quad \forall p\in\mathbf{N}. \tag{19}$$

Second stage

$$F_s\left(p,\mathbf{S}\right)=\min_{z\in Z(\omega)}\sum_{i=1}^{p}\sum_{j=1}^{m}z_{ij}\sqrt{\left(x_i-a_j\right)^2+\left(y_i-b_j\right)^2} \tag{20}$$

is subject to

$$\sum_{i=1}^{p}z_{ij}=\xi_j\left(\omega\right),\quad \forall j\in M,\ \omega\in\Omega; \tag{21}$$

$$\sum_{j=1}^{m}z_{ij}\le s_i,\quad \forall i\in N,\ s_i\in\mathbf{S}; \tag{22}$$

$$z_{ij}\ge 0,\quad \forall i\in N,\ j\in M; \tag{23}$$

$$\left(x_i,y_i\right)\in\mathbf{R}^2,\quad \forall i\in N; \tag{24}$$

$$\left(a_j,b_j\right)\in\mathbf{R}^2,\quad \forall j\in M, \tag{25}$$

where A ($A>0$) denotes the fixed cost of constructing a supply center and δ ($0\le\delta\le 1$) denotes the variable cost coefficient to maintaining an operating supply center. Equation (17) represents the expected generalized cost. The first portion, $\sum_{i=1}^{p}(A+\delta\cdot s_i)$, is the sum of the cost of construction and the cost of maintaining the supply centers, and the latter portion, $F_s(p,\mathbf{S})$, is the sum of the costs of the demands for transportation from each supply center to the customer, similar to the initial EVM. The first constraint in (18) ensures that the capacity of each supply center is limited within a reasonable range from $S_{\min}$ to $S_{\max}$. Equation (19) stipulates that the total capacity of supply centers must exceed the total expected demands of customers regardless of the value of $\mathbf{N}$ or the number of supply centers p. The formulations in the second stage denote the same items as those in the initial model, except that fixed constants n and s_i become variables.

2.4. Deterministic Equivalent Models. Evidently, the two-stage EVM model is much more complicated than the initial model, and the stochastic simulation method is extraordinarily time-consuming because the number and the capacities of supply centers are both unknown. However, on the basis of the definitions and theorems in Section 2.1, if we assume that the demand variables are nonnegative normal distributions, that is, $\forall j\in M,\ \xi_j\sim\mathcal{N}(e_j,\sigma_j)$, we can convert the two-stage EVM of the SCLAP into a deterministic two-stage model. The following equivalent formula of the first stage model can be expressed as

$$\min_{p,\mathbf{S},\mathbf{x},\mathbf{y}} pA+\delta\sum_{i=1}^{p}s_i+E\left(F_s\left(p,\mathbf{S}\right)\right) \tag{26}$$

and subject to

$$\begin{aligned} &S_{\min}\le s_i\le S_{\max},\quad \forall i\in N,\\ &\sum_{i=1}^{p}s_i\ge\sum_{j=1}^{m}e_j,\quad \forall p\in\mathbf{N}. \end{aligned} \tag{27}$$

Proof. Customer demand ξ follows the normalized distribution (Theorem 8), which belongs to the *regular uncertainty distribution*, and $f=\sum_{j=1}^{m}\xi_j(\omega)$ is a strictly increasing function. Hence, one has the following.

(1) Objective conversion is as follows:

$$\begin{aligned} &E\left(\sum_{i=1}^{p}\left(A+\delta\cdot s_i\right)+F_s\left(p,\mathbf{S}\right)\right)\\ &\quad=E\left(\sum_{i=1}^{p}\left(A+\delta\cdot s_i\right)\right)+E\left(F_s\left(p,\mathbf{S}\right)\right)\\ &\quad=pA+\delta\sum_{i=1}^{p}s_i+E\left(F_s\left(p,\mathbf{S}\right)\right). \end{aligned} \tag{28}$$

(2) Constraint conversion is as follows:

$\because \xi_j\sim\mathcal{N}(e_j,\sigma_j)$ and according to Theorem 8

$\therefore \sum_{j=1}^{m}\xi_j(\omega)\sim\mathcal{N}(\sum_{j=1}^{m}e_j,\sum_{j=1}^{m}\sigma_j)$

$\therefore \sum_{i=1}^{p}s_i\ge E(\sum_{j=1}^{m}\xi_j(\omega))\Rightarrow\sum_{i=1}^{p}s_i\ge\sum_{j=1}^{m}E(\xi_j(\omega))=\sum_{j=1}^{m}e_j$

The following equivalent formula of the second-stage model can be established as

$$\begin{aligned} &E\left(F_s\left(p,\mathbf{S}\right)\right)\\ &\quad=\min_{z\in E(Z(\omega))}\sum_{i=1}^{p}\sum_{j=1}^{m}z_{ij}\sqrt{\left(x_i-a_j\right)^2+\left(y_i-b_j\right)^2} \end{aligned} \tag{29}$$

and subject to

$$\begin{aligned} &\sum_{i=1}^{p}z_{ij}=e_j,\quad \forall j\in M,\ \omega\in\Omega;\\ &\sum_{j=1}^{m}z_{ij}\le s_i,\quad \forall i\in N,\ s_i\in\mathbf{S};\\ &z_{ij}\ge 0,\quad \forall i\in N,\ j\in M;\\ &\left(x_i,y_i\right)\in\mathbf{R}^2,\quad \forall i\in N;\\ &\left(a_j,b_j\right)\in\mathbf{R}^2,\quad \forall j\in M. \end{aligned} \tag{30}$$

□

Proof. Customer demand $\xi_j = \sum_{i=1}^{p} z_{ij}$ follows the normalized distribution (Theorem 8), which belongs to the *regular uncertainty distribution*. Hence, one has the following.

(1) Objective conversion is as follows:

$$\begin{aligned} & E\left(\min_{z\in Z(\omega)} \sum_{i=1}^{p}\sum_{j=1}^{m} z_{ij}\sqrt{\left(x_i-a_j\right)^2+\left(y_i-b_j\right)^2}\right) \\ &= \min_{z\in Z(\omega)} \sum_{j=1}^{m} E\left(\sum_{i=1}^{p} z_{ij}\sqrt{\left(x_i-a_j\right)^2+\left(y_i-b_j\right)^2}\right) \\ &= \min_{z\in Z(\omega)} \sum_{j=1}^{m}\sum_{i=1}^{p} E\left(z_{ij}\right)\sqrt{\left(x_i-a_j\right)^2+\left(y_i-b_j\right)^2} \\ &= \min_{z\in E(Z(\omega))} \sum_{i=1}^{p}\sum_{j=1}^{m} z_{ij}\sqrt{\left(x_i-a_j\right)^2+\left(y_i-b_j\right)^2}. \end{aligned} \tag{31}$$

(2) Constraint conversion is as follows:

$$\begin{aligned} & \xi_j = \sum_{i=1}^{p} z_{ij} \Longrightarrow E\left(\xi_j\right) = E\left(\sum_{i=1}^{p} z_{ij}\right) = \sum_{i=1}^{p} E\left(z_{ij}\right) \\ &= \sum_{i=1}^{p} z'_{ij} = e_j, \quad z'_{ij} \in E\left(Z\left(\omega\right)\right). \end{aligned} \tag{32}$$

After the conversion of the two-stage EVM, we can obtain a two-stage deterministic model, which can be solved much easily. Similarly, we can also convert the chance-constrained programing (CCP) model into a deterministic model. Assuming that the supply centers satisfy the demand of customer j with a probability β_j, the total supply capacity satisfies the total demand of customers with probability α, and the demand variable ξ_j $(\forall j \in M)$ complies with the same kind of regular stochastic distribution Φ_j, the CCP of the two-stage SCLAP can be formulated as follows (only the values different from the EVM are listed for simplicity).

Equation (19) is reformulated as

$$\Pr\left\{\sum_{i=1}^{p} s_i \geq \sum_{j=1}^{m} \xi_j\left(\omega\right)\right\} \geq \alpha, \quad \forall p \in \mathbf{N}; \tag{33}$$

and (21) is reformulated as

$$\Pr\left\{\sum_{i=1}^{p} z_{ij} \geq \xi_j\left(\omega\right)\right\} \geq \beta_j, \quad \forall j \in M, \quad \omega \in \Omega \tag{34}$$

According to Theorem 7, the stochastic function $f = \sum_{j=1}^{m} \xi_j(\omega)$ is strictly increasing. Therefore, one has the following.

Equation (33) can be converted into

$$\sum_{i=1}^{p} s_i \geq \sum_{j=1}^{m} \Phi_j^{-1}\left(\alpha\right), \quad \forall p \in \mathbf{N}, \tag{35}$$

and (34) can be converted into

$$\sum_{i=1}^{p} z_{ij} \geq \Phi_j^{-1}\left(\beta_j\right), \quad \forall j \in M. \tag{36}$$

However, if the SCLAP is a dependent-chance programing (DCP) model (only the values different from the EVM are listed for simplicity), (17) is reformulated as

$$\min_{p,\mathbf{S},\mathbf{x},\mathbf{y}} \Pr\left\{\sum_{i=1}^{p}\left(A+\delta\cdot s_i\right)+F_s\left(p,\mathbf{S}\right) \leq C\right\} \tag{37}$$

and (19) is reformulated as

$$\sum_{i=1}^{p} s_i \geq \sum_{j=1}^{m} \xi_j\left(\omega\right), \quad \forall p \in \mathbf{N}. \tag{38}$$

The objective is to maximize the probability, which cannot be converted into a deterministic model. Hence, the stochastic simulation process must be activated to calculate the probability. In this paper, we only focus on the EVM and develop an improved PSO (IPSO) algorithm to solve this problem given that the two-stage SCLAP is newly proposed. □

3. Improved IPSO

The LAP has been proven to be NP-hard [24]. Hence, it is a two-stage SCLAP. Heuristic methods are the best at handling large NP-hard problems, especially under uncertain environments. We develop an improved IPSO combined with the interior point method [25] to solve the deterministic equivalent conversion of the two-stage EVM.

3.1. Design of the IPSO. In 1995, Kennedy and Eberhart [26] first proposed a swarm intelligence optimization algorithm, which was inspired by the flocking of birds and schooling of fish against predation, called the particle swarm optimization (PSO). PSO has been successfully applied to parameter optimization [27], combinatorial optimization [28], pattern recognition [29, 30], data mining [31], and other fields [32, 33]. However, local optimal solution is often obtained when general PSO algorithm is applied to solve the continuous location of distribution centers. Therefore, scholars have proposed many improvement strategies for a much efficient and effective PSO [34–37]. Recently, Zhan et al. [38] proposed an improved PSO based on neighbor heuristic and Gaussian cloud learning, and the results proved its superiority over many PSO variants. We also adopt a Gaussian cloud operator combined with an adaptive parameter strategy and a Restart strategy to improve the general PSO algorithm of the two-stage SCLAP. For a detailed description of the IPSO algorithm, an algorithm flow chart is provided as a supplement in Figure 1.

In Figure 1, the loop program is repeated for IterMax times, and three improvement strategies, which are shown in the curved boxes, are adopted. If no better solution is found during ΔT ($\Delta T = \Delta$Iter), the *Restart strategy* will be activated;

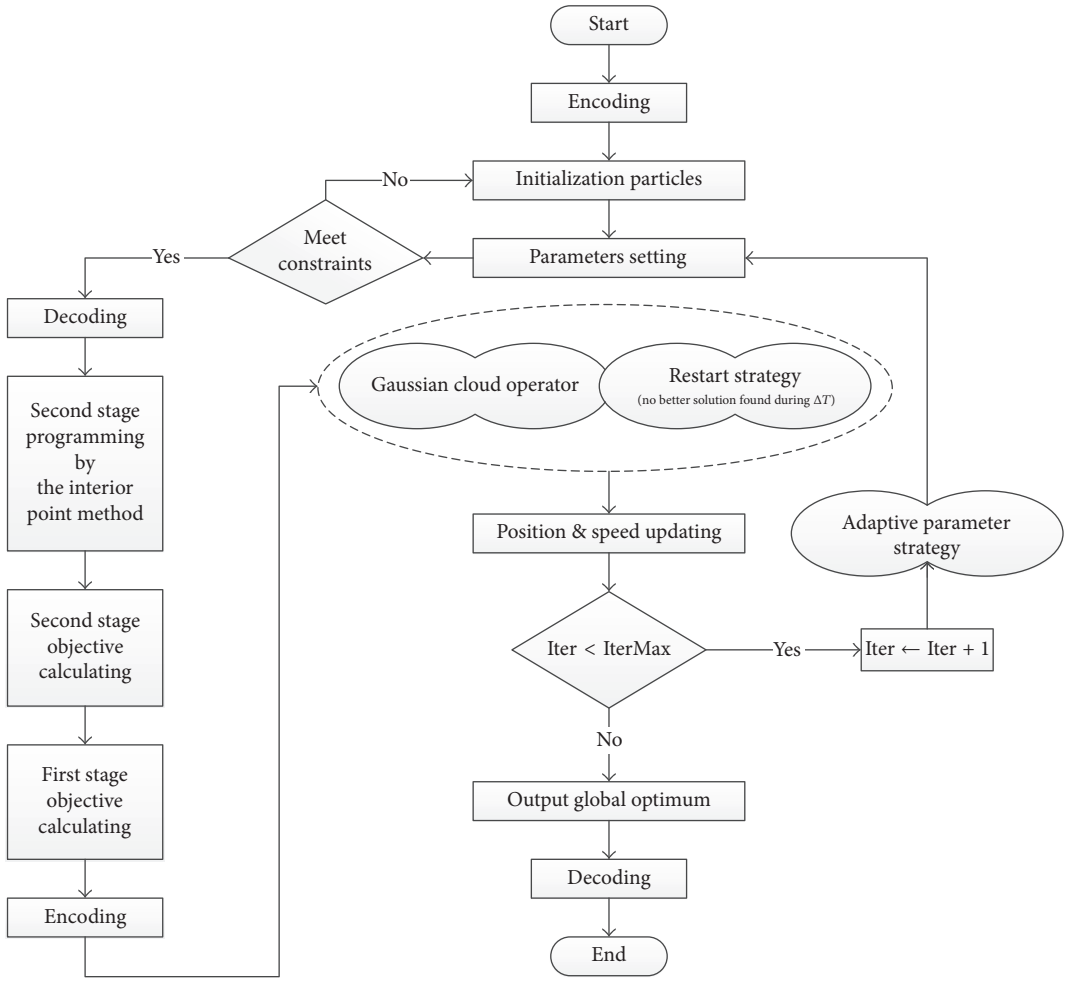


Figure 1: IPSO algorithm flow chart.

otherwise, the *Gaussian cloud operator* will be activated. In addition, several IPSO parameters will adaptively change along with the iteration (the *adaptive parameter strategy*).

Based on (27), the range of supply center number p can be determined by

$$p \geq \left[\sum_{j=1}^{m} \frac{e_j}{S_{\max}} \right] + 1, \tag{39}$$

where $[x]$ denotes taking the integer portion of x. The capacity of each supply center must be within $[S_{\min}, S_{\max}]$, which means that if a candidate solution of s_i exceeds $S_{\max}$, a penalty will be given in the general cost function, and if s_i is less than $S_{\min}$, it should be reset to $S_{\min}$. Therefore, the minimum number p exists but the maximum number does not. The cost of building and maintaining supply centers will be greatly wasted if the total capacity of supply centers exceeds the actual expected demands of customers, especially in the EVM and its crisp variant model. Fixed value A, which denotes building a supply center, is the decision factor in choosing the best number p for the optimal solution (discussed in Section 4). For each p, the main procedure of IPSO can be described as follows.

Step 1. Initialize each individual of the particle swarm population with a given size pop_size as follows:

$$\begin{aligned} \mathrm{Ind}^k &= \left\{ \mathbf{x}^k, \mathbf{y}^k, \mathbf{S}^k \right\}_{1\times 3p} \\ &= \left\{ x_1^k, x_2^k, \ldots, x_p^k, y_1^k, y_2^k, \ldots, y_p^k, s_1^k, s_2^k, \ldots, s_p^k \right\}, \\ & \qquad k = 1, 2, \ldots, \mathrm{pop\_size}, \end{aligned} \tag{40}$$

where Ind^k must satisfy the constraints (introduced in Section 3.3), which are generated randomly as in Algorithm 1. TD denotes the total demand of customers and $E(\mathrm{TD})$ denotes the expected value equal to $\sum_{j=1}^{m} e_j$.

Step 2. Encode the initial individuals into normalized values to update the particle swarm. The normalization formula is as follows:

$$N\mathrm{Ind}^k = \mathrm{Norm}\left(\mathrm{Ind}^k\right) = \left\{ \frac{\left(\mathbf{x}^k - x_{\min}\right)}{\left(x_{\max} - x_{\min}\right)}, \right.$$

Note: Initialize individuals with pop_size, p, $S_{\min}$, $S_{\max}$, and TD
(1) **for all** $k \in$ pop_size **do**
(2) **for** $j = 1 \rightarrow 2p$ **do**
(3) **if** $j <= p$ **then**
(4) $x \leftarrow$ random(0, 1)
(5) $N\text{Ind}_{kj} \leftarrow x$
(6) **else if** $j > p$ & $j <= 2p$ **then**
(7) $y \leftarrow$ random(0, 1)
(8) $N\text{Ind}_{kj} \leftarrow y$
(9) **end if**
(10) **end for**
(11) **while** sum(S) < E(TD) **do**
(12) **for** $j = 1 \rightarrow p$ **do**
(13) $S_j \leftarrow$ random($S_{\min}, S_{\max}$)
(14) **end for**
(15) **end while**
//Normalize S and complete the construction of NInd
(16) $S' \leftarrow (S - S_{\min})/(S_{\max} - S_{\min})$
(17) **for** $j = 2p + 1 \rightarrow 3p$ **do**
(18) $N\text{Ind}_{kj} \leftarrow S'_{j-2p}$
(19) **end for**
(20) **end for**
Output: NInd

Algorithm 1: The initialization algorithm.

$$\left.\frac{\left(\mathbf{y}^k - y_{\min}\right)}{\left(y_{\max} - y_{\min}\right)}, \frac{\left(\mathbf{S}^k - S_{\min}\right)}{\left(S_{\max} - S_{\min}\right)}\right\}_{1\times 3p} = \left\{\mathbf{x}'^k, \mathbf{y}'^k, \mathbf{S}'^k\right\}_{1\times 3p}, \tag{41}$$

where $x_{\max}$ and $y_{\max}$ denote the upper-bounds of the location coordinates and $x_{\min}$ and $y_{\min}$ denote the lower-bounds of the location coordinates. Specifically, we set the position coordinate bounds as follows:

$$\begin{aligned} x_{\min} &= \min_{j\in M} a_j, \\ x_{\max} &= \min_{j\in M} a_j; \\ y_{\min} &= \min_{j\in M} b_j, \\ y_{\max} &= \min_{j\in M} b_j. \end{aligned} \tag{42}$$

Then, the particle individuals are normalized, and $N\text{Ind}^k$ is equal to the position of particle k in the continuous search space.

Step 3. Set the parameters for the IPSO algorithm operating at each iteration. These parameters are listed in Table 1.

w, c_1, and c_2 change adaptively at each iteration. This phenomenon is called adaptive unfixed phenomenon, which is discussed in Section 3.2. The positions are set between 0 and 1 because the individuals have been normalized. The speed limitation can be set to any reasonable symmetric interval. It is set at [−0.3, 0.3]. The dimensions of the position and the speed vectors are both equal to $3p$.

Step 4. Check the constraints for each individual. If feasible, calculate the fitness values of the second and first stage objective functions. The second-stage programing is linear programing, which is solved by the interior point method and discussed in Section 3.3. Otherwise, regenerate the individual using the initialization process in Algorithm 1.

Step 5. Update the position and speed of particles using different strategies. If the global optimum does not improve within a given cumulative iteration $\Delta T = \Delta\text{Iter}$, the *Restart strategy* is activated to reinitialize all individuals based on the global best-so-far solution and the Gaussian cloud model. Otherwise, the *Gaussian cloud operator* is activated to update the particles based on the global best-so-far solution and the individual best-so-far solution. The two updating strategies are discussed in Section 3.2.

Step 6. Repeat Steps 3–5 for IterMax.

Step 7. Decode the global best-so-far solution into the initial range and generate the global optimum. The inverse normalization formula is as follows:

$$\begin{aligned} \text{Ind}^k = \text{Norm}^{-1}\left(N\text{Ind}^k\right) = \left\{\mathbf{x}'^k \cdot \left(x_{\max} - x_{\min}\right)\right. \\ + x_{\min}, \mathbf{y}'^k \cdot \left(y_{\max} - y_{\min}\right) + y_{\min}, \mathbf{S}'^k \\ \left.\cdot \left(S_{\max} - S_{\min}\right) + S_{\min}\right\}_{1\times 3p} = \left\{\mathbf{x}^k, \mathbf{y}^k, \mathbf{S}^k\right\}_{1\times 3p}. \end{aligned} \tag{43}$$

Table 1: Description of parameters in the IPSO algorithm.

Notation	Name	Range	State
w	Inertia weight	$0 < w < 1$	Adaptive unfixed
c_1, c_2	Acceleration factor	$c_1, c_2 \geq 0$	Adaptive unfixed
$[X_{\min}, X_{\max}]^{D=3p}$	Position limitation	$X_{\min} = 0$, $X_{\max} = 1$	Fixed
$[V_{\min}, V_{\max}]^{D=3p}$	Speed limitation	$V_{\min} = -0.3$, $V_{\max} = 0.3$	Fixed

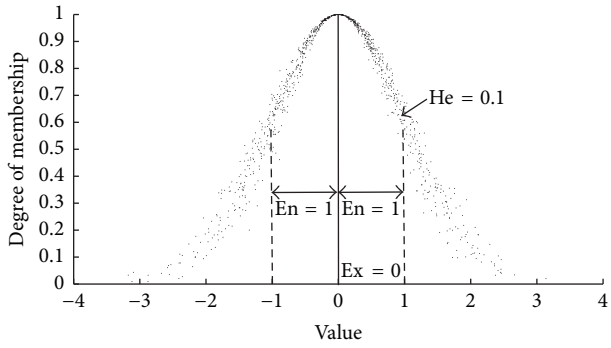


Figure 2: Classic Gaussian cloud with expectation Ex, entropy En, and hyperentropy He.

3.2. Improvement Strategies

3.2.1. Gaussian Cloud Operator. The *Gaussian cloud operator* is an algorithm application of the membership clouds. The membership cloud was first proposed by Deyi et al. [39], a member of the Chinese Academy of Sciences. The method bridges the gap between quantitative methodology and qualitative methodology based on the fuzzy set theory and has been successfully applied in algorithm improvement [38, 40]. A classic Gaussian cloud is described in Figure 2.

Assuming $\forall k \in$ pop_size, we have $X_k = N\text{Ind}^k$, and the particle population is $\mathbf{X} = (X_1, X_2, \ldots, X_{\text{pop\_size}})^T$. Each particle position vector $X_k = (x_{k1}, x_{k2}, \ldots, x_{kD})$ in a D-dimension solution space ($D = 3p$) represents a potential solution, which is accordingly associated with a speed vector $V_k = (v_{k1}, v_{k2}, \ldots, v_{kD})$. The individual best-so-far solution of each particle can be expressed as $P_k = (p_{k1}, p_{k2}, \ldots, p_{kD})$ and the global best-so-far solution can be expressed as $P_g = (p_{g1}, p_{g2}, \ldots, p_{gD})$ based on the previous fitness value of each particle. In a traditional PSO updating process, P_k and P_g are used as heuristic factors in

$$v_{kd}^{t+1} = wv_{kd}^t + c_1 r_1 \left(p_{kd}^t - x_{kd}^t\right) + c_2 r_2 \left(p_{gd}^t - x_{kd}^t\right), \quad (44)$$

$$x_{kd}^{t+1} = x_{kd}^t + v_{kd}^{t+1}, \quad (45)$$

where $d = 1, 2, \ldots, D$, and other notations are shown in Table 1. However, with the *Gaussian cloud operator*, (39) will be modified as

$$v_{kd}^{t+1} = wv_{kd}^t + c_1 r_1 \left(p_{kd}^t - x_{kd}^t\right) + c_2 r_2 \left(gcp_{gd}^t - x_{kd}^t\right), \quad (46)$$

where $gcp_{gd} = \text{GC}(\text{Ex}, \text{En}, \text{He})$, $\text{Ex} = p_{gd}$, and $x = \text{GC}(\text{Ex}, \text{En}, \text{He})$ are the cloud drop generator functions, which can be described as follows:

Step 1. Generate a normal random number σ based on the expectation value En, and the standard deviation value is He, that is, $\sigma \sim \mathcal{N}(\text{En}, \text{He})$.

Step 2. Generate a normal random number x based on the expectation value Ex, and the standard deviation value is σ; that is, $x \sim \mathcal{N}(\text{Ex}, \sigma)$.

We do not replace each dimension of initial P_g with cloud drop gcp_{gd} because it may induce excessive noise and disturbance. The reasonable approach is to replace one random dth-dimension of the initial P_g at each iteration.

3.2.2. Restart Strategy. The *Restart strategy* is another method to keep the algorithm from premature convergence. If the global optimum P_g does not improve within ΔT time ($\Delta T = \Delta\text{Iter}$), that is, $P_g^T = P_g^{T+\Delta T}$, the particle population is disrupted and reinitialized based on P_g. In $\forall k \in$ pop_size, we also use the *Gaussian cloud operator* to reinitialize the different input parameters as follows:

$$x_{kd} = \text{GC}\left(p_{gd}, \text{En}', \text{He}'\right), \quad (47)$$

where $\text{En}' = 0.2$ and $\text{He}' = 0.02$. Subsequently, the position vector should be checked and modified, as in Algorithm 1, to satisfy the limitation constraint.

3.2.3. Adaptive Parameter Strategy. Parameter setting always significantly affects the intelligent algorithm performance. The main related parameters of the IPSO algorithm are as follows: the number of iteration IterMax, the population size (pop_size), the inertia weight (w), the acceleration factors (c_1 and c_2), the entropy (En), and the hyperentropy (He) in the *Gaussian cloud operator*. IterMax and pop_size are relatively fixed and have strong effects on computing time. Hence, we only utilize the adaptive parameter strategies on w, c_1, c_2, En, and He.

The inertia weight (w) represents the particle inheritability of the previous speed, that is, a larger w means a stronger ability in global searching, whereas a smaller w means a stronger ability in local searching. A linear decreasing inertia weight (LDIW), which was first proposed by Shi and Eberhart [41], is adopted in

$$w_{\text{Iter}} = w_{\max} - \left(w_{\max} - w_{\min}\right) \frac{\text{Iter}}{\text{IterMax}}, \quad (48)$$

where $w_{\min}$ is set to 0.4 and $w_{\max}$ is set to 0.9.

The acceleration factors (c_1 and c_2) represent the particle weights on the attractiveness of P_k and P_g, respectively, which

also reflect the balance between global and local searching. A typical setting is $c_1 = c_2 = 2$. Several studies also proposed other effective settings, such as $c_1 = c_2 = 1.49$ and $c_1 = 2.8$, $c_2 = 1.3$ [42]. We propose a linear adaptive parameter strategy that can be formulated as shown in

$$c_1 = 2.5 - \frac{\text{Iter}}{\text{IterMax}}, \qquad c_2 = 3.5 - c_1. \tag{49}$$

As a result, the algorithm will have a larger c_1 and a smaller c_2 at the beginning, which benefit global searching, and a smaller c_1 and a larger c_2 in the latter stages, which benefit local searching.

The two important parameters in the *Gaussian cloud operator*, which are called the entropy En and the hyperentropy He, could also be changed adaptively. The mutation of a random dimension value is more beneficial at the beginning of the iteration than at the later time. Hence, we propose a nonlinear adaptive parameter strategy for En and He as follows:

$$\text{En} = \text{En}_0 \left(1 - \frac{\text{Iter}}{\text{IterMax}}\right)^2, \qquad \text{He} = 0.1\text{En}, \tag{50}$$

where En_0 is set to 0.05 based on multiple tests.

3.3. Interior Point Method. To rapidly solve the second-stage programing ((29) and (30)), we adopt the interior point method instead of the simplex method in this paper. The interior point method was first proposed by Karmarkar [25] in 1984. It is a polynomial algorithm for linear programing that requires $O(n^{3.5}L)$ arithmetic operations on $O(L)$ bit numbers in the worst case, where n is the number of variables and L is the number of bits in the input. Compared with the simplex method, it is very suitable for solving large-scale problems because the computing time does not increase with the size of the problem. Moreover, it is not strict with the initial point requirements of quadratic convergence and robustness. The improvement of the algorithm efficiency is verified in Section 4.3.

4. Numerical Example

4.1. Data and Implementation. In this section, a 20-customer network of allocation and transportation in emergency logistics is designed, and the decision objective is to determine the number, capacities, and locations of supply centers for material support. The customer locations and their stochastic demands are listed in Table 2.

We adopt the stochastic simulation method [7] to solve the two-stage SCLAP, and the simulation number of time R is set to 100 for a fair comparison with the traditional EVM. The generalized cost is significantly affected by fixed cost A and variable cost coefficient δ. Hence, we test eight pairs of different values for comparison and analysis. The eight pairs are divided into two classes using the controlling variable method for testing. The values are listed in Table 3.

Table 2: Customer locations and stochastic demand distributions.

Number	Locations	Demand distributions
(1)	(28, 42)	$\mathcal{N}(2, 1)$
(2)	(18, 50)	$\mathcal{N}(4, 1)$
(3)	(74, 34)	$\mathcal{N}(7, 1)$
(4)	(74, 6)	$\mathcal{N}(6, 1)$
(5)	(70, 18)	$\mathcal{N}(4, 1)$
(6)	(72, 98)	$\mathcal{N}(10, 1)$
(7)	(60, 50)	$\mathcal{N}(4, 1)$
(8)	(36, 40)	$\mathcal{N}(2, 1)$
(9)	(12, 4)	$\mathcal{N}(5, 1)$
(10)	(18, 20)	$\mathcal{N}(10, 1)$
(11)	(14, 78)	$\mathcal{N}(2, 1)$
(12)	(90, 36)	$\mathcal{N}(6, 1)$
(13)	(78, 20)	$\mathcal{N}(7, 1)$
(14)	(24, 52)	$\mathcal{N}(10, 1)$
(15)	(54, 6)	$\mathcal{N}(8, 1)$
(16)	(62, 60)	$\mathcal{N}(7, 1)$
(17)	(98, 14)	$\mathcal{N}(2, 1)$
(18)	(36, 58)	$\mathcal{N}(6, 1)$
(19)	(38, 88)	$\mathcal{N}(9, 1)$
(20)	(32, 54)	$\mathcal{N}(7, 1)$

The experiments are conducted on a laptop with Intel® Core™ i7-6700HQ 2.60 GHz quad-core processors, and the algorithms are coded in MATLAB R2016a environment. For convenience, we adopt the built-in function *linprog* in MATLAB to achieve linear programing with the simplex method or the interior point method for the second-stage optimization problem.

4.2. Computational Results

4.2.1. Results of the Deterministic Model. We can achieve an optimal solution for the deterministic model from the initial EVM using the IPSO. pop_size is set to 10 and IterMax is set to 500 for the IPSO algorithm. The capacity limitation of supply centers is set as $S_{\min} = 30$ and $S_{\max} = 100$. The fixed cost and the variable cost coefficient for the generalized cost function are set as $A = 500$ and $\delta = 0.1$, respectively. The results are listed in Table 4 where the number of supply centers p will accordingly change from two to four. Cap. denotes the capacity and Pos. denotes the position of each supply center. GC denotes the generalized cost of each solution with different numbers of centers and CT denotes the computing time of each situation.

The best decision solution is to build three supply centers, and the optimum generalized cost is 3643.147. We also present the convergence curves of different p numbers in Figure 3 and illustrate the solutions with LA maps in Figures 4–6. In addition, the amount of traffic is simply represented with different shades of color.

In the traditional one-stage problem, the number and capacity of a supply center are fixed, whereas the number (p) and capacity (Cap.) are decision variables in this problem.

Table 3: Fixed cost A and variable cost coefficient δ.

Class 1	A	200	300	500	1000
	δ	0.1	0.1	0.1	0.1
Class 2	A	400	400	400	400
	δ	0.1	0.2	0.4	0.7

Table 4: Results of the deterministic model.

p	Cap.	Pos.	GC	CT
2	78.000	(32.002, 54.000)	3775.291	29.57 s
	40.014	(76.318, 19.881)		
3	30.000	(57.580, 86.522)	3643.147	31.33 s
	44.000	(75.068, 20.762)		
	48.000	(25.065, 49.983)		
4	30.000	(57.521, 86.513)	3655.400	33.63 s
	40.000	(76.379, 19.922)		
	30.000	(18.000, 20.000)		
	37.000	(31.922, 53.975)		

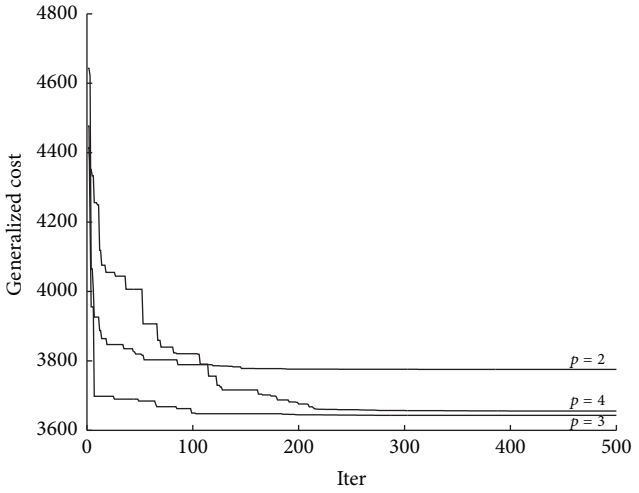


Figure 3: Convergence curves with different number of supply centers.

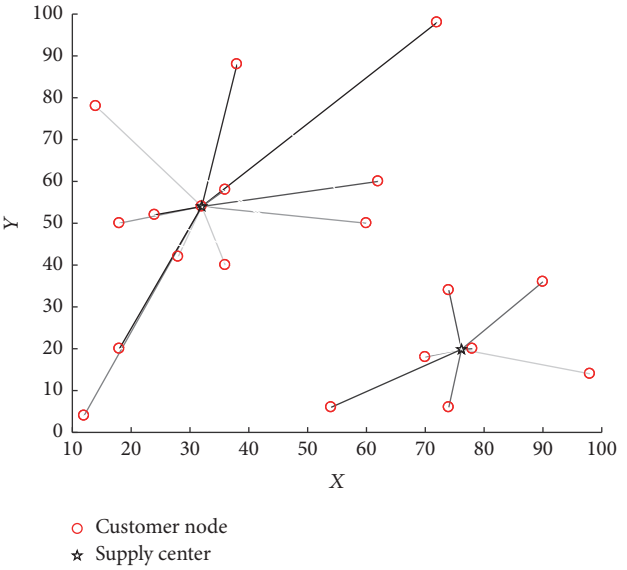


Figure 4: LA map with $p = 2$.

Moreover, the capacity has a range. The results of the two-stage SCLAP can provide significant practical conclusions for actual emergency logistics because the scarcity of supply materials is extremely serious and urgent such that the number and capacity of supply centers must be determined accurately based on practical situations. Therefore, we must achieve a balance between minimizing the number of centers and minimizing the distance of transport.

4.2.2. Results of the EVM with Stochastic Simulation. Stochastic simulation is time-consuming and extremely sensitive to the number of simulation iterations (R). Only when R is sufficiently large can the solution become stable. As a result, R is set to 100 and a parallel strategy, that is, a *parfor* loop with the MATLAB software, is adopted. For each simulation, the random demands of all customers are generated, and the programing model is solved. The IPSO obtains a stable optimal solution based on the expected value of the generalized cost at each iteration. The results of the EVM with stochastic simulation are listed in Table 5.

Table 5 shows that the best decision solution is also to build three supply centers, and the optimum generalized cost is 3660.836. The GC values are slightly larger than those in Table 4 because the simulation inevitably induces several random factors. Thus, obtaining the theoretical optimal solution is impossible. Cap. and Pos. are also different than those in Table 4 for the same reason. However, the CT values are significantly larger than those in the deterministic model, which implies that the deterministic model can dramatically improve the computational efficiency.

Table 5: Results of the EVM with stochastic simulation.

p	Cap.	Pos.	GC	CT
2	40.123	(76.615, 19.827)	3785.601	1010.467 s
	78.093	(32.578, 55.244)		
3	30.000	(55.178, 86.894)	3660.836	1035.310 s
	40.034	(76.309, 19.186)		
	52.326	(25.479, 48.314)		
4	37.128	(29.474, 52.955)	3674.496	1055.373 s
	30.000	(56.231, 89.416)		
	30.000	(18.218, 18.944)		
	40.071	(76.446, 18.986)		

Table 6: Best results of the 10 tests with different A or δ.

p	Class 1 ($\delta = 0.1$)				Class 2 ($A = 400$)			
	$A = 200$	$A = 300$	$A = 500$	$A = 1000$	$\delta = 0.1$	$\delta = 0.2$	$\delta = 0.4$	$\delta = 0.7$
2	3180.616	3378.905	3775.291	**4768.959**	3582.563	3583.987	3613.299	3660.975
3	2744.854	3045.227	**3643.147**	5147.107	3344.534	3361.372	3382.718	3421.571
4	2460.016	2861.945	3655.400	5658.357	**3259.020**	**3274.130**	**3302.576**	**3345.515**
5	2353.784	**2858.423**	3854.877	6352.455	3349.796	3370.884	3408.197	3449.036
6	**2290.738**	2921.248	4107.793	7174.363	3451.416	3466.693	3505.663	3558.175
7	2313.417	3052.983	4516.247	7971.316	3671.614	3719.131	3775.696	3838.061
8	2424.430	3214.099	4881.651	8877.735	4031.114	4031.241	4062.632	4141.791

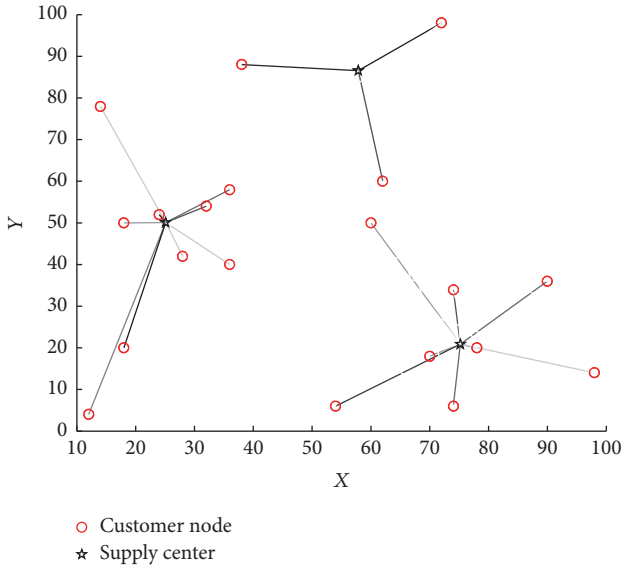


Figure 5: LA map with $p = 3$.

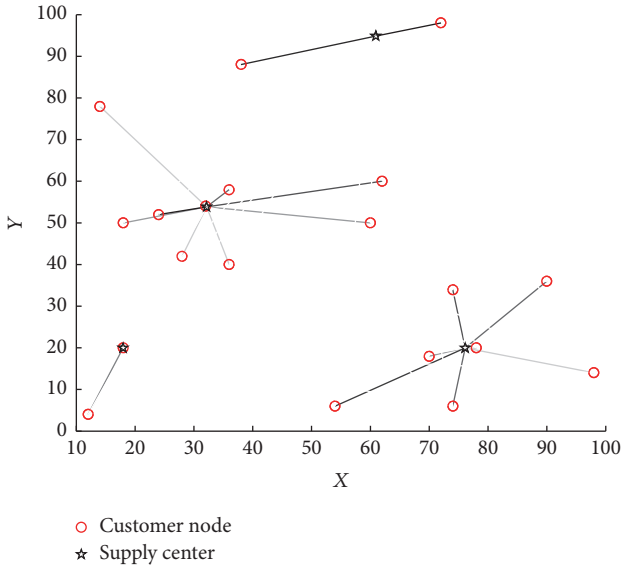


Figure 6: LA map with $p = 4$.

4.2.3. Parameter Analysis of Generalized Cost. The generalized cost of the optimal solution is dramatically affected by fixed cost A and variable cost coefficient δ. We design two experiments based on the data in Table 3 to demonstrate this phenomenon, and the number of supply centers (p) is set from 2 to 8 for each class testing. The experiments are carried out by the IPSO algorithm with the same parameters for 10 times. The best results are recorded in Table 6 and shown in Figures 7 and 8.

In Table 6, the bold values represent the minimum GCs under different p values. The most suitable number of supply centers should be built according to the given fixed cost value A and variable cost coefficient δ. Parameter A plays an important role in determining the best p. The best p increases when A decreases, as shown in Figure 7. In addition, the GC value nonlinearly increases with p. Therefore, if the number of supply centers exceeds the actual required number based on the actual total demand, a large cost will be wasted. With

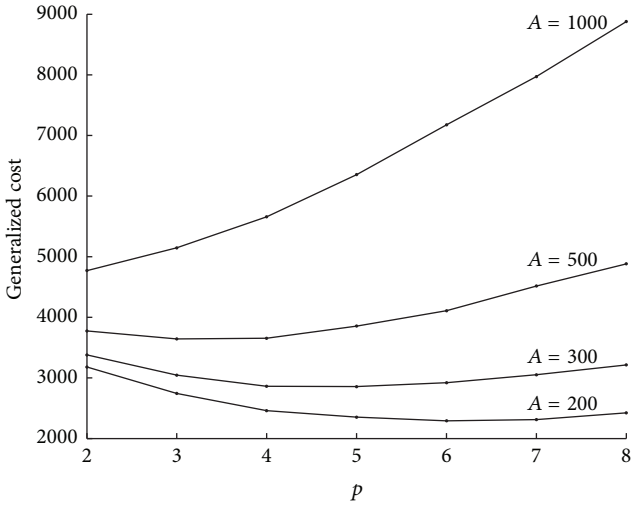


Figure 7: Changes in GC with different fixed cost values A ($\delta = 0.1$).

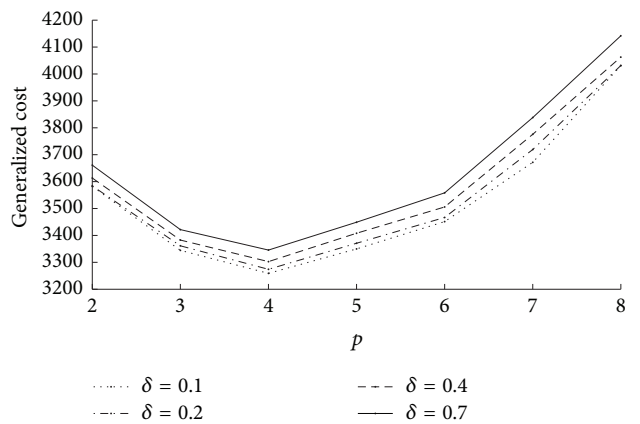


Figure 8: Changes in GC with variable cost coefficient δ ($A = 400$).

variable cost coefficient δ, the GC value increases linearly with p and has no effect on the best p decision, as shown in Table 6 and Figure 8. As a result, the most logical and economical approach is to reduce the fixed cost value of building each supply center.

4.3. Comparison and Analysis of the Algorithms. To provide a fair comparison of the IPSO and PSO algorithms, we use the same data in Table 2 and the common parameter settings in Section 4.2.1 to solve the two-stage EVM. However, in the basic PSO algorithm, the fixed parameters are set as follows: $w = 0.9$ and $c_1 = c_2 = 2$. The test is repeated 10 times, and the statistical results are presented in Table 7. Avg GC denotes the average GC value of the 10 tests, and the Best GC denotes the best value of these tests. Avg Time denotes the average running time. Table 7 shows that the IPSO algorithm is better than the basic PSO algorithm in both convergence accuracy and efficiency, which means that the improvement strategies are effective.

The interior point method is adopted in the IPSO algorithm. However, its advantage over the simplex method must also be verified. As a result, we test the IPSO algorithm with the simplex method in solving the second-stage programing of the deterministic model. The results are listed in row 3 of Table 7 with an asterisk. The Avg GC and Best GC results obtained by the simplex method are close to the results of the interior point method. However, Avg Time shows that the interior point method is much better than the simplex method in computing efficiency.

To further illustrate the applicability and superiority of the proposed IPSO algorithm, additional tests, based on the classic one-stage EVM, are performed. Solving the classic one-stage EVM is the same as solving the second stage of a two-stage EVM with known p and $\mathbf{S}$. As a result, the initial individuals are recoded and $\text{Ind}^k = \{\mathbf{x}^k, \mathbf{y}^k, \mathbf{S}^k\}_{1\times 3p}$ is changed to $\text{Ind}^k = \{\mathbf{x}^k, \mathbf{y}^k\}_{1\times 3p^*}$, where p^* is the given number of supply centers, which is 4, and $\mathbf{S}$ becomes a known variable [40, 50, 60, 70]. Zhou and Liu [7] developed a hybrid intelligent algorithm (HIA) that consists of a network simplex algorithm, a simulation, and a genetic algorithm to solve this model. HIA is used to compare the proposed IPSO algorithm and the classic PSO and IPSO* algorithms. The PSO, IPSO, and IPSO* algorithms share the same parameter setting with the previous test. HIA parameters are set as in [7]; that is, the crossover probability is 0.3, the mutation probability is 0.2, the population size is 10, the iteration number is 500, and the simulation number is 500. The numbers of customers and locations in the test data are completely consistent with the data in Table 2. However, the demand distributions are different. The expected demand for the first 10 customers is 5, that for the last 10 customers is 10, and the customer demand variance is 1. The test is repeated 10 times, and the results are recorded in Table 8. Then, some statistical results are calculated and presented in Table 9, where Best GC denotes the minimum value of general cost as shown in (20), Avg GCE denotes the average percent error of the general cost, Best CT denotes the minimum value of computing time, and Avg CTE denotes the average percent error of computing time. The percent error is calculated as (actual value − optimal value)/optimal value × 100%, which can be used to show the deviation from the optimal solution.

The results in Tables 8 and 9 show that the performance of the IPSO algorithm is the most superior in terms of convergence accuracy and convergence speed. In addition, on the basis of the Avg GCE and the Avg CTE, IPSO is the most stable algorithm among the four in terms of accuracy. It also performs stably in terms of computing time. Adopting the simulation method for HIA results in spending excessive time in the evaluation cycle. Hence, the computing time is much longer than those of the other three algorithms.

In addition, the two tests also illustrate the difference between the two EVMs. The proposed two-stage EVM assumes that the supply center number, capacities, and locations are decision variables. This approach is applicable in emergency logistics because it avoids wasting emergency resources for inappropriate number of supply centers or capacities. However, in the classic one-stage EVM, the numbers of centers and capacities are fixed. If the fixed cost A is set at 500 and the variable cost coefficient δ is set at 0.1, we can

Table 7: Statistical results of the different algorithms used to solve the two-stage EVM.

Algorithm	Avg GC	Best GC	Avg Time
PSO	3664.726	3651.823	30.60 s
IPSO	3646.752	3643.318	30.33 s
IPSO*	3648.000	3639.652	83.91 s

∗ denotes the use of the simplex method.

Table 8: Results of the 10 tests of different algorithms solving the classic one-stage EVM.

	PSO		IPSO		IPSO*		HIA	
	GC	CT	GC	CT	GC	CT	GC	CT
1	2202	31.8 s	2151	31.5 s	2136	81.4 s	2194	66 min
2	2191	31.5 s	2151	30.2 s	2138	87.4 s	2185	65 min
3	2132	31.7 s	2151	31.1 s	2132	79.1 s	2192	68 min
4	2202	32.4 s	2132	31.9 s	2151	80.5 s	2158	71 min
5	2132	31.1 s	2132	32.2 s	2151	92.1 s	2239	65 min
6	2132	31.2 s	2151	30.9 s	2132	82.0 s	2173	69 min
7	2292	31.4 s	2151	31.3 s	2151	83.1 s	2204	66 min
8	2217	31.8 s	2132	31.3 s	2202	83.0 s	2153	69 min
9	2247	32.1 s	2132	32.5 s	2132	81.8 s	2214	68 min
10	2221	31.8 s	2132	31.4 s	2152	81.6 s	2169	69 min

∗ denotes the use of the simplex method.

Table 9: Statistical results of different algorithms solving the classic one-stage EVM.

	PSO	IPSO	IPSO*	HIA
Best GC	2132	2132	2132	2153
Avg GCE (%)	3.04	0.45	0.74	1.63
Best CT	31.1 s	30.2 s	79.1 s	65 min
Avg CTE (%)	1.86	4.07	5.18	4.00

∗ denotes the use of the simplex method.

also obtain the same general cost with the two-stage EVM. The first stage objective in the two-stage EVM is 4147 when the optimal solution has a one-stage EVM general cost of 2132, as shown in Table 9. However, if we initially use the two-stage EVM, we can obtain the first stage objective optimal solution of 4076. The optimal decision is to set three supply centers with capacities of 30, 65, and 55, and the second-stage objective is 2561. Evidently, the total cost obtained by the two-stage EVM is lower and much reasonable than the classic one-stage EVM.

5. Conclusion

In this paper, a two-stage SCLAP in the context of emergency logistics is considered. In this problem, the number and capacities of supply centers are uncertain and must be determined. This condition is practical in emergency logistics compared to general logistics, which assumes that the number and capacities of supply centers are explicit and fixed. To solve this problem, a two-stage EVM and a generalized cost function are proposed. However, we convert the initial model into a deterministic model based on the stochastic theory and the uncertain theory proposed by Liu [5, 6] given that the traditional stochastic programing with a simulation method is time-consuming and unstable. To solve this model, we develop an IPSO algorithm and test it based on well-known 20-customer data. In this IPSO algorithm, three improvement strategies are introduced, namely, the Gaussian cloud operator, the Restart strategy, and the adaptive parameter strategy. In addition, we adopt the interior point method instead of the simplex method [7] to solve the second stage of programing. The comparison tests prove that the methods can improve the precision and convergence rates dramatically. The tests reveal the difference between the classic one-stage EVM and the two-stage EVM proposed in this paper and verify that the latter is much reasonable for emergency logistics. We also perform an analysis of the parameters in the proposed generalized cost function, drawing a conclusion that the fixed cost value has an important influence on the decision on the best number of supply centers.

However, this study has the following limitations. (1) The conversion of the deterministic model is only possible when the demand of the customer follows a normal distribution.

(2) In this paper, only the EVM is converted and studied. According to different optimization problems, stochastic programing can also be constructed as other models, such as the CCP model and the DCP model. However, only the CCP model can be converted into a deterministic model. (3) In actual emergency logistics, the transportation cost between two nodes is probably uncertain and can also be assumed as a stochastic variable. However, it is taken as a deterministic variable in this paper and can be a significant research direction in the future.

In addition, several interesting subjects should be considered in future research. For example, a new SCLAP that builds new centers and opens old centers must be considered. The objective must be to choose the best locations to build new centers and find the most appropriate old centers to close simultaneously. Other models with multiple objectives, complex constraints, or fuzzy variables can also be considered. The uncertain CLAP can also be solved with robust optimization, which is becoming a popular new research topic.

Descriptions of Notations and Variables

N: The supply center index set; $N = \{1, 2, \ldots, n\}$ when in the initial SCLAP model, and $N = \{1, 2, \ldots, p\}$ when in the two-stage SCLAP model

M: The customer index set in the initial SCLAP model, $M = \{1, 2, \ldots, m\}$

(x_i, y_i): The location of supply center i, which belongs to a 2D real vector space $\mathbf{R}^2$ and $(\mathbf{x}, \mathbf{y}) = \bigcup_{i \in N}(x_i, y_i)$

(a_j, b_j): The location of customer j, which belongs to a 2D real vector space $\mathbf{R}^2$

z_{ij}: The unit transportation cost of service from i to j, which is equal to the transportation volume

s_i: The capacity of supply center i

$Z(\omega)$: The solution space of transportation volumes under scenario ω

$\xi_j(\omega)$: The stochastic demand variable of customer j under scenario ω

p: The number of supply centers in the two-stage SCLAP model

$\mathbf{S}$: The capacity set of all supply centers in the two-stage SCLAP model; $\mathbf{S} = \{s_1, s_2, \ldots, s_p\}$.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the preeminent youth fund of Jiangsu Province in China (no. BK20150724).

References

[1] L. Cooper, "Location-allocation problems," *Operations Research*, vol. 11, pp. 331–343, 1963.

[2] R. Logendran and M. P. Terrell, "Uncapacitated plant location-allocation problems with price sensitive stochastic demands," *Computers & Operations Research*, vol. 15, no. 2, pp. 189–198, 1991.

[3] E. Carrizosa, E. Conde, M. Muñoz-Marquez, and J. Puerto, "The generalized Weber problem with expected distances," *RAIRO—Operations Research*, vol. 29, no. 29, pp. 35–57, 1995.

[4] E. Carrizosa and J. Puerto, "A note on the optimal positioning of service units," *Operations Research*, vol. 46, no. 1, pp. 155-156, 1998.

[5] B. Liu, *Theory and Practice of Uncertain Programming: Physica-Verlag*, 2002.

[6] B. Liu, "uncertainty theory," *Studies in Computational Intelligence*, vol. 154, no. 3, pp. 1–79, 2015.

[7] J. Zhou and B. Liu, "New stochastic models for capacitated location-allocation problem," *Computers & Industrial Engineering*, vol. 45, no. 1, pp. 115–125, 2003.

[8] J. Zhou and B. Liu, "Modeling capacitated locationallocation problem with fuzzy demands," *Computers & Industrial Engineering.*, vol. 53, no. 3, pp. 454–468, 2007.

[9] F. J. F. Silva and D. S. De La Figuera, "A capacitated facility location problem with constrained backlogging probabilities," *International Journal of Production Research*, vol. 45, no. 21, pp. 5117–5134, 2007.

[10] B. H. Wang and H. E. Shi-Wei, "robust optimization model and algorithm for logistics center location and allocation under uncertain environment," *Journal of Transportation Systems Engineering & Information Technology*, vol. 9, no. 2, pp. 69–74, 2009.

[11] Z. Yao, L. H. Lee, W. Jaruphongsa, V. Tan, and C. F. Hui, "Multi-source facility location-allocation and inventory problem," *European Journal of Operational Research*, vol. 207, no. 2, pp. 750–762, 2010.

[12] M. Wen and R. Kang, "Some optimal models for facility location-allocation problem with random fuzzy demands," *Applied Soft Computing*, vol. 11, no. 1, pp. 1202–1207, 2011.

[13] S. M. Mousavi and S. T. Niaki, "Capacitated location allocation problem with stochastic location and fuzzy demand: a hybrid algorithm," *Applied Mathematical Modelling. Simulation and Computation for Engineering and Environmental Systems*, vol. 37, no. 7, pp. 5109–5119, 2013.

[14] N. Vidyarthi and S. Jayaswal, "Efficient solution of a class of location-allocation problems with stochastic demand and congestion," *Computers & Operations Research*, vol. 48, no. 9, pp. 20–30, 2013.

[15] M. A. Pereira, L. C. Coelho, L. A. N. Lorena, and L. C. D. Souza, "A hybrid method for the Probabilistic Maximal Covering Location-Allocation Problem," *Computers & Operations Research*, vol. 57, pp. 51–59, 2015.

[16] M. Alizadeh, I. Mahdavi, S. Shiripour, and H. Asadi, "A non-linear model for a capacitated location-allocation problem with bernoulli demand using sub-sources," *International Journal of Engineering*, vol. 26, no. 9, pp. 1025–2495, 2012.

[17] M. Alizadeh, I. Mahdavi, N. Mahdavi-Amiri, and S. Shiripour, "A capacitated location-allocation problem with stochastic demands using sub-sources: an empirical study," *Applied Soft Computing Journal*, vol. 34, article 2973, pp. 551–571, 2015.

[18] A. Klose, "An LP-based heuristic for two-stage capacitated facility location problems," *Journal of the Operational Research Society*, vol. 50, no. 2, pp. 157–166, 1999.

[19] A. Marín and B. Pelegrín, "Applying Lagrangian relaxation to the resolution of two-stage location problems," *Annals of Operations Research*, vol. 86, no. 1–6, pp. 179–198, 1999.

[20] Y. K. Liu and X. Zhu, "Capacitated fuzzy two-stage location-allocation problem," *International Journal of Innovative Computing Information & Control Ijicic*, vol. 3, no. 4, pp. 987–999, 2007.

[21] G. A. Ghodsi, *A Lagrangian Relaxation Approach to a Two-Stage Stochastic Facility Location Problem with Second-Stage Activation Cost*, University of Waterloo, Ontario, Waterloo, Canada, 2012.

[22] A. Adibi and J. Razmi, "2-Stage stochastic programming approach for hub location problem under uncertainty: A case study of air network of Iran," *Journal of Air Transport Management*, vol. 47, pp. 172–178, 2015.

[23] J.-P. Arnaout, "Ant colony optimization algorithm for the Euclidean location-allocation problem with unknown number of facilities," *Journal of Intelligent Manufacturing*, vol. 24, no. 1, pp. 45–54, 2013.

[24] N. Megiddo and K. J. Supowit, "On the complexity of some common geometric location problems," *SIAM Journal on Computing*, vol. 13, no. 1, pp. 182–196, 1984.

[25] N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatorica*, vol. 4, no. 4, pp. 373–395, 1984.

[26] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *proceeding of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995.

[27] J. Ciurana, G. Arias, and T. Ozel, "Neural network modeling and particle swarm optimization (PSO) of process parameters in pulsed laser micromachining of hardened AISI H13 steel," *Materials & Manufacturing Processes*, vol. 24, no. 3, pp. 358–368, 2009.

[28] B. Liu, L. Wang, and Y. Jin, "An effective PSO-based memetic algorithm for flow shop scheduling," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 1, pp. 18–27, 2007.

[29] X. Li and J. Wang, "A steganographic method based upon JPEG and particle swarm optimization algorithm," *Information Sciences*, vol. 177, no. 15, pp. 3099–3109, 2007.

[30] M. G. Omran, A. Salman, and A. P. Engelbrecht, "Dynamic clustering using particle swarm optimization with application in image segmentation," *Pattern Analysis and Applications*, vol. 8, no. 4, pp. 332–344, 2006.

[31] N. Holden and A. A. Freitas, "A hybrid PSO/ACO algorithm for discovering classification rules in data mining," *Journal of Artificial Evolution & Applications*, vol. 2008, Article ID 316145, 11 pages, 2008.

[32] R. C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *proceeding of the Transactions on Evolutionary Computation (IEEE '01)*, vol. 1, pp. 81–86.

[33] X. Li and A. P. Engelbrecht, "Particle swarm optimization: an introduction and its recent developments," *the 15th annual Conference Companion on Genetic and Evolutionary Computation*, pp. 33–57, 2007.

[34] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.

[35] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer," in *proceeding of Swarm Intelligence Symposium (sis '05)*, pp. 124–129, 2005.

[36] T. Yamaguchi and K. Yasuda, "Adaptive particle swarm optimization–self-coordinating mechanism with updating information," in *proceeding of the IEEE International Conference on Systems, Man and Cybernetics (SMC '06)*, pp. 2303–2308, twn, October 2006.

[37] W. Liao and J. Wang, "Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization," in *proceeding of the Second international conference on Advances in swarm intelligence (ICSI '11)*, pp. 859–871, Chongqing , China, June 12-15.

[38] D. Zhan, H. Lu, W. Hao, and D. Jin, "Improving particle swarm optimization: Using neighbor heuristic and Gaussian cloud learning," *Intelligent Data Analysis*, vol. 20, no. 1, pp. 167–182, 2016.

[39] L. Deyi, M. Haijun, and S. Xuemei, "Membership cloud and membership cloud generator," *Computer Research & Development*, vol. 32, no. 6, 1995.

[40] Z. Guangwei, H. Rui, L. Yu, L. Deyi, and C. Guisheng, "An evolutionary algorithm based on cloud model," *Chnises Journal of Computers*, vol. 31, no. 7, 2008.

[41] B. Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *proceeding of the 1998 IEEE World Congress on Computational Intelligence*, pp. 69–73, 1998.

[42] A. Carlisle and G. Dozier, "An off-the-shelf PSO," in *proceeding of the Workshop on Particle Swarm Optimization*.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 1063045, 11 pages
https://doi.org/10.1155/2017/1063045

Research Article

An Improved Particle Swarm Optimization Algorithm Using Eagle Strategy for Power Loss Minimization

Hamza Yapıcı[1] and Nurettin Çetinkaya[2]

[1]*The Graduate School of Natural and Applied Science, Selçuk University, Konya, Turkey*
[2]*Electrical & Electronics Engineering Department, Selçuk University, Konya, Turkey*

Correspondence should be addressed to Hamza Yapıcı; hyapici@konya.edu.tr

Received 12 January 2017; Accepted 15 March 2017; Published 30 March 2017

Academic Editor: Blas Galván

The power loss in electrical power systems is an important issue. Many techniques are used to reduce active power losses in a power system where the controlling of reactive power is one of the methods for decreasing the losses in any power system. In this paper, an improved particle swarm optimization algorithm using eagle strategy (ESPSO) is proposed for solving reactive power optimization problem to minimize the power losses. All simulations and numerical analysis have been performed on IEEE 30-bus power system, IEEE 118-bus power system, and a real power distribution subsystem. Moreover, the proposed method is tested on some benchmark functions. Results obtained in this study are compared with commonly used algorithms: particle swarm optimization (PSO) algorithm, genetic algorithm (GA), artificial bee colony (ABC) algorithm, firefly algorithm (FA), differential evolution (DE), and hybrid genetic algorithm with particle swarm optimization (hGAPSO). Results obtained in all simulations and analysis show that the proposed method is superior and more effective compared to the other methods.

1. Introduction

The reactive power optimization approach is important for power quality, system stability, and optimal operation of electrical power systems. Reactive power control can be set with adjusting the voltage levels, tap positions of transformers, shunt capacitors, and other control variables. The reactive power optimization approach can minimize the power losses and improve the voltage profiles. Many conventional methods such as dynamic programming, linear and nonlinear programing, interior point method, genetic algorithm, and quadratic programming have been employed for solving reactive power optimization problem [1–5].

Moreover, in the last years, various intelligence computation methods have proposed for the reactive power optimization such as particle swarm optimization, differential evolution, ant colony, and BC [6–9]. The optimization methods along with fuzzy logic [10] have been used to adjust the optimal setting of power system variables, containing flexible AC transmission systems (FACTs) devices, where the power system losses have been reduced by the optimal placement of thyristor-controlled series compensation (TCSC) and static VAR compensator (SVC). A dynamic weights based particle swarm optimization (PSO) algorithm has been used for reducing power loss [11]. This approach has been implemented to IEEE 6-bus system. The PSO based reactive power optimization method has been presented in [12] for minimizing the total support cost from generators and reactive compensators. In [13], modified artificial fish swarm algorithm (MAFSA) has been proposed to optimize the reactive power optimization and this method has been applied to IEEE 57-bus system. A seeker optimization algorithm has been presented for reactive power dispatch method in [14]; the authors applied the algorithm to several benchmark functions, IEEE 57 and IEEE 118 test systems and then compared with different conventional nonlinear programming methods (such as different versions of GA, DE, and PSO). The optimum conditions for operating the electric power systems have been determined by GA [15]. Authors selected the main objective of the study as the definition of the load buses

voltage amplitude values. By this way, they get the minimum power losses in the transmission lines. In [16], reactive power optimization has been solved by adjusting generator voltages, transformer taps, and capacitors/reactors and three GA/SA/TS hybrid algorithms have been used. In the other study [17], the reactive power optimization problem has been solved by using evolutionary computation techniques such as genetic and particle swarm optimization algorithms, and voltage bus magnitude, transformer tap setting, and the reactive power injected by capacitor banks were selected as control variables. Authors applied the proposed algorithms to IEEE 30-bus and IEEE 118-bus systems. In [18], distributed generation has been taken into distribution system and a multiobjective model for reactive power optimization has been investigated to reduce the system loss and voltage deviation and minimize the total reactive compensation devices capacity. Simulation and analysis have been carried out on IEEE 33-bus system with a dynamically adaptive multiobjective particle swarm optimization (DAMOPSO) algorithm. In [19], an improved mind evolutionary algorithm (IMEA) has been presented for optimal reactive power dispatch and voltage control. The proposed method has been carried out on IEEE 30-bus system and simulation results have been compared with GA to prove its efficiency and superiority. On the other study, GA has been proposed for optimization of reactive power flow in a power system [20]. The authors used two different objective functions: active power losses and the voltage-stability-oriented index.

In this study, a particle swarm optimization algorithm using eagle strategy (ESPSO) has been developed to implement reactive power optimization for reducing power losses. Eagle strategy (ES) has been originated by the foraging behavior of eagles such as golden eagles. This strategy has two important parameters: random search and intensive chase. At first it explores the search space globally, and then in the second case the strategy makes an intensive local search with using an effective local optimizer method [21–23]. So, SPO has been improved using ES and implemented to reactive power optimization problem.

Moreover, simulations and analysis of reactive power optimization problem have been performed on IEEE 30-bus test system, IEEE 118-bus test system, and a real power subsystem, and the proposed approach has been compared with various algorithms to show performance. It can be seen that, in case studies, the proposed approach has been outperformed compared to other methods mentioned.

2. Problem Formulation

This paper defines the objective function as reducing power losses of power system. The objective function given in [14, 15, 17, 24] for reactive power optimization problem is given in (1). The equality and inequality constraints are denoted in (3). Note that the real distribution subsystem considered in this paper has no generators; therefore, we did not handle the generator constraints for this power system. Moreover, control variables are self-constrained but dependent variables are implemented to the objective function by using penalty terms. So, the objective function can be given as in (2) as follows:

$$P_{\text{Loss}} = \sum_{i=1}^{N_i} \left[g_{h(i,j)} \cdot \left(V_i^{\,2} + V_j^{\,2} - 2 \cdot V_i \cdot V_j \cdot \cos\left(\theta_{i,j}\right)\right)\right], \quad (1)$$

$$f\left(\vec{x}, \vec{y}\right) = P_{\text{Loss}} + \lambda_1 \sum_{N_V} \Delta V_L^{\,2} + \lambda_2 \sum_{N_Q} \Delta Q_G^{\,2} + \lambda_3 \sum_{N_i} \Delta I_{l(i,j)}^{\,2}, \quad (2)$$

where λ_1, λ_2, and λ_3 are the penalty factors which are equal to 1000, N_V is the number of load buses, and N_Q is the number of generator buses which injected reactive power. If $V_L < V_{L\min}$ then ΔV_L is equal to $V_{L\min} - V_L$ and if $V_{L\max} < V_L$ then ΔV_L is equal to $V_L - V_{L\max}$. On the other hand, if $Q_G < Q_{G\min}$ then ΔQ_G is equal to $Q_{G\min} - Q_G$ and if $Q_{G\max} < Q_G$ then ΔQ_G is equal to $Q_G - Q_{G\max}$. If $I_{l(i,j)} > I_{l(i,j)\max}$ then $I_{l(i,j)}$ is equal to $I_{l(i,j)\max}$.

Note that superscripts "min" and "max" in (3) express lower and upper limits, respectively. Here, $\vec{x}$ is the vector of control variable where $\vec{x} = \left[V_G \;\; T_K \;\; Q_C\right]^T$ and $\vec{y}$ is the vector of dependent variable where $\vec{y} = \left[V_L \;\; Q_G \;\; I_l\right]^T$.

$$\begin{aligned}
&P_{Gi} - P_{li} - V_i \cdot \sum_{j \in Ni} \left[V_j \cdot \left(g_{h(i,j)} \cdot \cos\left(\theta_{(i,j)}\right) + b_{h(i,j)} \cdot \sin\left(\theta_{(i,j)}\right)\right)\right] = 0, \quad i \in N_{\text{slb}},\\
&Q_{Gi} - Q_{li} - V_i \cdot \sum_{j \in Ni} \left[V_j \cdot \left(g_{h(i,j)} \cdot \sin\left(\theta_{(i,j)}\right) - b_{h(i,j)} \cdot \cos\left(\theta_{(i,j)}\right)\right)\right] = 0, \quad i \in N_{\text{pq}},\\
&T_{K\min} \le T_K \le T_{K\max}, \quad K \in N_K,\\
&Qc_{i\min} \le Qc_i \le Qc_{i\max}, \quad i \in N_C,\\
&P_{Gi\min} \le P_{Gi} \le P_{Gi\max}, \quad i \in N_G,\\
&Q_{Gi\min} \le Q_{Gi} \le Q_{Gi\max}, \quad i \in N_G,\\
&V_{i\min} \le V_i \le V_{i\max}, \quad i \in N_i,\\
&I_{l(i,j)} < I_{l(i,j)\max}, \quad i, j \in N_i.
\end{aligned} \quad (3)$$

3. Method

3.1. Eagle Strategy (ES). Eagle strategy (ES) is a two-stage process, developed by Yang et al. [22]. ES is inspired by the foraging behavior of eagles that they fly random in analogy to the Lévy flights. It uses different algorithms which make global search and local search for fitting different proposes. ES has some similarities with random restart hill climbing method, but there are two important differences: ES is a two-stage method, and so a global search randomization method and an intensive local search are combined and ES uses Lévy walks so it can explore the global search space more effectively.

Load objective function $f(x)$
Initial population (at $k = 0$)
While ($\|$minimum $f(k+1)$ − minimum $f(k)\| \leq$ tolerance or $k >$ *max number of iterations*)
    Random global search (Levy walks)
    Local search by using PSO
    If a better solution is found
        Update the current best
    End if
    $k = k + 1$
end

Pseudocode 1

Load objective function $f(x)$
Generate the initial population and velocity of n particles
Find global best (at $k = 0$)
While ($\|$minimum $f(k+1)$ − minimum $f(k)\| \leq$ tolerance or $k >$ *max number of iterations*)
    Calculate new velocity and position of each particle via (5) and (6)
    Evaluate the new fitness
    If $f(x_i(k+1)) < f(p\text{best}_i(k))$
    $p\text{best}_i(k+1) = x_i(k+1)$
    Update global best
    Update weight
    $k = k + 1$
end

Pseudocode 2

Essentially, ES makes the global search in the n-dimensional space with Lévy flights; if any probable solution is found, an intensive local optimizer is put to use for local search such as differential evolution, particle swarm optimization algorithm, and artificial bee colony that these have local search capability. Then the procedure starts again with new global search in the new area [21–23, 25, 26].

Note that ES is not an algorithm; it is a method. In fact, various algorithms can be used at the different stages. This provides that it combines the advantages of these different algorithms so as to obtain better results.

Lévy distribution [22] is given as follows:

$$L(s) \sim \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}}, \quad (s \gg s_0 > 0),\ (1 < \lambda \leq 3), \tag{4}$$

where $\Gamma(\lambda)$ is standard gamma function and s is the step length. When $\lambda = 3$, it becomes Brownian motion as a special case. We used $\lambda = 2$; thus Lévy walks become the Cauchy distribution.

Pseudocode of ES can be given as in Pseudocode 1.

3.2. Particle Swarm Optimization (PSO). This technique processing is so easy that PSO utilizes some parameters and definitions of the optimization process and then it starts the process with an initial random population, named particles. Each of these particles has a possible solution for the main problem and is processed as a part in n-dimensional space. In the variable space, each particle has a position identified by $x_i^k = (x_{i1}^k, x_{i2}^k, \ldots, x_{in}^k)$ and a velocity identified by $v_i^k = (v_{i1}^k, v_{i2}^k, \ldots, v_{in}^k)$. Velocity and position of each particle is updated using (5) and (6), respectively. If a particle has a best position, it is carried to the next. Additionally, best positions are represented as $p$best and the best position of all particles is represented as $g$best [27–29].

$$v_i^{k+1} = w_i v_i^k + c_1 r_1 \left(p\text{best}_i - x_i^k\right) + c_2 r_2 \left(g\text{best}_i - x_i^k\right), \tag{5}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1}. \tag{6}$$

The weight function is

$$w_i = w_{\max} - \frac{w_{\max} - w_{\min}}{k_{\max}} \times k, \tag{7}$$

where r_1 and r_2 are selected randomly in the range $[0, 1]$ and c_1 and c_2 are acceleration coefficients that inspect the motion of a particle.

Pseudocode of PSO see Pseudocode 2.

3.3. Eagle Strategy with Particle Swarm Optimization (ESPSO). We know that for local search we can use an algorithm such as PSO, DE, and ABC; therefore, PSO is applied to the local

Step 1. Load function and its parameters
Step 2. Generate initial population randomly
Step 3. While $\|\text{minimum}\, f(k+1) - \text{minimum}\, f(k)\| \leq$ tolerance or
$k >$ *max number of iterations*,
performing random global search using Levy Flight $x^{k+1} = x^k + \alpha L(s, \lambda)$,
($\lambda = 1.5$, $\alpha = 1$, and step length s set as $s = 5$)
Then, find a promising solution
Step 4. Determine a random number. Set switching parameter p for controlling
between global search and local search. (We set $p = 0.2$)
If $p <$ rand
switch to local search stage (go to Step 5)
else
switch to global search stage (go to Step 6)
Step 5. In intensive local search stage, search around a promising solution,
Calculate new velocity and position of each particle via (5) and (6),
Then evaluate new fitness (Use the objective function based on Newton–Raphson
power flow for reactive power optimization problem)
If $f(x_i(k+1)) < f(p\text{best}_i(k))$
$p\text{best}_i(k+1) = x_i(k+1)$
Step 6. Update,
$k = k + 1$
Step 7. Stopping criterion,
Maximum number of iterations or a given tolerance (tolerance set as $1.0000e-9$
for reactive power optimization problem)
Step 8. If any criterion is provided, then stop the algorithm else go to Step 3

Algorithm 1

search stage of ES method. On the other side, randomization with Lévy walks can be used in the global search. The proposed method is a population-based algorithm.

We used the parameters of PSO used in most applications [18, 19], where $c_1 = c_2 = 2$, $w_{\min} = 0.4$, and $w_{\max} = 0.9$; then we set $\lambda = 2$. We used 500 iterations for benchmark functions and 100 iterations for reactive power optimization problem. There are two important situations, $\Gamma \to \infty$ and $\Gamma \to 0$. If $\Gamma \to \infty$, the velocity of particles cannot be decreased and particles are far from one another. If $\Gamma \to 0$, then the particles are short sighted, so particles will be trapped in a confined space and velocity of this particles can be very small.

There are several stopping criteria given in the literature: a fixed number of generations, the number of iterations since the last change of the best solution being greater than a specified number, the number of iterations reaching maximum number, a located string with a certain value, and no change in the average fitness after some generations. In this paper, the stopping criteria are chosen as the maximum number of iterations and the tolerance value for fitness where $\|\text{minimum}\, f(k+1) - \text{minimum}\, f(k)\| \leq$ tolerance.

Steps of proposed method have been explained in Algorithm 1.

4. Performance of ESPSO on Benchmark Functions

For testing proposed method, seven well-known benchmark functions are handled with comparison of widely used six algorithms (GA, PSO, ABC, FA, and hGAPSO). These benchmark functions are listed in Table 1.

All the methods run in 500 iterations and over 50 times for each function. The results obtained by proposed ESPSO algorithm on some benchmark functions are statistically different from the other algorithms and ESPSO has a good performance on the test functions. Results point out that ESPSO algorithm is appropriate for optimizations of unimodal and multimodal functions. Table 2 shows the results obtained by all methods mentioned in this section. In Table 2, it can be seen that the proposed approach has obtained better results than the other methods.

Unlike other algorithms, ESPSO performs global search and local search by using parameter p until converging to optimum as possible. In addition, it generates initial population randomly using uniform distribution function in MATLAB. Consequently, from the local search to global search, fitness value is changed from the worst to the best, so as a generation can not only find a good search but also switch to new search area in the search space.

5. Implementing Reactive Power Optimization

In reactive power optimization problem, transformer tap positions and shunt capacitor banks are discrete or integer variables. However, the proposed method only uses continuous variable. To use these variables, each particle of ESPSO explores in the space as searching for continuous variable, and objective function is evaluated by cutting the corresponding dimensions of particles into integers. This means that the

Table 1: Benchmark functions.

Equation	Name	D	Feasible bounds
$f_1 = \sum_{i=1}^{D} x_i^2$	Sphere/parabola	30	$(-100, 100)^D$
$f_2 = -20 \exp\left\{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right\} - \exp\left\{\frac{1}{D}\sum_{i=1}^{D} \cos(2\pi x_i)\right\} + 20 + e$	Ackley	30	$(-32, 32)^D$
$f_3 = \sum_{i=1}^{D-1} \left\{100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right\}$	Rosenbrock	30	$(-30, 30)^D$
$f_4 = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	Generalized Griewank	30	$(-600, 600)^D$
$f_5 = -\sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right]$	Hartman 3	3	$(0, 1)^D$
$f_6 = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	Six-hump camel-back	2	$(-5, 5)^D$
$f_7 = \left\{1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right\} \times \left\{30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right\}$	Goldstein-Price	2	$(-2, 2)^D$

Table 2: Comparison of ESPSO with other methods on test functions.

Function	Index	GA	PSO	ABC	FA	hGAPSO	ESPSO
f_1	Best	1.1091	0.9206	0.8991	0.0037	0.8874	0.0026
	Mean	1.7623	1.0038	1.2978	0.0126	1.1172	0.9032
f_2	Best	0.9398	0.7646	0.8991	0.0317	0.0918	0.0125
	Mean	1.6693	1.6237	2.6842	0.1004	1.9044	1.0005
f_3	Best	101.9834	24.0076	38.5673	17.2783	22.9814	16.5622
	Mean	217.0324	65.3482	94.0345	19.2953	28.0912	54.8731
f_4	Best	0.1983	0.0078	0.0659	$9.9047e-04$	$1.1092e-3$	$3.4597e-4$
	Mean	0.9219	0.0396	0.15415	0.0117	0.0335	0.1931
f_5	Best	−3.8664	−3.8642	−3.8652	−3.8628	−3.8629	−3.8628
	Mean	−3.8824	−3.8659	−3.8739	−3.8656	−3.8728	−3.8638
f_6	Best	−1.0525	−1.0393	−1.0353	−1.0317	−1.0318	−1.0316
	Mean	−1.1316	−1.0626	−1.0445	−1.0319	−1.0336	−1.0330
f_7	Best	3.1870	3.0782	3.0390	3.0256	3.0280	3.0130
	Mean	3.7683	3.5900	3.4802	3.3380	3.4517	3.2300

real position values of particles consist of values of capacitor banks and transformers tap positions represented as a vector to be used for calculating objective function.

Here, inequality constraints are handled in iterations as follows: if variables violate their limit, they are clamped to their upper limit and the remaining mismatch is taken by another one not on limit based on their inertia, and during computation if any PV bus reactive power is violated then PV bus is assumed as PQ bus fixing at the brink value.

Furthermore, the procedure of solving reactive power problem can be explained as follows.

The proposed approach of ESPSO loads parameters of power system and initial condition and specifies upper and lower limits. Initial particles are determined randomly via uniform distribution to locate their initial positions, and initial velocities of particles are constituted. The algorithm evaluates the fitness using objective function and adjusts local bests and global best from locals.

While stopping criteria are provided, ESPSO performs that searching global best by Lévy flight and then determining a random number. If this determined random number is less then p parameter, local search would be performed and locals would be updated via local search property of PSO (update velocities and positions of particles). Then, new fitness computed and if fitness found in local search stage is less then fitness found in first stage, global solution and positions of particles would be updated: ($f(k+1) \leftarrow f(k)$, $p\text{bests}(k+1) \leftarrow x(k)$, and then $k = k + 1$).

6. Results of Simulations

To verify the ability, capability, and performance of the proposed ESPSO on reactive power optimization problem, it is implemented to IEEE 30-bus test system, IEEE 118-bus test system, and a real distribution subsystem. For testing and

Table 3: Limits of variables (pu).

Bus	1	2	5	8	11	13
$P_{G\min}$	0.500	0.200	0.150	0.100	0.100	0.120
$P_{G\max}$	2.000	0.800	0.500	0.350	0.300	0.400
$Q_{G\min}$	−0.200	−0.200	−0.150	−0.150	−0.100	−0.150
$Q_{G\max}$	2.000	1.000	0.800	0.600	0.500	0.600
Variable	$V_{\min}$	$V_{\max}$	$T_{K\min}$	$T_{K\max}$	$Q_{C\min}$	$Q_{C\max}$
Limit	0.90	1.10	0.90	1.10	0.00	0.50

Table 4: Control variables values and power loss (control variables given pu).

Variables	GA	PSO	ABC	FA	DE	hGAPSO	ESPSO
V_1	1.0345	1.0606	1.0927	1.1000	1.0500	1.0300	1.0770
V_2	1.0463	1.0524	1.0880	1.0967	1.0446	1.0400	1.0775
V_5	1.0294	1.0284	1.0695	1.0850	1.0247	1.0400	1.0700
V_8	1.0283	1.0289	1.0722	1.0895	1.0265	1.0200	1.0700
V_{11}	1.0517	0.9833	1.0860	1.0930	1.1000	1.0300	1.0800
V_{13}	1.0214	0.9924	1.0926	1.0969	1.1000	0.9500	1.0810
$T_{K(6-9)}$	1.0314	1.0530	0.9983	1.0478	1.0000	1.0300	1.0050
$T_{K(6-10)}$	1.0994	1.1000	0.9994	0.9439	1.1000	1.0800	1.0050
$T_{K(4-12)}$	1.1000	1.0745	0.9984	1.0318	1.0800	1.7000	0.9800
$T_{K(28\text{-}27)}$	0.9071	0.9247	1.0034	1.0044	0.9200	1.0400	0.9720
Q_{C10}	0.0153	0.1174	0.0155	0.0534	0.2600	—	0.2050
Q_{C24}	0.0063	0.0056	0.0371	0.0663	0.1000	—	0.1170
P_{loss} (MW)	4.78	4.39	3.09	4.71	5.01	3.69	3.079

proving the performance of proposed method, it is compared with various algorithms such as GA, PSO [17], ABC [24], FA [30], DE [31], and hGAPSO [32].

The population-based algorithms GA, PSO, DE, and their improved version have a great interest in engineering optimization problems and they all have been successfully implemented to reactive power optimization problems [4, 6, 7, 11, 12, 16, 17, 31, 32]. In some other studies, ABC and FA [24, 30] have also been used for reactive power optimization. So, ESPSO is compared with these algorithms.

In [17], authors proposed GA and PSO to solve reactive power optimization problem for some test systems where, on IEEE 30-bus system, they obtained power losses 4.78 MW and 4.39 MW with GA and PSO, respectively. In [24], authors obtained power losses with ABC in the study for 30-bus system as 3.09 MW. In a different study for IEEE 30-bus power system, power loss has been tabulated as 4.71 MW with FA [30]. Authors considered four different test systems with various algorithm and they obtained power losses for 30-bus system as 5.01 MW with DE [31]. In [32] minimum power losses have been tabulated as 3.69 MW for IEEE 30-bus test system. (Note that data and settings of all methods can be found in these studies.)

The results of studies in [17] with GA and PSO, [24] with ABC, [31] with DE, and [32] with hGAPSO are taken into consideration for IEEE 118-bus test system. In these studies, authors obtained power losses with GA, PSO, ABC, and DE as 139.16, 135.64, 119.6923, and 128.318, respectively.

The proposed method is coded in MATLAB software for 30-bus and 118-bus test systems. Maximum number of iterations is set as 100 and ESPSO is run over 50 times. The best solutions for both systems are tabulated after 50 times running.

6.1. IEEE 30-Bus Test System. Power system includes 41 branches, 6 generators, 21 load buses, and 9 shunt compensators. Branches of 4–12, 6–9, 6–10, and 28-27 are adjustable tab under load and symbolized as $T_{K(4-12)}$, $T_{K(6-9)}$, $T_{K(6-10)}$, and $T_{K(28\text{-}27)}$, respectively. All data and parameters of power system used can be found in [33]. The system loads are as follows: $P_{\text{load}} = 2.834$ pu and $Q_{\text{load}} = 1.262$ pu. The transformers tap settings and bus voltages lower and upper limits are 0.9 pu and 1.1 pu and shunt compensators limits are 0.0 pu and 0.05 pu, respectively, that variables limits are denoted in Table 3.

For comparing the proposed approach with other algorithms mentioned previously, index related to performance consists of the minimization of power loss given in Table 4. Table 4 shows that ESPSO found system loss less than other techniques, so that it can be deduced that the proposed method is clearly more robust and better than others in this case study. Variables values obtained by solving reactive power optimization problem for IEEE 30-bus system are also illustrated in Table 4. The power loss is reduced to 3.0128 MW with ESPSO, which is less than that obtained by the other methods. On the other hand, the bus voltages are kept

Table 5: Generators powers (MW).

Variables	ESPSO
P_{G1}	51.50
P_{G2}	80.00
P_{G5}	50.00
P_{G8}	35.00
P_{G11}	30.00
P_{G13}	39.98

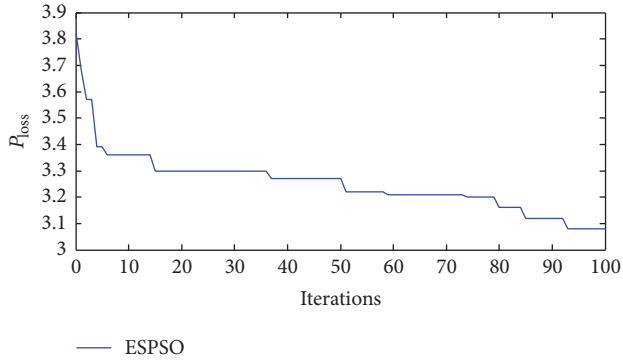


Figure 1: The convergence curve of power loss (MW) for 30-bus power system.

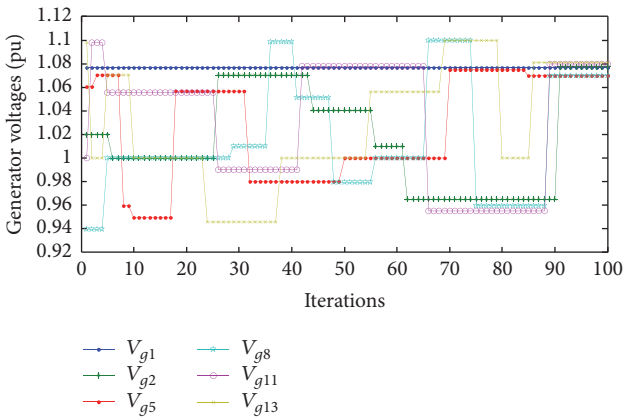


Figure 2: The variation of generator voltages for IEEE 30-bus power system.

within their limits; it shows that the proposed method has a best performance for optimizing voltage quality. Outputs of generators are denoted in Table 5 and the variation of power loss obtained by proposed approach is also given in Figure 1. Furthermore, the variation graphs of control variables are illustrated in Figures 2–4.

6.2. IEEE 118-Bus Test System. The IEEE 118-bus system data are given in [34]. This power system contains 54 generators, 9 transformers, and 186 lines. The total load of system is 4242 MW. In this study, tap settings of the transformers and the voltage limits are taken into consideration within 0.9 pu–1.1 pu and 0.94 pu–1.06 pu, respectively.

The results obtained by proposed method are given in Table 6. To compare the performance of proposed method,

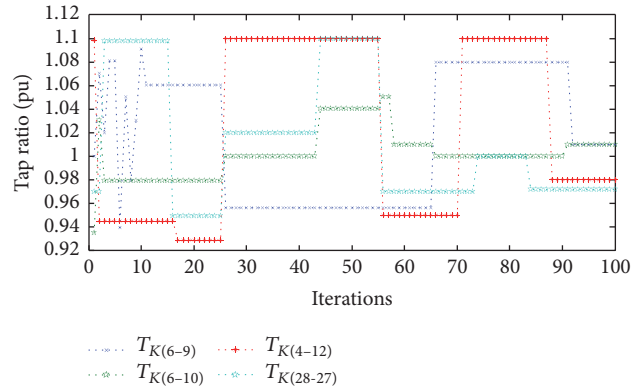


Figure 3: The variation of tap ratios for IEEE 30-bus power system.

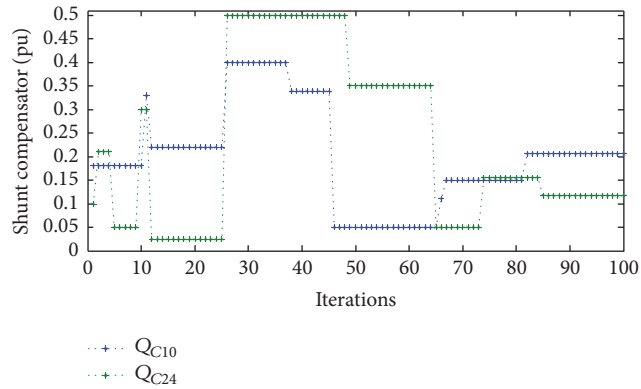


Figure 4: The variation of shunt compensators for IEEE 30-bus power system.

the minimum power loss obtained by different methods is also tabulated. The power loss has been reduced to 119.5500 MW. Table 6 illustrates that in this section the proposed method is more robust and effective than the other methods. Moreover, the convergence of power loss for IEEE 118-bus power system is given in Figure 5. Table 7 and Figure 6 show the tap ratios, the values of shunt compensators, the voltage, and power of generators, respectively.

6.3. Case Study on a Real Distribution Subsystem. The proposed method has been used to minimize the power loss by using reactive power optimization on Eregli Distribution Subsystem in Turkey. All data and parameter have been supplied from Eregli Branch Office of Meram Electricity Distribution Corporation.

The power subsystem considered in this paper includes 1311 buses at 0.4 kV, 9 buses at 31.5 kV, and 3 buses at 15.8 kV and 12 branches with adjustable tab under load. First bus is selected as the slack bus, 2–9 buses are PV buses, and others are selected as PQ buses (switching capacitor banks are located on all 0.4 kV buses). In this respect, vector dimensions become 1333 dimensions composed of nine voltage magnitude values, 12 tab settings, and 1311 switchable capacitor banks. The discrete control variables given in Table 8 are changed with step of 0.01 per unit (pu). The active power

Table 6: Comparison of the power loss for IEEE 118-bus test system.

Variables	P_{loss} (MW)
GA	139.16
PSO	135.64
ABC	119.6923
FA	—
DE	128.318
hGAPSO	—
ESPSO	119.5500

Table 7: Tap ratio.

Variables	ESPSO
$T_{K(8-5)}$	0.9850
$T_{K(26-25)}$	0.9900
$T_{K(30-17)}$	0.9910
$T_{K(38-37)}$	0.9850
$T_{K(63-59)}$	0.9500
$T_{K(64-61)}$	1.0000
$T_{K(65-66)}$	0.9950
$T_{K(68-69)}$	0.9450
$T_{K(81-80)}$	1.0050

load and reactive power load are 0.9869 pu and 0.2242 pu, respectively, and total power losses are 1.0961 MW (note that pu values are computed on 100 MVA base).

For comparing the proposed method with GA, PSO, ABC, FA, and hGAPSO, the performance index including minimum active power losses is illustrated in Figure 7. All algorithms are run in 100 iterations and over 50 times. The results tabulated by all algorithms have been obtained after 50 times running. In addition, ESPSO has found the minimum power loss as 0.8973 MW, and all results are given in Table 9 (values of capacitor banks have not been given because of containing too many items). It can be seen that ESPSO is more effective and superior compared to other algorithms.

In order to clarify the ability of ESPSO that is statistically more robust and better than other methods, it can be shown that the minimum power loss obtained by ESPSO is smaller than all others listed and it conducts global searches accurately. After 50 times running it can be seen that the power losses are decreased from 1.0961 MW to 0.8973 MW by the proposed ESPSO which is the highest power loss reduction. The results obtained by all algorithms handled in this paper are given below, which indicates minimum power loss, voltage magnitude of buses, and tap ratios.

7. Conclusion

Eagle strategy is a method of combination of global search and intensive local search for optimization. In this paper, an approach, ES with PSO based reactive power optimization method, has been implemented to a reactive power optimization problem for minimizing the power losses of 30-bus test system, 118-bus test system, and a real distribution subsystem.

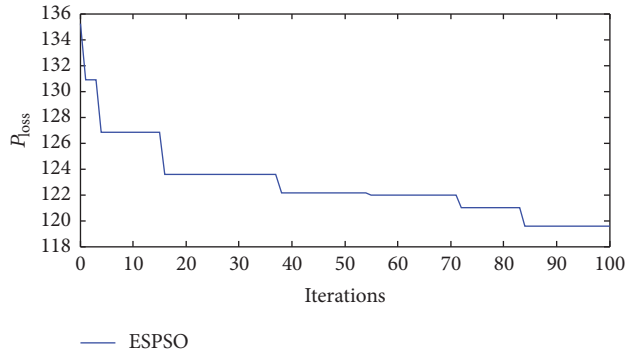


Figure 5: The convergence curve of power loss (MW) for 118-bus power system.

Table 8: The limits of control variables.

Parameter	Min. val. (pu)	Max. val. (pu)
Voltage	0.9	1.1
Capacitor	0.1	0.61
Tap position	0.9	1.1

It can be seen that two case studies in this paper which are performance test on benchmark functions and the minimization of power losses of various power systems may help to clarify the capability of proposed approach for optimization problem. Furthermore, ESPSO is effectively solving optimization problems and finding the optimum more successfully than the other algorithms. On the other hand, about benchmark functions performance, the proposed approach has a better performance than all other algorithms. So, it can be seen that the proposed ESPSO algorithm is influential and able to solve power loss minimization problem and may become a good candidate for other optimization problems such as reactive power dispatch, cost minimization, or multiobjective optimization.

Nomenclature

P_{loss}: Total power losses
$g_{h(i,j)}$: Conductance between nodes i and j
$b_{h(i,j)}$: Admittance between nodes i and j
$\theta_{i,j}$: Angle difference between nodes i and j
N_i: Number of power system buses
N_{slb}: Number of slack buses
N_{pq}: Number of PQ buses
N_K: Number of transformer branches
N_C: Number of capacitor installed buses
N_G: Number of generator buses
T_K: Tap position of transformers at branches K
Q_C: Shunt capacitor value
V_i: Voltage magnitude of bus i
V_j: Voltage magnitude of bus j
V_G: Voltage magnitude of each generator
V_L: Voltage magnitude of each bus
Q_G: The injected reactive power
P_G: The injected active power

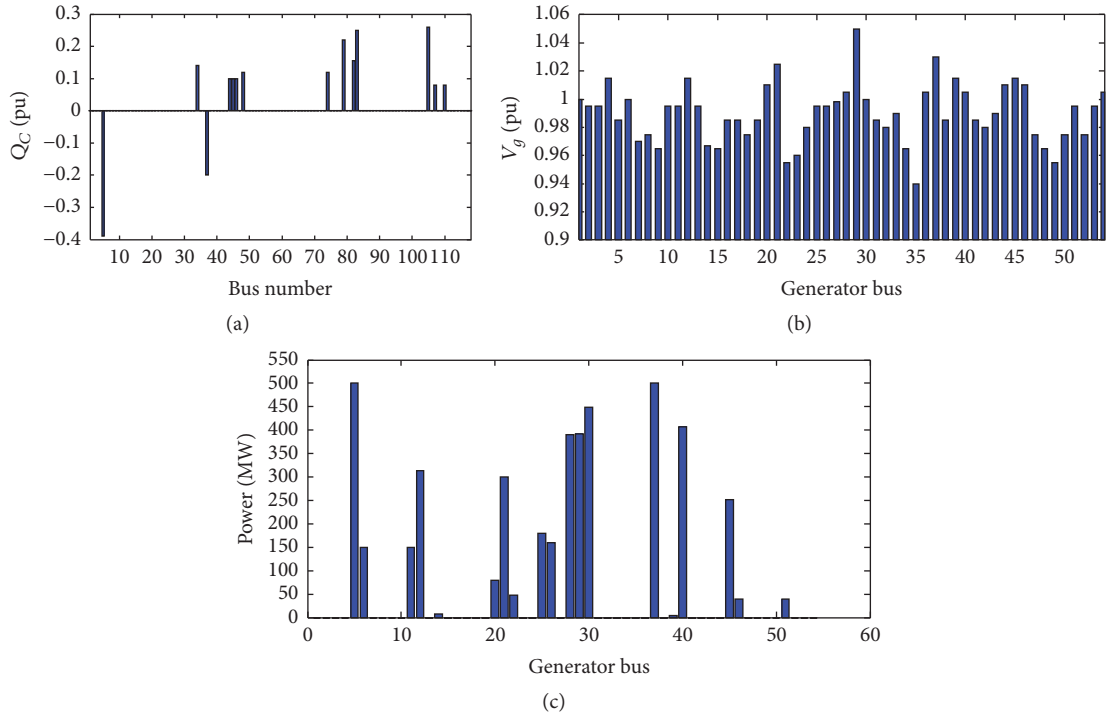


Figure 6: Variables of IEEE 118-bus power system: (a) value of shunt compensator (pu), (b) voltages of generators (pu), and (c) power of generators (MW).

Table 9: Control variables values and power loss (control variables given pu).

Variables	GA	PSO	ABC	FA	DE	hGAPSO	ESPSO
V_1	1.0100	1.0200	1.0400	1.0600	1.0600	1.0600	1.0600
V_2	1.0300	1.0400	1.0200	1.0100	0.9900	1.0300	1.0500
V_3	0.9800	1.0200	1.0400	1.0100	1.0600	1.0600	1.0300
V_4	1.0400	1.0300	1.0200	1.0400	0.9800	1.0300	1.0500
V_5	0.9900	1.0600	1.0400	0.9900	0.9500	1.0600	1.0200
V_6	0.9600	1.0600	1.0200	1.0500	1.0100	1.0300	1.0100
V_7	1.0600	1.0600	1.0400	1.0600	1.0100	1.0600	1.0100
V_8	1.0600	1.0500	1.0200	1.0500	0.9700	1.0300	1.0400
V_9	1.0200	1.0400	1.0000	1.0000	0.9900	1.0000	1.0500
T_1	1.07	1.03	1	0.98	1.05	1	1
T_2	1.05	1	1	0.98	1.04	0.97	0.98
T_3	1	1.06	1	1.04	1.05	0.97	1.04
T_4	1.01	1	1	1.01	1	0.98	1.01
T_5	1.03	1	1.05	0.97	0.98	1.01	1
T_6	1	1	1.03	1.01	0.98	1.05	1
T_7	0.98	0.97	1.03	0.99	0.97	1.04	0.98
T_8	0.96	0.98	0.97	1.03	1	0.98	1.03
T_9	1	1	0.98	0.98	0.98	1	0.99
T_{10}	0.98	0.98	0.97	1.04	1.04	1	1
T_{11}	0.97	0.96	1.06	1	1.05	1.01	1
T_{12}	0.99	0.07	1.05	1	1.01	1.01	1
P_{loss} (MW)	0.91576	0.91189	0.91758	0.90622	0.91760	0.90922	0.89730

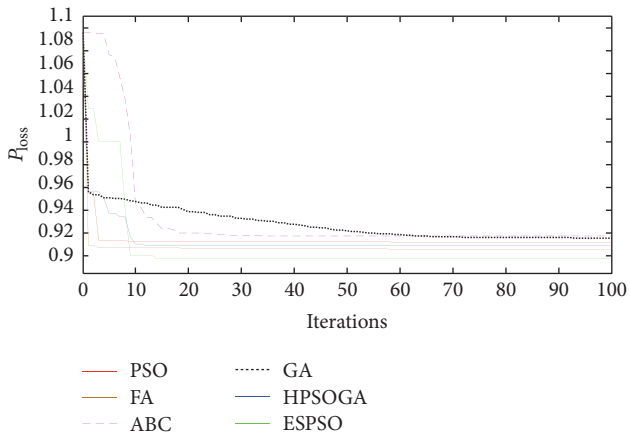


Figure 7: Convergence of active power loss (MW).

Q_l: The reactive power demanded by load
P_l: The active power demanded by load
$I_{l(i,j)}$: Transmission line current flow capacity at line between nodes i and j
$L(s)$: Lévy distribution function
$\Gamma(\lambda)$: Standard gamma function
λ: Gamma function parameter
s_0: The initial step length value
s: Step length
k: Number of iterations
v: Velocity of a particle in PSO
x: Position of a particle in PSO
$p$best: Local best position
$g$best: Global best
w: The weight function
r_1, r_2: Selected randomly in the range [0, 1]
c_1, c_2: Acceleration coefficients
p: Control parameter of switching
f: Objective function.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] F.-C. Lu and Y.-Y. Hsu, "Reactive power/voltage control in a distribution substation using dynamic programming," *IEE Proceedings: Generation, Transmission and Distribution*, vol. 142, no. 6, pp. 639–645, 1995.

[2] O. Alsaç, J. Bright, M. Prais, and B. Stott, "Further developments in LP-based optimal power flow," *IEEE Transactions on Power Systems*, vol. 5, no. 3, pp. 697–711, 1990.

[3] S. Granville, "Optimal reactive dispatch through interior point methods," *IEEE Transactions on Power Systems*, vol. 9, no. 1, pp. 136–146, 1994.

[4] K. Iba, "Reactive power optimization by genetic algorithm," *IEEE Transactions on Power Systems*, vol. 9, no. 2, pp. 685–692, 1994.

[5] N. Grudinin, "Reactive power optimization using successive quadratic programming method," *IEEE Transactions on Power Systems*, vol. 13, no. 4, pp. 1219–1225, 1998.

[6] E. Wu, Y. Huang, and D. Li, "An adaptive particle swarm optimization algorithm for reactive power optimization in power system," in *Proceedings of the 8th World Congress on Intelligent Control and Automation (WCICA '10)*, pp. 3132–3137, Jinan, China, July 2010.

[7] X. Zhang, W. Chen, C. Dai, and A. Guo, "Self-adaptive differential evolution algorithm for reactive power optimization," in *Proceedings of the 4th International Conference on Natural Computation (ICNC '08)*, pp. 560–564, Jinan, China, October 2008.

[8] G. Lirui, H. Limin, Z. Liguo, L. Weina, and H. Jie, "Reactive power optimization for distribution systems based on dual population ant colony optimization," in *Proceedings of the 27th Chinese Control Conference (CCC '08)*, pp. 89–93, IEEE, Kunming, China, July 2008.

[9] H. Wei, Z. Cong, Y. Jingyan et al., "Using bacterial chemotaxis method for reactive power optimization," in *Proceedings of the Transmission and Distribution Exposition Conference: IEEE PES Powering Toward the Future (PIMS '08)*, April 2008.

[10] B. Bhattacharyya and V. K. Gupta, "Fuzzy based evolutionary algorithm for reactive power optimization with FACTS devices," *International Journal of Electrical Power and Energy Systems*, vol. 61, pp. 39–47, 2014.

[11] A. Q. H. Badar, B. S. Umre, and A. S. Junghare, "Reactive power control using dynamic particle swarm optimization for real power loss minimization," *International Journal of Electrical Power and Energy Systems*, vol. 41, no. 1, pp. 133–136, 2012.

[12] P. R. Sujin, T. R. D. Prakash, and M. M. Linda, "Particle swarm optimization based reactive power optimization," *Journal of Computing*, vol. 2, pp. 73–78, 2010.

[13] S. Liu, Y. Han, Y. Ouyang, and Q. Li, "Multi-objective reactive power optimization by Modified Artificial Fish Swarm Algorithm in IEEE 57-bus power system," in *Proceedings of the 6th IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC '14)*, pp. 1–5, IEEE, Hong Kong, December 2014.

[14] C. Dai, W. Chen, Y. Zhu, and X. Zhang, "Seeker optimization algorithm for optimal reactive power dispatch," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1218–1231, 2009.

[15] A. Ozturk and S. Duman, "Determination of the optimal operating conditions in power systems using genetic algorithm," *Gazi University Engineering and Architecture Faculty Journal*, vol. 24, pp. 539–548, 2009.

[16] Y. Liu, L. Ma, and J. Zhang, "GA/SA/TS hybrid algorithms for reactive power optimization," in *Proceedings of the Power Engineering Society Summer Meeting*, pp. 245–249, Washington, DC, USA, July 2000.

[17] S. Durairaj and B. Fox, "Evolutionary computation based reactive power optimization," in *Proceedings of the IET-UK International Conference on Information and Communication Technology in Electrical Sciences (ICTES '007)*, pp. 120–125, Dr. M.G.R. University, Chennai, India, December 2007.

[18] S. Cheng and M.-Y. Chen, "Multi-objective reactive power optimization strategy for distribution system with penetration of distributed generation," *International Journal of Electrical Power and Energy Systems*, vol. 62, pp. 221–228, 2014.

[19] Q. Liu and J. Yue, "An improved mind evolutionary algorithm for reactive power optimization," in *Proceedings of the 32nd Chinese Control Conference (CCC '13)*, pp. 8048–8051, Xi'an, China, July 2013.

[20] R. Lukomski and K. Wilkosz, "Optimization of reactive power flow in a power system for different criteria: stability problems,"

in *Proceedings of the 8th International Symposium on Advanced Topics in Electrical Engineering (ATEE '13)*, pp. 1–6, Bucharest, Romania, May 2013.

[21] X.-S. Yang, S. Deb, and S. Fong, "Metaheuristic algorithms: optimal balance of intensification and diversification," *Applied Mathematics and Information Sciences*, vol. 8, no. 3, pp. 977–983, 2014.

[22] X.-S. Yang, S. Deb, and X. He, "Eagle strategy with flower algorithm," in *Proceedings of the 2nd International Conference on Advances in Computing, Communications and Informatics (ICACCI '13)*, pp. 1213–1217, IEEE, Mysore, India, August 2013.

[23] R. Jia and D. He, "Artificial bee colony algorithm with two-stage eagle strategy," in *Proceedings of the 9th International Conference on Computational Intelligence and Security (CIS '13)*, pp. 16–20, IEEE, Leshan, China, December 2013.

[24] K. Ayan and U. Kılıç, "Artificial bee colony algorithm solution for optimal reactive power flow," *Applied Soft Computing*, vol. 12, no. 5, pp. 1477–1482, 2012.

[25] A. H. Gandomi, X.-S. Yang, S. Talatahari, and S. Deb, "Coupled eagle strategy and differential evolution for unconstrained and constrained global optimization," *Computers and Mathematics with Applications*, vol. 63, no. 1, pp. 191–200, 2012.

[26] X. S. Yang, S. Deb et al., "Eagle strategy using Lévy walk and firefly algorithms for stochastic optimization," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, J. R. Gonzalez, D. A. Pelta, C. Cruz, G. Terrazas, and N. Krasnogor, Eds., vol. 284, pp. 101–111, Springer, 2010.

[27] W. Zhang and Y. Liu, "Reactive power optimization based on PSO in a practical power system," in *Proceedings of the IEEE Power Engineering Society General Meeting*, pp. 239–243, June 2004.

[28] N. K. Sharma, D. Suresh Babu, and S. C. Choube, "Application of particle swarm optimization technique for reactive power optimization," in *Proceedings of the International Conference on Advances in Engineering, Science and Management (ICAESM '12)*, pp. 88–93, IEEE, Nagapattinam, India, March 2012.

[29] W. Zhang and Y. T. Liu, "Multi-objective reactive power and voltage control based on fuzzy optimization strategy and fuzzy adaptive particle swarm," *International Journal of Electrical Power and Energy Systems*, vol. 30, no. 9, pp. 525–532, 2008.

[30] H. Deenadhayalan, "Real power loss minimization using firefly algorithm," *International Journal of Advanced Information and Communication Technology*, vol. 1, no. 8, pp. 677–682, 2014.

[31] M. Varadarajan and K. S. Swarup, "Differential evolutionary algorithm for optimal reactive power dispatch," *International Journal of Electrical Power and Energy Systems*, vol. 30, no. 8, pp. 435–441, 2008.

[32] K. Lenin, B. R. Reddy, and M. S. Kalavathi, "A new improved GA and PSO combined hybrid algorithm (HIGAPSO) for solving optimal reactive power dispatch problem," *Journal of Industrial and Intelligent Information*, vol. 2, no. 3, pp. 205–209, 2014.

[33] K. Y. Lee, Y. M. Park, and J. L. Ortiz, "A united approach to optimal real and reactive power dispatch," *IEEE Transactions on Power Apparatus and Systems*, vol. 104, no. 5, pp. 1147–1153, 1985.

[34] Power system data for IEEE 118-bus test system, 2014, http://www.ece.ubc.ca/~hameda/download_files/case118.m.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 1048081, 13 pages
https://doi.org/10.1155/2017/1048081

Research Article

Optimizing Production Scheduling of Steel Plate Hot Rolling for Economic Load Dispatch under Time-of-Use Electricity Pricing

Mao Tan,[1] Hua-li Yang,[1] Bin Duan,[1] Yong-xin Su,[1] and Feng He[2]

[1]*Hunan Province Cooperative Innovation Center for Wind Power Equipment and Energy Conversion, College of Information Engineering, Xiangtan University, Xiangtan, China*
[2]*Department of Energy and Environmental Protection, Xiangtan Iron and Steel Corporation Ltd., Xiangtan, China*

Correspondence should be addressed to Mao Tan; mr.tanmao@gmail.com

Received 14 November 2016; Accepted 15 February 2017; Published 7 March 2017

Academic Editor: Domenico Quagliarella

Time-of-Use (TOU) electricity pricing provides an opportunity for industrial users to cut electricity costs. Although many methods for Economic Load Dispatch (ELD) under TOU pricing in continuous industrial processing have been proposed, there are still difficulties in batch-type processing, since power load units are not directly adjustable and nonlinearly depend on production planning and scheduling. In this paper, for hot rolling, a typical batch-type and energy intensive process in steel industry, a production scheduling optimization model for ELD is proposed under TOU pricing, in which the objective is to minimize electricity costs while considering penalties caused by jumps between adjacent slabs. A NSGA-II based multiobjective production scheduling algorithm is developed to obtain Pareto optimal solutions, and then TOPSIS based multicriteria decision-making is performed to recommend an optimal solution to facilitate field operation. Experimental results and analyses show that the proposed method cuts electricity costs in production, especially in case of allowance for penalty score increase in a certain range. Further analyses show that the proposed method has effect on peak load regulation of power grid.

1. Introduction

Time-of-Use (TOU) electricity pricing, a practical demand response program implemented by many power suppliers to improve the peak load regulation ability of power grid, provides an opportunity for electricity users to implement Economic Load Dispatch (ELD), that is, cut electricity costs by reducing power loads during on-peak periods and shifting loads from on-peak to off-peak periods.

Unlike conventional energy conservation to reduce absolute energy consumption, optimizing electricity costs under TOU pricing means that industrial users adjust their production schedule to avoid on-peak time periods, which will have significant effect on cutting electricity costs. In recent years, ELD under TOU pricing has become a hot area. Shrouf et al. [1] proposed a single machine scheduling problem, in which each time period has an associated price and the objective is to minimize electricity costs while considering traditional scheduling performance measures. Fang et al. [2] also considered job scheduling on a single machine to minimize total electricity costs under TOU pricing and proposed the algorithms for uniform-speed and speed-scalable machine environments, respectively. Mitra et al. [3] formulated mixed integer linear programming for continuous industrial processing, which allows optimal production planning, and provided a case study for time horizon of one week and hourly changing electricity prices. Furthermore, they improved the model with integration of operational and strategic decision-making [4]. Ashok [5] presented a theoretical model for batch-type load processing and proposed an integer programming method to reschedule their operations to reduce electricity costs under time-varying electricity price, but the model is an abstract theoretical model and is difficult to be applied to production directly. Wang et al. [6] proposed an optimization model to minimize electricity costs for steel plant, in which both power generation scheduling and batch production scheduling were considered; although the model has been believed to be effective under TOU pricing, the

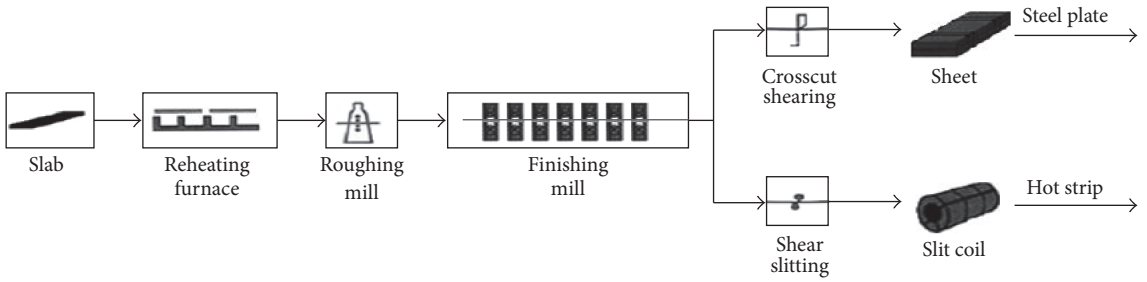


Figure 1: A process flow diagram of the hot rolling production procedure.

results cannot always be optimal because the production load units are determined by fixed production planning and scheduling.

The above analyses motivate the potential for more benefits by ELD under TOU pricing in hot rolling production scheduling. Until now, most of the related literatures focused on specific part of the problem or the abstract simplified problem; thus there are still difficulties, since the power load units are not directly adjustable and nonlinearly depend on the results of production planning and scheduling.

Hot rolling, a typical batch-type and energy intensive process in steel production with characteristics of strong schedulability, has become an important aspect of production organization and energy saving [7]. The general process flow of hot rolling production is illustrated in Figure 1. Hot rolling is mainly organized and carried out by batch scheduling program in steel mill, the primary task of which is arranging and sequencing slabs into rolling units to smooth jumps in width, gauge, and hardness between adjacent slabs; all of these will directly affect product quality. Hot rolling production scheduling has attracted attention from academia and industry for a long time. An early method proposed by Kosiba et al. treated steel production scheduling as a discrete event sequencing problem and thus formulated it as a traveling salesman problem [8]. Lopez et al. [9] formulated the problem as a generalized prize collecting traveling salesman problem with multiple conflicting objectives and constraints and proposed a heuristic tabu search method to determine good approximate solutions. Tang and Wang [10] proposed a modified genetic algorithm based on the multiple travelling salesman problem. Chen et al. [11] formulated the problem as a nonlinear integer programming model, and later it is corrected by Kim [12] and changed to a linear programming model. Furthermore, Alidaee and Wang [13] proposed a corrected integer programming formulation and reduced the quantity of variables. Nevertheless, most of proposed models are single objective or transformed models based on weighted-sum approach. Jia et al. [14] formulated the problem as a multiobjective vehicle routing problem with double time windows and proposed a decomposition-based hierarchical optimization algorithm to solve it. Soon after, they proposed a P-MMAS algorithm to solve the problem; multicriteria decision-making is performed to recommend the optimal solution from the Pareto frontier [15]. Moon et al. [16] proposed a production scheduling model with time-dependent and machine-dependent electricity cost, in which makespan was considered by using the weighted-sum objective but batch sizing was not considered, which is obviously simpler than batch scheduling problem. Because of complexity of batch sizing problem, Sarakhsi et al. [17] proposed a hybrid algorithm of scatter search and Nelder-Mead algorithms to improve the performance of solving algorithm.

Due to high energy consumption and rising energy costs in hot rolling production [18], energy saving has also been considered combined with the traditional objective mentioned above. As is shown in Figure 1, slabs are heated to high temperature before being rolled; the total energy consumed in heating is affected by batch schedule. Since Direct Hot Charge Rolling (DHCR) has significant benefits on energy cost, great efforts have been made to improve the ratio of DHCR while performing batch scheduling [19, 20]. Besides that, optimization of rolling schedule by adjusting thickness reduction ratio of slabs between the rolling passes, another way to reduce power consumption that used to drive rolling motor, has also been proposed [21–23].

As mentioned previously, most methods of hot rolling production scheduling concentrate on internal production organization. Although some technical means have been proposed and applied to achieve energy conservation, their potential would be exhausted due to equipment and technology constraints. In this context, methods utilizing favorable external environments should be explored for energy saving. TOU pricing provides an opportunity to reduce electricity costs, but until now there are few published papers to implement ELD under TOU pricing for hot rolling production.

This paper considers the Hot Rolling Production Scheduling Problem (HRPSP) as a mixture of batch scheduling problem and time-dependent job-shop scheduling problem. The rolling units, modeled as power load units, are planned and scheduled according to TOU prices. Primary objective of the proposed model is to minimize electricity costs while considering the traditional objective to minimize penalties caused by jumps between adjacent slabs. A multiobjective optimization model and corresponding solving algorithm are additionally proposed.

The rest of this paper is classified as follows: in Section 2, characteristics of the problem and opportunities under TOU pricing are presented, and a mathematical model with objective to minimize electricity costs in production is formulated. A multiobjective optimization algorithm is developed in Section 3 to solve the problem. Section 4

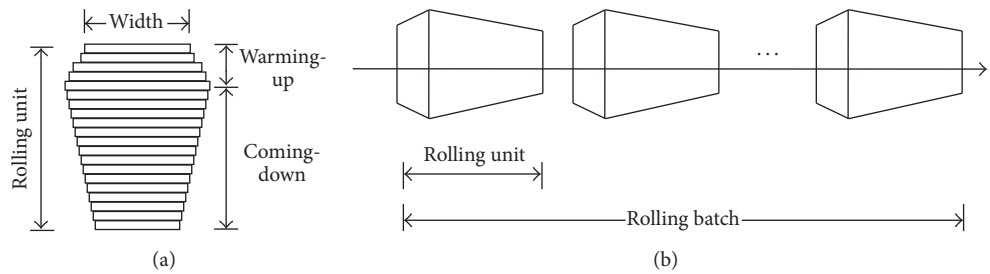


Figure 2: Diagrammatic sketch of batch scheduling: (a) rolling unit and (b) rolling batch.

is dedicated to the experimental procedure and results to evaluate the proposed method; also the peak load regulation effect and robustness of the proposed method are further discussed. Finally, conclusion and future research planning are given in Section 5.

2. Problem Description and Formulation

HRPSP is an extremely complex problem which has significant influence on product quality, production efficiency, and energy consumption. In this paper, we study the Hot Rolling Batch Scheduling Problem (HRBSP) combined with the job-shop scheduling problem (JSP), where HRBSP focuses on how rolling units are organized and the JSP concentrates on when the rolling units are processed.

2.1. Problem Description. Hot rolling batch scheduling is a key process in hot rolling. The task of HRBSP, as is depicted in Figure 2, is to select, group, and sequence slabs into rolling units with the constraints of production capacity and rolling rules. Each rolling unit has a coffin-shaped width profile consisting of a warming-up section and a coming-down section. In the previous section slabs are arranged from narrow to wide to warm up the rolls, and in the later section slabs are scheduled with decreasing width to avoid marking the coils surface. The major part of a rolling unit is the coming-down section, in which the quality of rolling mainly depends on the sequence of slabs. In most cases, the warming-up section is trivial and can be determined manually.

Several constraints restrict the scheduling, the most important one of which is to smooth jumps in width, gauge, and hardness between adjacent slabs. Other constraints, such as cumulative rolling length of slabs in a rolling unit and continuous rolling length of slabs with same width, are also considered to ensure product quality and production capability.

Because hot rolling is a key energy intensive process in steel industry, many approaches, such as optimization of batch scheduling with the objective of improving DHCR ratio and optimization of reduction schedule, have been proposed to achieve energy saving. In smart grid, TOU electricity pricing, which is one of the most commonly implemented demand response programs [24], provides a new opportunity for steel mill to achieve ELD in hot rolling production, which means cutting costs by shifting loads according to the electricity price.

As is shown in Figure 3, a whole day is partitioned into four types of periods based on the price of electricity: on-peak, mid-peak, flat-peak, and off-peak periods. We can see that the power cost for each rolling unit, which not only is determined by the quantity of power demand but also is dependent on the corresponding electricity pricing, should be accumulated piecewise during the processing time.

Compared with flat electricity pricing, the objective of ELD under TOU pricing is to minimize total power cost, including charges for power consumed from shifting loads. In this paper, we assume that rolling units can be scheduled freely; therefore no operating costs from load shifting are included. Consequently, rolling production is encouraged during off-peak periods and discouraged during on-peak periods. In addition, we should know that the scheduling on fixed jobs is not always optimal, so the scheduled jobs (i.e., the rolling units obtained by hot rolling batch scheduling) should be created and associated with their operation time. Finally, the problem is turned into optimal scheduling for minimizing the electricity costs determined by batch scheduling solution and job-shop scheduling solution under specified electricity pricing, while the traditional objective that smoothing changes between adjacent slabs should not be ignored to ensure product quality.

2.2. Mathematical Formulation. We interpret the basic model of the HRBSP as a vehicle routing problem (VRP), which is a classical combinatorial optimization problem. In the model, it can be considered that each rolling unit is a vehicle within limited capacity and each slab is a customer that should be visited at most once. Suppose that there are n slabs to be scheduled into m rolling units; the objective of the problem is to determine m routes (rolling units) to minimize the total distance traveled (penalties caused by jumps between adjacent slabs).

The variables used in formulation are listed as follows:

N: a set of slabs; $N = \{1, 2, \ldots, n\}$

M: a set of rolling units; $M = \{1, 2, \ldots, m\}$

T: a set of time periods; $T = \{1, 2, \ldots, t\}$

π_j: electricity price during time period j

W_i: power demand of slab i during rolling procedure

l_j: rolling length of slab j

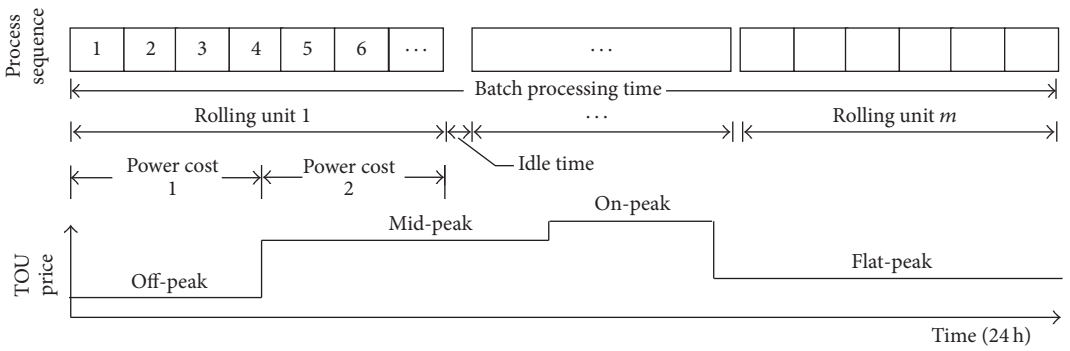


Figure 3: Relationship between production scheduling and electricity costs under TOU pricing.

p_i: processing time for slab i

P_{ij}: the penalty for rolling slab j immediately after slab i, where $P_{ij} = p_{ij}^w + p_{ij}^g + p_{ij}^h$; p_{ij}^w, p_{ij}^g, and p_{ij}^h, respectively, represent the contribution due to width, gauge, and hardness

s_{ij}: binary variable with value of 1 if the widths of slab i and j are the same; otherwise it is 0

ts_i: processing start time of slab i

L: lower bound of the cumulative length of slabs scheduled in a single rolling unit

U: upper bound of the cumulative length of slabs scheduled in a single rolling unit

R: upper bound of the cumulative length of slabs with the same width in a single rolling unit

TS: total time that can be allocated for production

Five decision expressions are defined to identify the scheduling solution as follows:

$$x_{ij}^k = \begin{cases} 1 & \text{if slab } j \text{ is immediately after slab } i \text{ in unit } k, \\ 0 & \text{otherwise,} \end{cases}$$

$$y_i^k = \begin{cases} 1 & \text{if slab } i \text{ is scheduled in rolling unit } k, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

$$r_{ij}^k = \begin{cases} 1 & \text{if slab } j \text{ is rolled after slab } i \text{ in rolling unit } k, \\ 0 & \text{otherwise,} \end{cases}$$

$$d_i^j = \begin{cases} 1 & \text{if slab } i \text{ is processed in time periods } j, \\ 0 & \text{otherwise.} \end{cases}$$

v_i, a positive integer or 0, is a variable to indicate the idle time allocated to rolling unit i before production.

Note that production efficiency may not always be the only one target in engineering, especially when production capacity is abundant; then the target of our model is to minimize electricity costs on the premise of processing all products in given time horizon. According to basic VRP model combined with consideration of relationship between slab processing sequence and processing time as shown in Figure 3, we formulate the hot rolling production optimization problem as

$$\min \quad f_1 = \sum_{k\in M}\sum_{i\in N}\sum_{j\in N} P_{ij}\cdot x_{ij}^k \tag{2}$$

$$\min \quad f_2 = \sum_{j\in T}\left(\pi_j \cdot \sum_{i\in N} W_i \cdot d_i^j\right) \tag{3}$$

$$\text{s.t.} \quad \sum_{i\in N} x_{ij}^k = y_j^k, \quad j \in N,\ k \in M \tag{4}$$

$$\sum_{j\in N} x_{ij}^k = y_i^k, \quad i \in N,\ k \in M \tag{5}$$

$$\sum_{k\in M} y_i^k = 1, \quad i \in N \tag{6}$$

$$\sum_{i\in N} r_{ij}^k \cdot s_{ij} \cdot l_j \le R, \quad j \in N,\ k \in M \tag{7}$$

$$L \le \sum_{i\in N} y_i^k \cdot l_i \le U, \quad k \in M \tag{8}$$

$$0 \le \sum_{i\in M} v_i \le TS - \sum_{i\in N} p_i \tag{9}$$

$$\sum_{k\in M} r_{ij}^k \le 1, \quad i \in N,\ j \in N \tag{10}$$

$$x_{ij}^k \le r_{ij}^k, \quad i \in N,\ j \in N,\ k \in M \tag{11}$$

$$r_{ij}^k \le y_i^k, \quad i \in N,\ j \in N,\ k \in M \tag{12}$$

$$r_{ij}^k \le y_j^k, \quad i \in N,\ j \in N,\ k \in M, \tag{13}$$

where objective f_1 is the traditional objective to ensure product quality, which means to minimize the total penalties caused by jumps between adjacent slabs, and objective f_2

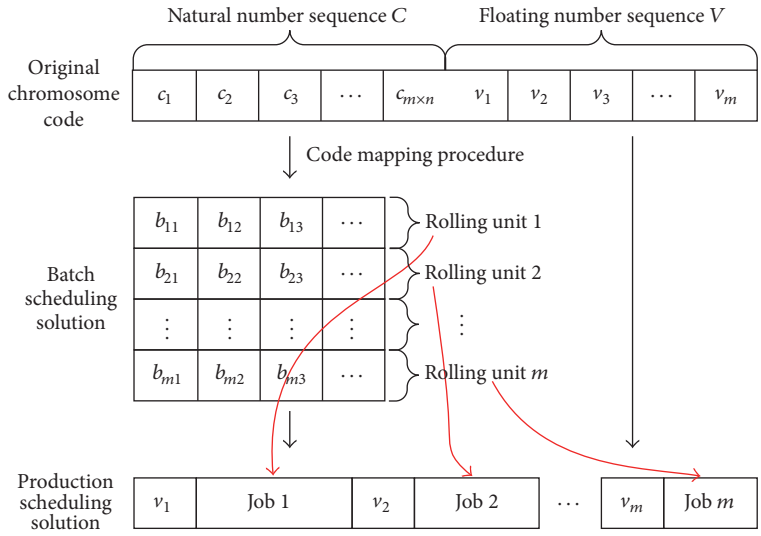


Figure 4: Relationship between production scheduling and electricity costs under TOU pricing.

means to minimize the total electricity costs in hot rolling production, in which d_i^j can be further formulated as

$$d_i^j = \begin{cases} 1 & \text{if } \sum_{\alpha<j} \lambda_\alpha \le ts_i < \sum_{\alpha\le j} \lambda_\alpha, \\ 0 & \text{otherwise,} \end{cases} \tag{14}$$

where the condition correspond to $d_i^j = 1$ means that slab i is processed in time period j. Note that variable ts_i not only is determined by which rolling units the slab is scheduled in but also depended on the processing time of previous slabs and the idle time allocated for rolling units; then it can be expressed as

$$ts_i = \sum_{\delta\in M} y_i^\delta \cdot \left(\sum_{\beta<\delta} \sum_{\alpha\in N} y_\alpha^\beta \cdot p_\alpha + \sum_{\alpha\le\delta} v_\alpha + \sum_{\beta=\delta} \sum_{\alpha\in N} r_{\alpha i}^\beta \cdot p_\alpha \right), \tag{15}$$

where δ is a traversal variable to search the rolling unit that slab i is allocated in; expression in brackets indicates the cumulative time before processing slab i. If slab i is not allocated in rolling unit δ, the expression in brackets would be ignored because $y_i^\delta = 0$.

Constraints (4) and (5) specify the sequence of slabs in a rolling unit. Constraint (6) ensures that each slab can be scheduled only once. Constraint (7) restricts the cumulative length of continuously rolled slabs with the same width in each rolling unit. Constraint (8) indicates rolling mill production capacity, which restricts the lower and upper bounds of cumulative length of slabs in each rolling unit. Constraint (9) means that the total idle time allocated for rolling units cannot be greater than margin of production capability. Constraints (10)–(13) restrict the value of r_{ij}^k, x_{ij}^k, and y_i^k according to their logical relationship.

3. Production Scheduling Optimization Method

As known that VRP is a classical NP-hard problem, it is hard to find the optimal solution for large scale problem. Since there are a large number of slabs in the day-ahead scheduling problem combined with complex objective functions, such as f_2 with quadratic equation (15), it is difficult to find the exact optimal solution, even a feasible solution. In this paper, the production scheduling method consists of two stages. In the first stage, objectives shown in (2)-(3) are optimized simultaneously, and a set of Pareto optimal solutions is generated by the multiobjective optimization algorithm. In the second stage, TOPSIS based multicriteria decision-making is performed to recommend an optimal solution to facilitate field operation.

3.1. NSGA-II Based Multiobjective Optimization. Recently, many swarm intelligence algorithms are introduced to solve complex optimization problem, in which Nondominated Sorting Genetic Algorithm with Elitism (NSGA-II) proposed by Deb [25] is a typical method to solve multiobjective problem. NSGA-II has been widely used to solve combinatorial optimization problems in engineering, such as hydrothermal power scheduling problem [26], job sequencing problem [27], and flow-shop scheduling problem [28]. In this paper, a NSGA-II based Multiobjective Production Scheduling Algorithm (MOPSA) is developed to solve the HRPSP; some personalized changes are made to instantiate the algorithm, in which the most important things are designing customized chromosome code and genetic operators to adapt specific problem.

3.1.1. Chromosome Encoding. In order to contain information of both batch scheduling and job-shop scheduling, a hybrid chromosome code consisting of two sections as shown in Figure 4 is designed. The first section is a natural number

sequence C that can be transformed to a two-dimensional matrix B through a code mapping procedure, where B represents a batch scheduling solution and element b_{ij} in B is the original sequence of slab j in rolling unit i. For each i, if the minimal j is found while $b_{ij} = 0$, it can be resolved that the last slab in rolling unit i is $b_{i,j-1}$. The second section is a floating number sequence V that represents the idle time allocated during job-shop scheduling, where job means production of rolling units.

According to above description, the hybrid chromosome code G can be expressed as

$$\begin{aligned} G &= (C, V), \\ C &= (c_1, c_2, \ldots, c_{m\times n}), \\ V &= (v_1, v_2, \ldots, v_m), \end{aligned} \tag{16}$$

where element c_i in C is a natural number that ranged from 1 to $m \times n$, m is the quantity of rolling units, n is the quantity of slabs to be scheduled, any two numbers c_i and c_j are assigned to different values, and v_i in V represents the idle time allocated to rolling unit i before rolling production.

Detailed steps of the code mapping procedure as mentioned previously are listed as follows.

Step 1. Set f_i $(i = 1, 2, \ldots, n)$ to 0, where f_i is a flag and $f_i = 1$ represents the fact that slab i has been scheduled into rolling units; for rolling unit k $(k = 1, 2, \ldots, m)$, set $\text{num}_k = 0$, where num_k means the slab quantity in rolling unit k; set $d_k = 0$, where d_k is the accumulative rolling length in rolling unit k; set $q_k = 0$, where q_k means the continuously rolled length of slabs with same width in rolling unit k; define a loop variable j and set $j = 1$.

Step 2. Confirm the variables s and k in accordance with natural number c_j, by which slab s scheduled in rolling unit k can be determined. s and k can be calculated by

$$\begin{aligned} s &= c_j - \left[\frac{c_j - 1}{m}\right] \times m, \\ k &= \left[\frac{c_j - 1}{n}\right] + 1. \end{aligned} \tag{17}$$

Step 3. Check if condition $f_s = 0$ is satisfied:

(i) If it is satisfied, it means that slab s is an unscheduled slab. Then if $w_s \neq w'_k$, set $q_k = 0$, where w_s is the width of the slab s and w'_k is the width of the latest appended slab in rolling unit k. Furthermore, if $d_k + l_s \leq U$ and $q_k + l_s \leq R$, put slab s into rolling unit k and update matrix $B(= [b_{ij}])$ by $b_{k,\text{num}_k} = s$, set $\text{num}_k = \text{num}_k + 1$, $d_k = d_k + l_s$, $q_k = q_k + l_s$, and $f_s = 1$.

(ii) Otherwise, go to Step 4.

Step 4. Update $j = j + 1$; go to Step 2 to repeat the above operations until $j = m \times n + 1$.

Step 5. Check if $f_i = 1$ $(i = 1, 2, \ldots, n)$ and $d_k \geq L$ $(k = 1, 2, \ldots, m)$ are all satisfied:

(i) If they are satisfied, it means that all slabs are scheduled into rolling units with subjection to given constraints. Perform idle time allocation procedure to generate sequence V of chromosome code; then $G = (C, V)$ represent a feasible solution of the model in this paper.

(ii) Otherwise, a large number should be assigned to functions f_1 and f_2 to avoid chromosome being selected into new population in next selection operator.

The detailed steps of idle time allocation as mentioned above based on section C of chromosome code are listed as follows.

Step 1. Confirm the electricity price π_i^s corresponding to each rolling unit i when the production starts and price π_i^e when the production completes.

Step 2. Create a random floating number sequence V that represents the idle time allocated for rolling units before production. For elements in V, the constraint as (9) in Section 2.2 must be satisfied.

Step 3. Sort time periods in descending order based on electricity price; after that, a new set of time periods $T' = (t'_1, t'_2, \ldots, t'_t)$ is generated, in which the price associated with t'_k is π'_k; define a loop variable j and set $j = 1$.

Step 4. Adjust the idle time allocation for rolling units. For each rolling unit i that started from time period t'_j, if $\pi_i^e < \pi'_j$ and $v_{i+1} > 0$, set $v_{i+1} = 0$; $v_i = v_i + v_{i+1}$; for rolling unit i that completed in time period t'_j, if $\pi_i^s < \pi'_j$ and $v_i > 0$, set $v_i = 0$; $v_{i+1} = v_{i+1} + v_i$.

Step 5. Update variable $j = j + 1$; go to Step 4 to repeat the above operation for the left time periods until $j = t$, which represent the fact that adjustment of idle time allocation is completed.

The benefits of hybrid encoding and mapping procedure above are not only containing complete information of production scheduling but also handling constraints. From Step 5, we can see that all constraints from (4)–(8) in Section 2.2 are satisfied in accepted feasible solution, which is helpful to reduce the difficulty of problem solving.

3.1.2. Design of Genetic Operators. In order to instantiate the MOPSA algorithm, customized genetic operators are defined to match hybrid chromosome code; the most important operators for genetic algorithm are selection, crossover, and mutation.

Selection operator, which means selecting individuals from population, is done based on the frontier rank of individuals by nondominated sorting. If many individuals have the same rank, the individual with the maximum crowded distance will be selected preferentially.

Partially Mapped Crossover (PMX) mentioned in [29] and Scramble Sublist Mutation (SSM) mentioned in [30] are

adopted to perform operation on section C of chromosome code. The PMX operator is performed on two parent chromosomes: randomly select two crossover points k_1 and k_2 and separate the chromosome code into three sections; swap the gene codes in range $[k_1, k_2]$; after that, replace the other gene codes out of range $[k_1, k_2]$ according to mapping relationship determined by the middle section.

Unlike the crossover operator, SSM mutation operator is performed on single parent chromosome: randomly select two positions p_1 and p_2 that separated less than a fixed length in the chromosome code; then rearrange the gene codes within $[p_1, p_2]$.

After crossover or mutation operation, update section V of the chromosome code to allocate idle time for rolling units immediately.

3.1.3. Decision Expressions and Fitness Function Calculation. We choose the objective functions f_1 and f_2 to be the fitness functions in our genetic algorithm. f_1 represents penalties and f_2 represents electricity costs in production, which are both cost-oriented and need to find minimum value.

In fitness function calculation, most needed variables and expressions are static and can be precomputed except the variables x_{ij}^k in f_1 and d_i^j in f_2, so the key of fitness function calculation is to determine the values of x_{ij}^k and d_i^j based on chromosome code.

According to the characteristics of chromosome code in this paper, we use matrix $B(= [b_{ij}])$ generated in chromosome code mapping procedure instead of part C to perform the following calculation. In order to determine the value of x_{ij}^k, each row in matrix B should be traversed to search the adjacent elements that satisfy the following equation:

$$\begin{aligned} b_{k,j_1} &= i, \\ b_{k,j_1+1} &= j, \end{aligned} \tag{18}$$

where the first equation means slab i is assigned in rolling units k and processed with the sequence j_1 and the next equation indicates that slab j is allocated after slab i immediately in rolling unit k. x_{ij}^k can be determined to be 1 if (18) is satisfied; otherwise, it is 0. For each rolling unit k, penalties between adjacent slabs are accumulated by $P_{ij} \cdot x_{ij}^k$.

Meanwhile, it should be noted that calculation of d_i^j in f_2 would depend not only on B but also on sequence V that represents the allocated idle time for rolling units. According to (14)-(15) defined in Section 2.2, the determination of d_i^j mainly depends on variables y_i^k, r_{ij}^k, and v_i, in which the first two variables can be easily calculated on matrix B by a traversal procedure as done in determining x_{ij}^k, and the last variable v_i can be directly identified by the sequence V in chromosome code. Once d_i^j is known, fitness function f_2 can be accumulated by $\pi_j \cdot W_i \cdot d_i^j$ for each time period.

3.2. TOPSIS Based Multicriteria Decision-Making. As MOPSA generate more than one Pareto optimal solution, in order to facilitate field operation, only a few solutions should be accepted. In this paper, Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) [31], a widely used multicriteria decision-making method to identify solutions from finite alternatives, is adopted as the method to select a recommended optimal solution.

Detailed steps of the TOPSIS based multicriteria decision-making for HRPSP are listed as follows.

Step 1. The decision matrix X can be expressed as

$$X = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \vdots & \vdots \\ x_{m1} & x_{m2} \end{bmatrix}, \tag{19}$$

where X is a two-dimensional matrix with the size of $m \times n$, which means that there are m solutions generated by the multiobjective algorithm and n objectives for the HRPSP, where $n = 2$. The element x_{ij} in X is the value of the jth objective with respect to the ith solution. Then the normalized decision matrix $Z(= [z_{ij}])$ can be calculated according to

$$z_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}}. \tag{20}$$

Step 2. Multiply the normalized decision matrix by its associated weights to calculate the weighted normalized decision matrix $V(= [v_{ij}])$, in which v_{ij} is calculated as

$$v_{ij} = w_j \cdot z_{ij}, \tag{21}$$

where w_j is a weight factor associated with the jth objective. In our context, w_1 and w_2 are set to different values according to preference of two objectives.

Step 3. Identify the ideal solution s^+ and the nadir solution s^- of each objective according to the following equations:

$$\begin{aligned} s^+ &= \left(s_1^+, s_2^+\right), \\ s_j^+ &= \begin{cases} \max\limits_{1 \le i \le m} v_{ij} & \text{if } f_j \text{ is benefit-oriented,} \\ \min\limits_{1 \le i \le m} v_{ij} & \text{if } f_j \text{ is cost-oriented,} \end{cases} \\ s^- &= \left(s_1^-, s_2^-\right), \\ s_j^- &= \begin{cases} \min\limits_{1 \le i \le m} v_{ij} & \text{if } f_j \text{ is benefit-oriented,} \\ \max\limits_{1 \le i \le m} v_{ij} & \text{if } f_j \text{ is cost-oriented.} \end{cases} \end{aligned} \tag{22}$$

It should be known that both of the objectives in HRPSP are cost-oriented, which are said to find the minimum of objective functions.

Table 1: Production data description.

Group Id	Slab quantity	Rolling units quantity	Processing time/(min)	Characteristics
1	450	8	1421.75	Many varieties of slabs and full production load
2	415	8	1323.05	Many varieties of slabs and not full production load
3	450	8	1427.88	Few varieties of slabs and full production load
4	415	8	1318.33	Few varieties of slabs and not full production load

Table 2: TOU electricity tariffs.

Time period	Time frame	Electricity price/(CNY·kWh^{-1})
On-peak	18:00–21:00	0.878
Mid-peak	08:00–11:00, 15:00–18:00	0.778
Flat-peak	07:00-08:00, 11:00–15:00, 21:00-22:00	0.628
Off-peak	00:00–07:00, 22:00–24:00	0.428

Step 4. Measure the distances d_i^+ and d_i^- of the ith solution from the ideal solution s^+ and the nadir solution s^- by

$$d_i^+ = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - s_j^+\right)^2}, \quad i = 1, 2, \ldots, m,$$
$$d_i^- = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - s_j^-\right)^2}, \quad i = 1, 2, \ldots, m. \tag{23}$$

Step 5. Calculate C_i^* that represents the relative closeness of ith solution with respect to the ideal solution according to

$$C_i^* = \frac{d_i^-}{\left(d_i^- + d_i^+\right)}, \quad i = 1, 2, \ldots, m. \tag{24}$$

After completing the above steps, the decision-making can be finally performed on the Pareto optimal solutions according to the sequence determined by C_i^* $(i = 1, 2, \ldots, m)$ in descending order; the solution that owns the maximal relative closeness will be selected as the recommended optimal solution.

4. Experimental Results and Analyses

In this section, we perform a series of experiments to evaluate the effectiveness and performance of the proposed method in different scenario.

4.1. Experimental Procedure. In experimental procedure, four groups of production data as is shown in Table 1 are collected from a steel mill for experimental procedure. For each group of production data, if there are many slab varieties in width, gauge, and hardness, the penalty score between adjacent slabs will be larger. At the same time, full production load means that the idle time for processing slabs will be short.

According to constraints of production equipment and capability, the lower and upper bounds of the cumulative length of slabs scheduled in a single rolling unit are, respectively, set to 5 and 10 kilometers, and the upper bound of the continuously rolled length of slabs with the same width is set to 1 kilometer. For specific slab, rolling length, processing time, and power consumption can be obtained by the hot rolling process control system in steel mill. The penalties caused by jumps between adjacent slabs in width, gauge, and hardness are adopted by referring to [8]. The data in Table 2 are actually performed TOU electricity tariffs in steel mill. According to daily power load distribution, a whole day is split into eight periods that contain four types of time periods; each type of time period is associated with corresponding price.

In order to obtain excellent algorithm performance, the NSGA-II parameters are determined by parameter sensitivity analysis based on empirical value and a lot of tests. The probability of crossover and mutation are set to 0.4 and 0.6, respectively, the population size is set to 50, and the maximum iterations of algorithm are set to 5000. The production scheduling optimization algorithm and TOPSIS decision-making procedure are both implemented and performed in MATLAB.

In experimental procedure, the proposed method (named as PM) is compared with two conventional methods to evaluate effectiveness and performance. Since exact algorithm for large scale HRBSP problem is too difficult to implement, genetic algorithm is often used for solving this problem. In this paper, a relatively new method in [32] with the traditional objective to minimize jump penalties is adopted as a comparison method (named as CM1), in which a hybrid evolutionary algorithm with integration of genetic algorithm and extremal optimization is designed to solve the hot rolling scheduling problem.

Because electricity price during hot rolling changes over time, it is natural to allocate the processing sequence and the idle time of rolling units to avoid on-peak time periods; then the MILP method proposed by [6] is adopted as another comparison method (named as CM2) to find the low bound of electricity costs on the basis of solution obtained in CM1.

Unlike single objective optimization, the result of multiobjective optimization is not a single solution but a set of Pareto optimal solutions; in order to facilitate field operation, we choose different values of objective weight factors w_j in

Table 3: Scheduling results obtained by different methods.

Method	Group 1		Group 2		Group 3		Group 4	
	f_1	f_2	f_1	f_2	f_1	f_2	f_1	f_2
CM1	5035	313254	4528	296357	2957	315894	2659	299623
CM2	5035	309078	4528	276813	2957	312753	2659	278717
PM, $w = [0.9, 0.1]$	5129	308281	4573	275898	3090	311114	2710	277214
PM, $w = [0.4, 0.6]$	7493	305691	6905	274022	3445	309242	3022	275397
PM, $w = [0.1, 0.9]$	7701	305680	7665	273729	3478	309234	3308	274994

Table 4: Detailed parameters of scheduling results for group 1 of production data.

Method	RUS	SQ	RL	PT	PD	APL	PST	PCT	AIT
PM, $w = [0.1, 0.9]$	1	51	8.53	2.44	56.74	23.25	00:00	02:26	0
	2	58	9.95	2.79	64.17	23.00	02:26	05:14	0
	3	55	9.92	2.80	60.75	21.70	05:14	08:02	0
	4	61	9.94	3.44	68.20	19.83	08:02	11:28	0
	5	59	9.96	3.16	66.46	21.03	11:28	14:38	0
	6	57	9.81	3.17	62.24	19.63	14:38	17:48	0
	7	52	9.10	2.89	54.39	18.82	17:48	20:41	0
	8	57	9.94	3.00	66.69	22.23	20:59	23:59	0.3
CM1	1	57	9.80	3.07	63.38	20.64	00:00	03:04	0
	2	62	10.00	3.23	67.76	20.98	03:04	06:18	0
	3	56	9.92	2.96	63.13	21.33	06:18	09:16	0
	4	55	9.47	2.96	62.15	21.00	09:16	12:14	0
	5	58	9.91	2.99	63.72	21.31	12:14	15:13	0
	6	53	9.53	2.83	59.68	21.09	15:13	18:03	0
	7	51	8.54	2.70	56.86	21.06	18:03	20:45	0
	8	58	9.98	2.96	62.95	21.27	20:45	23:42	0

RUS: rolling unit sequence; SQ: slab quantity; RL: rolling length (km); PT: processing time (h); PD: power demand (MW·h); APL: average power load (MW); PST: processing start time (HH:mm); PCT: processing complete time (HH:mm); AIT: allocated idle time (hour).

TOPSIS decision-making procedure to recommend solution with different preference of penalty score and electricity cost. In our experimental procedure, the objective factors $w(= [w_1, w_2])$ of the proposed method are set to $[0.9, 0.1]$, $[0.4, 0.6]$, and $[0.1, 0.9]$, respectively.

Optimization results obtained by different methods are provided in Table 3, in which we can see that penalties obtained by PM with $w = [0.1, 0.9]$ are roughly the same as those obtained by CM1 and CM2 but electricity costs are cut down obviously. It is obvious that load shifting to reduce electricity cost inevitably results in an increasing of penalty score, and we would just like to point out that minimizing jump penalties is a guiding target but not a strictly rigid constraint in engineering. If there is allowance for penalty increase on electricity cost, more significant effect on electricity cost reduction is shown, which tells us that penalty relaxation can play an import role, while electricity costs are the key consideration in production; as a consequence, we can utilize objective weight factors in TOPSIS procedure to adjust preferences of the two objectives. In our cases, electricity cost obtained by PM with TOPSIS decision-making on each group of data is less than CM1; even compared to CM2, which includes load shifting on fixed rolling batches, the result is still better; this advantage is attributed to TOU pricing based batching to construct rolling units. Besides that, we can see that the optimization effect is more significant, while the production load is not full, that is, groups 2 and 4, which is caused by more idle time margin which existed to avoid on-peak time periods in such situation.

4.2. Scheduling Results Analysis. In this section, group 1 of data is chosen to have a detailed analysis on job scheduling results firstly. Because the main idea of this paper is ELD, the proposed method PM with $[w_1, w_2] = [0.1, 0.9]$, which has the most significant effect on electricity cost reduction, is selected to compare with the conventional method CM1. Rolling parameters obtained by both methods are given in Table 4, from which we can see that the parameters are subjected to instantiated constraints, which represent the fact that the schedule is feasible solution. Then, we analyze the scheduling results from two aspects.

On one hand, rolling units in Table 4 are considered as production jobs and are illustrated in Figure 5. As it can be seen, in any subfigure, heavy loads are allocated in off-peak and flat-peak periods by PM, while light loads are allocated in on-peak or mid-peak periods. In addition, idle time is allocated at 18:00 to 21:00 for our scenarios. Another

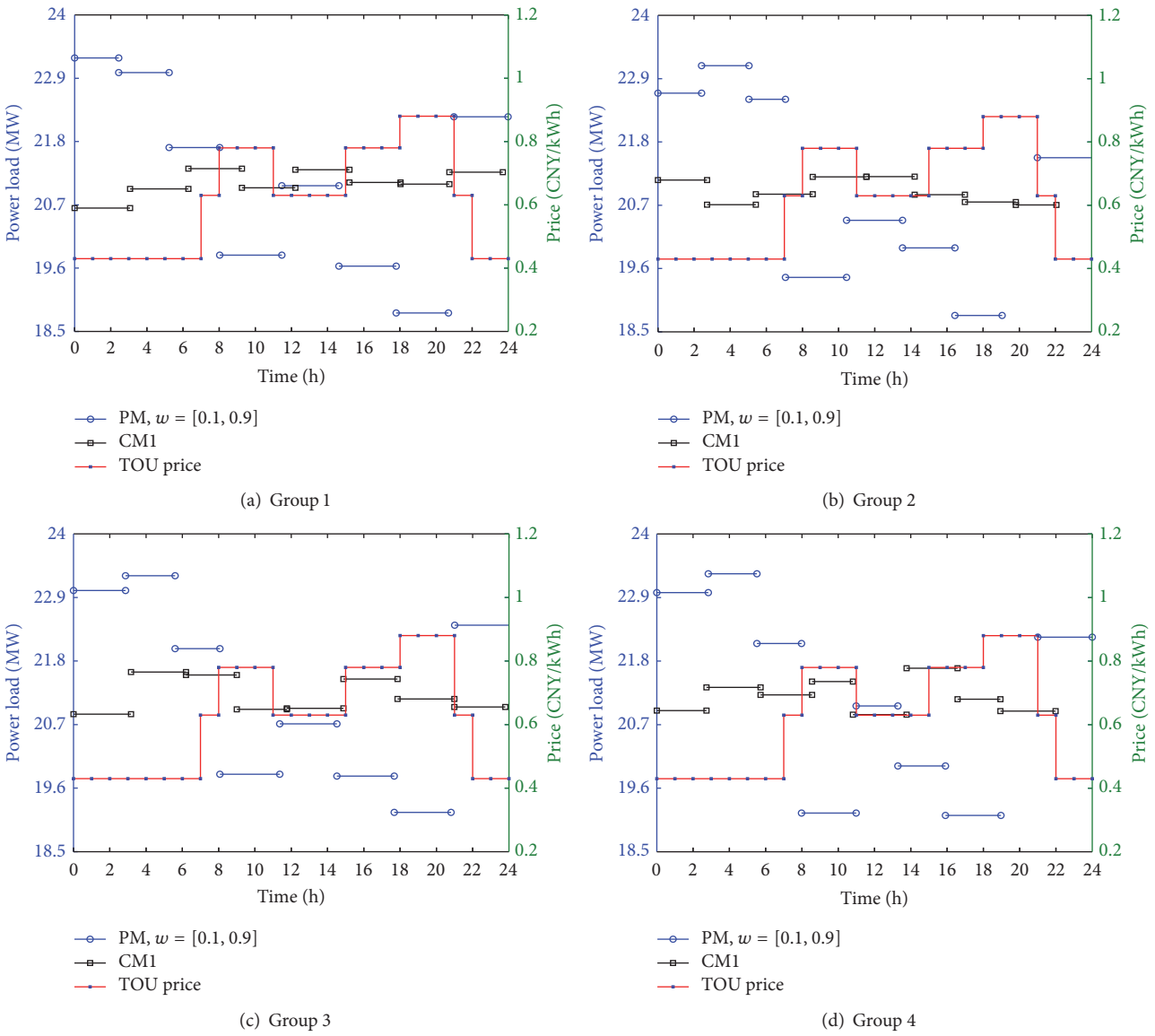


Figure 5: Illustration of job scheduling results obtained by PM and CM1.

phenomenon is that the power load difference between heavy load and light load in PM is greater than that in CM1 and CM2, which is due to the fact that rolling units in PM are organized by TOU electricity price and their processing time.

On the other hand, average power load distribution among time periods is illustrated in Figure 6. Compared to CM1, power load obtained by PM reduces greatly in the last on-peak periods and increases substantially in last off-peak period, especially for group 2 and group 4, which are characterized by not full production load. At the same time, power load in the first off-peak period increases in a certain extent. In addition, power load distribution obtained by PM is also better than that obtained by CM2 based on the principle of load shifting corresponding to TOU pricing, which confirms the effectiveness and advancement of the proposed method furthermore.

From above results and analyses, we know that the advantages of our proposed method on electricity cost reduction can be attributed to two aspects: one is load shifting to avoid on-peak time periods and the other one is TOU pricing based load planning.

4.3. Robustness of the Algorithm. It is well known that NSGA-II is a randomized algorithm; each run of the algorithm may get different results. For evaluating robustness of the algorithm, we use box plot to portray the convergence metric in repeated operation, which is represented by average value of minimum normalized Euclidean distance and indicates the disparity between approximate Pareto frontier and ideal Pareto frontier.

Assume that $P^* = (p_1, p_2, \ldots, p_{|P^*|})$ are the optimal solutions evenly distributed on ideal Pareto frontier and $A^* =$

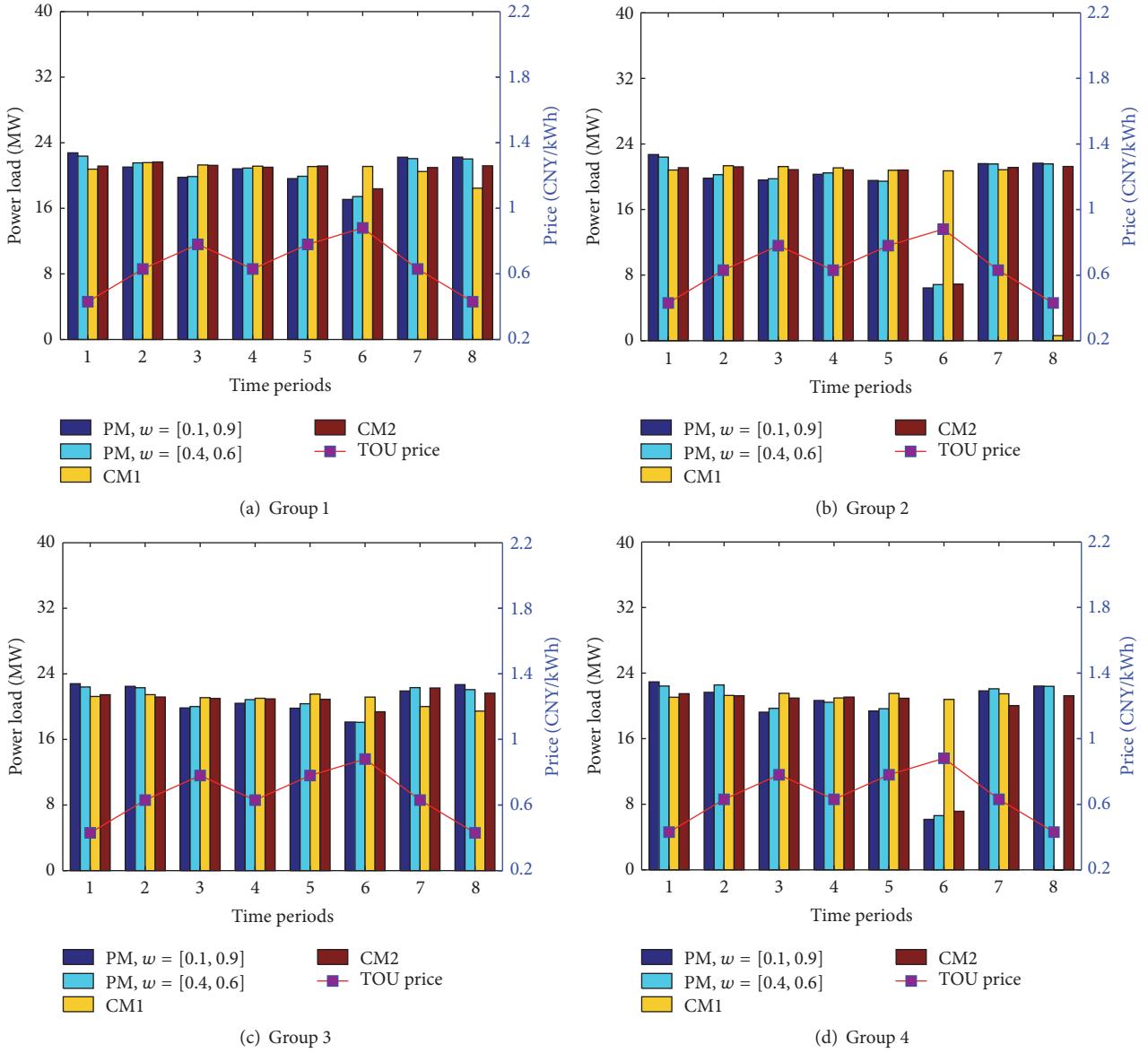


Figure 6: Power load distribution among time periods.

$(a_1, a_2, \ldots, a_{|A^*|})$ are the approximate solutions obtained in a single run of the proposed algorithm. For any a_i, minimum normalized Euclidean distance d_i between a_i and P^* can be calculated by

$$d_i = \min_{j=1}^{|P^*|} \sqrt{\sum_{m=1}^{2} \left(\frac{f_m(a_i) - f_m(a_j)}{f_m^{\max} - f_m^{\min}} \right)^2}, \tag{25}$$

where $f_m^{\max}$ and $f_m^{\min}$, respectively, represent the maximum and minimum values of the mth objective function in P^*, and then the convergence metric C can be expressed as

$$C(A^*) \triangleq \frac{\sum_{i=1}^{|A^*|} d_i}{|A^*|}. \tag{26}$$

Note that the ideal Pareto frontier is always unknown in real problem; the algorithm proposed in this application is run 30 times, respectively, on each group of production data, and then a pseudo-Pareto frontier, which consists of all the solutions in 30 runs with removing dominated solutions, is constructed to compare with the approximate Pareto frontiers. For every run, box plots based on convergence metrics are illustrated in Figure 7. In general, metric C in less than 10^{-2} means good statistical convergence performance in Pareto optimality based multiobjective optimization. The symbol "+" in Figure 7 refers to an outlier in box statistics; nevertheless, it can be seen that the outlier is very close to 10^{-2}. Overall, we can see that the upper edges on different groups of data are all less than 10^{-2}, except a slightly larger value on group 4 and an outlier on group 2. However, the 3rd quartile on group 4 is totally in the range of less than 10^{-2}. The statistical results show that the proposed algorithm is stable in repeated run. On the whole, we can conclude that

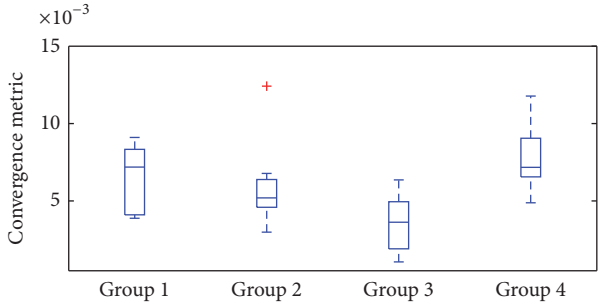


Figure 7: Box plots based on convergence metrics.

the proposed algorithm is robust and suitable for application in engineering.

5. Conclusions

This paper presented the challenge of energy saving in hot rolling production and formulated a multiobjective optimization model of HRPSP under TOU electricity pricing. Objective of the model is to minimize electricity costs in production while considering penalties caused by jumps between adjacent slabs. Since exact algorithm is difficult to implement for solving the large scale problem, a NSGA-II based production scheduling algorithm was developed to obtain Pareto optimal solutions, and then TOPSIS decision-making method was adopted to recommend solution with different objective preferences. Experimental results and analyses showed that the proposed method cuts electricity costs in production, and the performance is better than load shifting on fixed production load. Consider that multiple production lines existed in most steel mills; HRPSP integrated multiple parallel machine job-shop scheduling will be the subject of further study, which is expected to have greater benefits. Besides that, multistage scheduling problem will also be our next work.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61402391 and 61573299) and Natural Science Foundation of Hunan Province of China (2015JJ5027).

References

[1] F. Shrouf, J. Ordieres-Meré, A. García-Sánchez, and M. Ortega-Mier, "Optimizing the production scheduling of a single machine to minimize total energy consumption costs," *Journal of Cleaner Production*, vol. 67, pp. 197–207, 2014.

[2] K. Fang, N. A. Uhan, F. Zhao, and J. W. Sutherland, "Scheduling on a single machine under time-of-use electricity tariffs," *Annals of Operations Research*, vol. 238, no. 1-2, pp. 199–227, 2016.

[3] S. Mitra, J. M. Pinto, I. E. Grossmann, and N. Arora, "Optimal production planning under time-sensitive electricity prices for continuous power-intensive processes," *Computers & Chemical Engineering*, vol. 38, pp. 171–184, 2012.

[4] S. Mitra, J. M. Pinto, and I. E. Grossmann, "Optimal multi-scale capacity planning for power-intensive continuous processes under time-sensitive electricity prices and demand uncertainty. Part I: modeling," *Computers and Chemical Engineering*, vol. 65, pp. 89–101, 2014.

[5] S. Ashok, "Peak-load management in steel plants," *Applied Energy*, vol. 83, no. 5, pp. 413–424, 2006.

[6] Z. Wang, F. Gao, Q. Zhai, X. Guan, K. Liu, and D. Zhou, "An integrated optimization model for generation and batch production load scheduling in energy intensive enterprise," in *Proceedings of the IEEE Power and Energy Society General Meeting (PES '12)*, IEEE, San Diego, Calif, USA, July 2012.

[7] M. K. Bante, M. S. Tarnekar, and M. D. Tutakane, "Energy efficiency in a steel rolling mill by effective planning (case study)," *Energy*, vol. 2, no. 12, pp. 11–16, 2013.

[8] E. D. Kosiba, J. R. Wright, and A. E. Cobbs, "Discrete event sequencing as a traveling salesman problem," *Computers in Industry*, vol. 19, no. 3, pp. 317–327, 1992.

[9] L. Lopez, M. W. Carter, and M. Gendreau, "The hot strip mill production scheduling problem: a tabu search approach," *European Journal of Operational Research*, vol. 106, no. 2-3, pp. 317–335, 1998.

[10] L. Tang and X. Wang, "Iterated local search algorithm based on very large-scale neighborhood for prize-collecting vehicle routing problem," *International Journal of Advanced Manufacturing Technology*, vol. 29, no. 11-12, pp. 1246–1258, 2006.

[11] A. L. Chen, G. K. Yang, and Z. M. Wu, "Production scheduling optimization algorithm for the hot rolling processes," *International Journal of Production Research*, vol. 46, no. 7, pp. 1955–1973, 2008.

[12] B.-I. Kim, "Some comments on Chen et al. 'Production scheduling optimization algorithm for the hot rolling processes'," *International Journal of Production Research*, vol. 48, no. 7, pp. 2165–2167, 2010.

[13] B. Alidaee and H. B. Wang, "On the integer programming formulation of production scheduling optimisation algorithm for the hot rolling processes," *International Journal of Production Research*, vol. 50, no. 20, pp. 6036–6039, 2012.

[14] S. Jia, J. Zhu, G. Yang, J. Yi, and B. Du, "A decomposition-based hierarchical optimization algorithm for hot rolling batch scheduling problem," *International Journal of Advanced Manufacturing Technology*, vol. 61, no. 5—8, pp. 487–501, 2012.

[15] S. J. Jia, J. Yi, G. K. Yang, B. Du, and J. Zhu, "A multi-objective optimisation algorithm for the hot rolling batch scheduling problem," *International Journal of Production Research*, vol. 51, no. 3, pp. 667–681, 2013.

[16] J.-Y. Moon, K. Shin, and J. Park, "Optimization of production scheduling with time-dependent and machine-dependent electricity cost for industrial energy efficiency," *International Journal of Advanced Manufacturing Technology*, vol. 68, no. 1-4, pp. 523–535, 2013.

[17] M. K. Sarakhsi, S. M. T. F. Ghomi, and B. Karimi, "A new hybrid algorithm of scatter search and Nelder-Mead algorithms to optimize joint economic lot sizing problem," *Journal of Computational and Applied Mathematics*, vol. 292, pp. 387–401, 2016.

[18] K. Puttkammer, M. G. Wichmann, and T. S. Spengler, "Hot strip mill scheduling under consideration of energy consumption,"

in *Operations Research Proceedings 2013*, Operations Research Proceedings, pp. 355–361, Springer, 2014.

[19] J. Zhao, W. Wang, Q. Liu, Z. Wang, and P. Shi, "A two-stage scheduling method for hot rolling and its application," *Control Engineering Practice*, vol. 17, no. 6, pp. 629–641, 2009.

[20] D. Shan, A.-J. Xu, Y.-M. Lu, and H.-B. Wang, "Research on modeling and optimization algorithm for hot rolling batch planning of DHCR production," in *Proceedings of the International Asia Conference on Industrial Engineering and Management Innovation: Core Areas of Industrial Engineering (IEMI '12)*, pp. 527–536, Springer, August 2012.

[21] M. Tan, B. Duan, X. Zhou, and Y. Li, "Energy optimization of medium plate rolling schedule based on improved particle swarm optimization algorithm," *Journal of Central South University (Science and Technology)*, vol. 45, no. 5, pp. 1476–1483, 2014.

[22] J.-M. Yang, H.-J. Che, F.-P. Dou, and T. Zhou, "Genetic algorithm-based optimization used in rolling schedule," *Journal of Iron and Steel Research International*, vol. 15, no. 2, pp. 18–22, 2008.

[23] Y. Li, Y. Wang, X. Zhao, and M. Ren, "Multi-objective optimization of rolling schedules for tandem hot rolling based on opposition learning multi-objective genetic algorithm," in *Proceedings of the 25th Chinese Control and Decision Conference (CCDC '13)*, pp. 846–849, IEEE, Guiyang, China, May 2013.

[24] P. Palensky and D. Dietrich, "Demand side management: demand response, intelligent energy systems, and smart loads," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 3, pp. 381–388, 2011.

[25] K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan, *A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II*, Lecture Notes in Computer Science, Springer, Berlin, Germany, 2000.

[26] K. Deb, N. U. B. Rao, and S. Karthik, "Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4403, pp. 803–817, 2007.

[27] S. Bandyopadhyay and A. Das, "Proposing modified NSGA-II to solve a job sequencing problem," in *Proceedings of the International Conference on Advances in Computing*, pp. 387–392, Springer India, 2012.

[28] H. Asefi, F. Jolai, M. Rabiee, and M. E. Tayebi Araghi, "A hybrid NSGA-II and VNS for solving a bi-objective no-wait flexible flowshop scheduling problem," *The International Journal of Advanced Manufacturing Technology*, vol. 75, no. 5-8, pp. 1017–1033, 2014.

[29] A. S. Tasan and M. Gen, "A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries," *Computers and Industrial Engineering*, vol. 62, no. 3, pp. 755–761, 2012.

[30] Ö. Mutlu, O. Polat, and A. A. Supciller, "An iterative genetic algorithm for the assembly line worker assignment and balancing problem of type-II," *Computers and Operations Research*, vol. 40, no. 1, pp. 418–426, 2013.

[31] G. H. Tzeng and J. J. Huang, *Multiple Attribute Decision Making: Methods and Applications*, CRC Press, 2010.

[32] Y.-W. Chen, Y.-Z. Lu, M. Ge, G.-K. Yang, and C.-C. Pan, "Development of hybrid evolutionary algorithms for production scheduling of hot strip mill," *Computers and Operations Research*, vol. 39, no. 2, pp. 339–349, 2012.