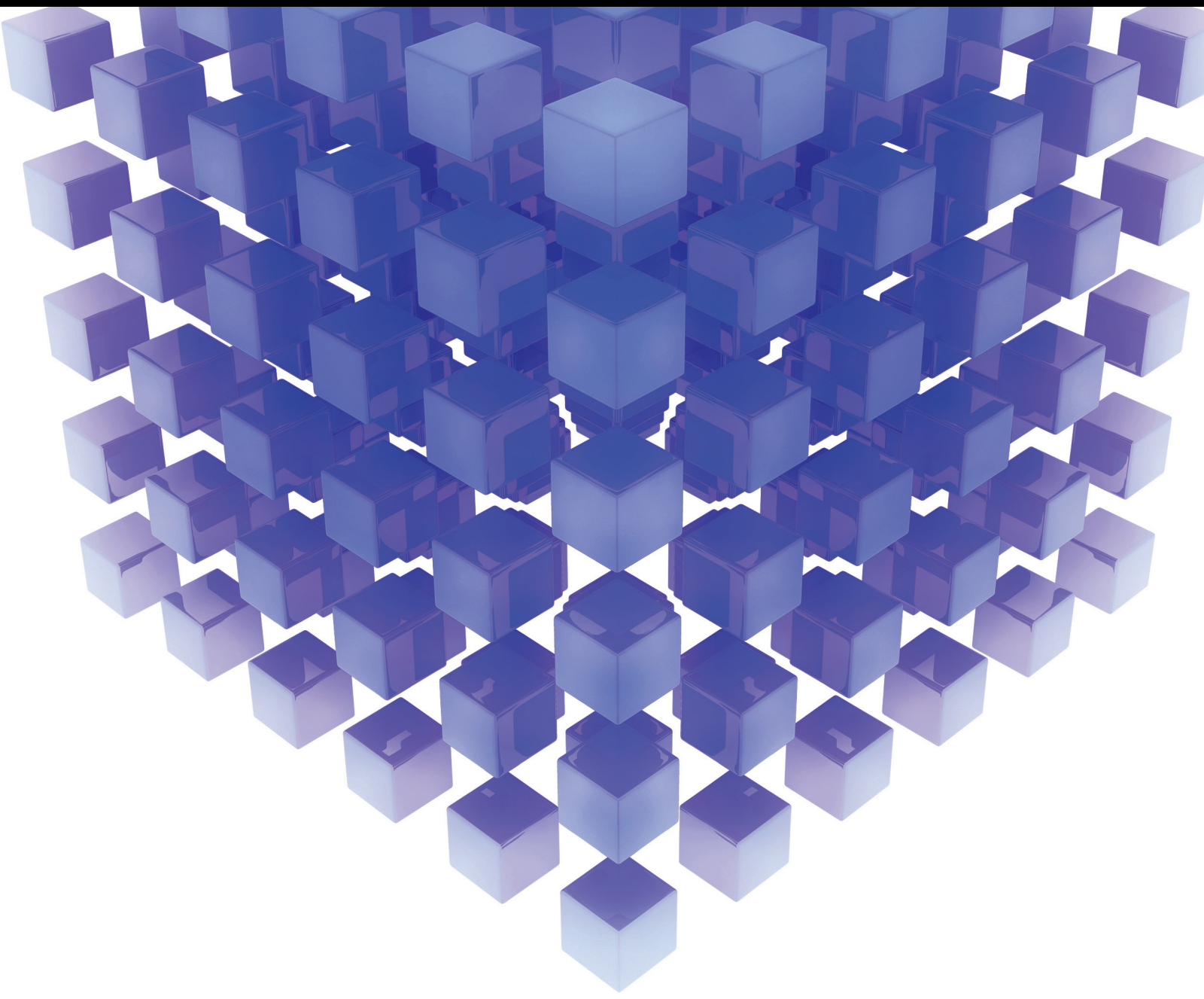
Mathematical Problems in Engineering

Classical and Hybrid Optimization Approaches and Their Applications in Engineering and Economics

Guest Editors: Pandian Vasant, Gerhard-Wilhelm Weber, and Vo Ngoc Dieu



Classical and Hybrid Optimization Approaches and Their Applications in Engineering and Economics

Mathematical Problems in Engineering

Classical and Hybrid Optimization Approaches and Their Applications in Engineering and Economics

Guest Editors: Pandian Vasant, Gerhard-Wilhelm Weber, and Vo Ngoc Dieu

Editorial Board

Contents

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 917093, 2 pages
http://dx.doi.org/10.1155/2015/917093

Editorial

Classical and Hybrid Optimization Approaches and Their Applications in Engineering and Economics

Pandian Vasant,[1] Gerhard-Wilhelm Weber,[2] and Vo Ngoc Dieu[3]

[1]*Department of Fundamental and Applied Sciences, Universiti Teknologi PETRONAS, 32610 Bandar Seri Iskandar, Malaysia*
[2]*Institute of Applied Mathematics, METU, 06531 Ankara, Turkey*
[3]*Department of Power Systems, HCMC University of Technology, Ho Chi Minh City, Vietnam*

Correspondence should be addressed to Pandian Vasant; pvasant@gmail.com

Received 9 April 2015; Accepted 9 April 2015

Classical optimization and, for some years, hybrid optimization have been successfully applied to many problems of engineering and economics. For instance, as exposed and detailed in the literature, gravitational search algorithm (GSA), genetic algorithm (GA), particle swarm optimization (PSO), ant colony optimization (ACO), and several hybrid swarm evolutionary algorithms could be adopted to overcome complex and uncertain real-world optimization challenges. What is more, advances in hybrid optimization techniques are an important section in engineering and economics and they also assist optimization algorithm experts to design and code better procedures. In order to bridge the notions, concepts, and methodologies between those two ends, our special issue focussed on the related topics of integrating and utilizing algorithms in hybrid computational intelligent techniques and their applications in engineering and economics. The hybrid systems can be a hybrid between the classical methods and artificial intelligence-based methods. This special issue may provide the opportunity for practitioners to handle their complicated real-world problems by using hybrid optimization methodologies and for researchers to realize their significant contributions, conveying their knowledge and insights into practice, to look and propose future directions.

This special issue aimed at providing a forum for adopting the state-of-the-art hybrid optimization techniques in engineering and economics, developing the advanced hybrid optimization techniques by using metaheuristics approaches, exchanging of related ideas, and discussing the future directions. A special attention was assigned to exchange, comparison, and combination of the classical and more mathematical and model-based methods of optimization with the many emerging model-free methods from computer science. By this, we aimed to strengthen the mathematical and engineering sciences and to contribute to markets, industries, and economies and, eventually, to the living conditions in the world. We welcomed researchers to present their newest original works, and so many experts, both scholars and practitioners, from all over the world followed our invitation.

Topics of this special issue included, but also went much beyond, the following:

(i) classical optimization methods of mathematics and their applications in engineering and economics;

(ii) continuous optimization applications in engineering and economics;

(iii) combinatorial optimization applications in engineering and economics;

(iv) mixed-integer programming applications in engineering and economics;

(v) hybrid optimization with metaheuristics techniques multiobjective hybrid optimization approaches handling uncertainties with hybrid optimization;

(vi) Lagrange optimization;

(vii) Kuhn-Tucker optimization;

(viii) chaotic hybrid optimization;

(ix) linear and nonlinear optimization;

(x) mathematical programming;

(xi) theoretical aspects of hybrid optimization methods;

(xii) emerging real-world and theoretical applications in engineering and economics.

Acknowledgments

The guest editors are very grateful to the editorial team and entire staff, for their confidence, interest, continuous guidance, and support at all levels of preparation of this special issue's preparation. Furthermore, we would like to sincerely thank Universiti Teknologi PETRONAS, METU, and HCMC University of Technology for their marvellous support and encouragement. We, guest editors, wish all the readers a pleasant and enjoyable, insightful and inspiring lecture of the contributions of this special issue. In fact, we cordially hope that this special issue will present and value this journal as a premium journal in science, engineering, economics, and finance, which strongly fosters deeply needed intellectual advances and their contributions to humankind in all over the world.

Pandian Vasant
Gerhard-Wilhelm Weber
Vo Ngoc Dieu

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 379250, 10 pages
http://dx.doi.org/10.1155/2015/379250

Research Article

Optimal Economic Operation of Islanded Microgrid by Using a Modified PSO Algorithm

Yiwei Ma,[1,2] Ping Yang,[1,2] Zhuoli Zhao,[1,2] and Yuewu Wang[1,2]

[1]*School of Electric Power, South China University of Technology, No. 381, Wushan Road, Tianhe District, Guangzhou 510641, China*
[2]*Guangdong Key Laboratory of Clean Energy Technology, No. 381, Wushan Road, Tianhe District, Guangzhou 510641, China*

Correspondence should be addressed to Yiwei Ma; ma.yiwei@mail.scut.edu.cn

Received 29 October 2014; Revised 18 February 2015; Accepted 18 February 2015

Academic Editor: Gerhard-Wilhelm Weber

An optimal economic operation method is presented to attain a joint-optimization of cost reduction and operation strategy for islanded microgrid, which includes renewable energy source, the diesel generator, and battery storage system. The optimization objective is to minimize the overall generating cost involving depreciation cost, operation cost, emission cost, and economic subsidy available for renewable energy source, while satisfying various equality and inequality constraints. A novel dynamic optimization process is proposed based on two different operation control modes where diesel generator or battery storage acts as the master unit to maintain the system frequency and voltage stability, and a modified particle swarm optimization algorithm is applied to get faster solution to the practical economic operation problem of islanded microgrid. With the example system of an actual islanded microgrid in Dongao Island, China, the proposed models, dynamic optimization strategy, and solution algorithm are verified and the influences of different operation strategies and optimization algorithms on the economic operation are discussed. The results achieved demonstrate the effectiveness and feasibility of the proposed method.

1. Introduction

In recent years, microgrid has received considerable attentions to improve the reliability and economy of power system [1]. Researches and practices show that islanded microgrid integrated with renewable energy sources (RES), the synchronous generators, and energy storage system has been an effective approach to solving the power supply problem in small rural or remote regions, where these communities do not access the utility grid due to the technical and economical reasons [2–4]. However, incorporation of renewable energy sources and other distributed generations (DG) poses a big challenge to the economic operation problem of islanded microgrid [5].

Economic operation is a very important problem to be solved in the planning and operation of power system, and optimal modeling, optimization, and simulation have received considerable attentions in the literature [6–8]. Reference [9] presents a novel energy management system (EMS) to coordinate the power forecasting, output power of different generators, and energy storage to minimize the total operation cost. Reference [10] proposes a multiobjective linear programming methodology to determine the operating levels of various generation units by minimizing two objectives of annual energy cost and annual CO_2 emissions. Similarly, a differential evolution algorithm is used to minimize the total cost comprising the emission cost and the operation and maintenance costs of microgrid in [11]. Clearly, the operation strategy has a significant influence on the actual performance of islanded microgrid system. Reference [12] proposes an idealized predictive dispatch strategy as a benchmark in evaluating simple, nonpredictive strategies, based on assumed perfect knowledge of future load and wind conditions. Reference [13] proposes an optimal dispatch strategy, considering the battery wear cost and the diesel fuel consumption cost, to determine the optimum values of set points for the starting and stopping of the diesel generator as to minimize the total operation costs of islanded microgrid. The optimum control algorithm based on combined dispatch strategies is developed to achieve the optimal unit cost of islanded microgrid [14].

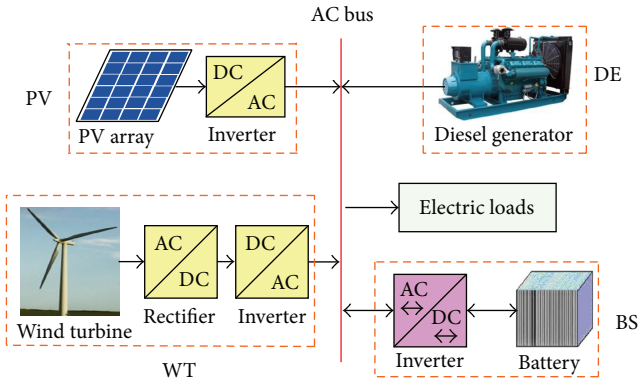


Figure 1: Structure of islanded microgrid.

From the literature survey exposed above, the economic operations of islanded microgrid in those studies are performed under the preset control strategies for minimizing operation cost. In addition, it can be clearly inferred that the cost of islanded microgrid depends on two factors: the unit generating cost of various distributed generation units and the operation strategy. In fact, there are two distinct levels in the operation control of islanded microgrid system: (1) dynamic control, which deals with control of the frequency and magnitude of the output voltage of the system; (2) dispatch control, which deals with the energy flow management from the various sources to load for cost minimization. Thus, the master-slave control strategy is popularly adopted to deal with the control of system frequency and voltage, as well as the energy flow management of islanded microgrid.

This paper presents a joint-optimization of operation strategy and cost reduction to allocate the load demand among various generation units in an economic, stable, and reliable way for islanded microgrid, which is integrated with wind turbine generator (WT) and solar PV generator (PV), the diesel generator (DE), and battery storage system (BS). Considering two different control modes where DE or BS acts as the master unit to guarantee stable operation of islanded microgrid, a novel dynamic optimization process integrated with a modified PSO (MPSO) algorithm is developed to find both optimal operating strategy and optimal cost optimization scheme for the practical economic operation problem of islanded microgrid.

2. Microgrid Operation Strategy

Figure 1 shows a typical system structure of hybrid energy sources islanded microgrid, where PV, WT, DE, and BS units are all connected together to supply power to the electric load through the AC bus. As we know, WT and PV both have a remarkable drawback of their unpredictable nature and dependence on weather and climatic conditions, and their intermittent output powers will cause the system power fluctuation [15]. BS has great advantages of peak shaving and energy buffer to improve RES fluctuation, but the existing big bottleneck of higher cost and shorter lifetime causes the small size of BS in islanded microgrid [16]. DE is still necessary and important to improve the system reliability in case of bad weather conditions that will cause little or zero RES power; however its higher fuel cost and pollution emission are still main concerns of economic operation in islanded microgrid.

The main concerns of islanded operation are to guarantee the system operation stability and reliability, as well as consider economic cost requirement of microgrid. Combining the system operation requirement and control characteristics of various DGs, the master-slave control method is a popular and preferable choice for islanded microgrid to achieve an optimal objective of stable operation and minimum generating cost [17]. In the master-slave control strategy, the controllable microsource is selected as the master unit to track the power fluctuation as to guarantee the stability of system voltage and frequency, and the other microsources do not follow the system power fluctuation [18]. As shown in Figure 1, DE and BS units both are qualified to act as the master unit, but the larger difference of their control response characteristics causes them not to operate as the master unit at the same time [19]. Thus, there are two different operation control modes as below.

(1) Mode 1: BS is the master unit that employs V/f control method to follow the system power fluctuation and guarantee the frequency and voltage stability of islanded microgrid. DE is off or operates at full power to supply power to microgrid and BS as required.

(2) Mode 2: DE is the master unit that uses droop control method to maintain the operation stability of islanded microgrid. However, as the slave unit, BS just use PQ method to absorb RES power or inject power to assist DE to keep system power balance according to the instruction of microgrid energy management system.

3. Economic Operation Model

The economic operation problem of islanded microgrid is to allocate the load demands among the various distributed generation units in such a secure and reliable manner as to minimize the overall system generating cost subject to a set of equality and inequality constraints, while the controllable power sources such as DE or BS act as the master unit to follow system power fluctuation.

3.1. Objective Functions. The objective of microgrid economic operation is to determine generation levels of WT, PV, DE, and BS units to meet the load demand as to minimize the overall system generating cost, including the depreciation cost, the operation cost, the pollutant emission cost, and economic subsidies available for RES over the entire dispatch period $[0, NT]$, where T is the sampling time interval and N is the number of the time intervals. The operation cost comprises the fuel cost and maintenance and operating (M&O) cost:

$$\min C(P) = \sum_{t=1}^{N} \sum_{j=1}^{G} C_j\left(P_j^t\right), \tag{1}$$

where $C(P)$ is the total generation cost of microgrid to produce P during an entire dispatch period NT; G is the numbers of various distributed generation units j; $C_j(P_j^t)$ is the generation cost for unit j to produce P_j^t; P_j^t is the output power of generation unit j during the tth time interval $[(t-1)T, tT]$.

3.1.1. RES Generating Cost. The renewable energy sources of WT and PV units incur only investment costs and exhibit very little operating cost which can be expressed as a fraction of the corresponding initial capital costs [5]. The economic subsidies available for WT and PV units are expressed as their different price subsidies per unit power according to the local green energy generating policy. Thus, the generating cost of RES is formulated in the following:

$$\begin{aligned} C_{\mathrm{RES},i}\left(P_i^t\right) &= C_{\mathrm{DC},i}\left(P_i^t\right) + C_{\mathrm{MO},i}\left(P_i^t\right) - S_{\mathrm{ES},i}\left(P_i^t\right) \\ &= \frac{C_{\mathrm{AIC},i}\left(1+\rho_i\right)}{E_{\mathrm{APG},i}} \cdot P_i^t - k_{\mathrm{ES},i} \cdot P_i^t \\ &= \left(\frac{C_{\mathrm{AIC},i}\left(1+\rho_i\right)}{E_{\mathrm{APG},i}} - k_{\mathrm{ES},i}\right) \cdot P_i^t, \end{aligned} \tag{2}$$

where $C_{\mathrm{DC},i}$, $C_{\mathrm{MO},i}$, and $S_{\mathrm{ES},i}$ are the depreciation cost, the M&O cost, and the economic subsidy for RES unit i to produce P_i^t, respectively; C_{AIC} is the annualized investment cost (¥); ρ is M&O cost coefficients; E_{APG} is the estimated total annual power generation (kWh) based on a typical historical data; k_{ES} is the proportionality coefficient of price subsidy (¥/kWh).

3.1.2. DE Generating Cost. The generating cost of DE is made up of the fixed cost and variable cost. The fixed cost is expressed as the depreciation cost; however the variable cost consists of the fuel cost, M&O cost, and emission cost. The M&O cost is assumed to be proportional with the produced energy. The fuel cost is expressed as an inverse proportion function about the relationship between the nominal power rating and actual output power [20]. The pollution emission costs (NO_x, SO_2, and CO_2) are gained by their proportional coefficients of the fuel cost [21]. The unit depreciation cost can be equal to the annualized investment cost divided by the estimated total annual power generation produced by DE. Thus, the overall generating cost of DE is formulated in the following:

$$\begin{aligned} &C_{\mathrm{DE}}\left(P_{\mathrm{DE}}^t\right) \\ &\quad = C_{\mathrm{DC}}\left(P_{\mathrm{DE}}^t\right) + C_{\mathrm{MO}}\left(P_{\mathrm{DE}}^t\right) + C_{\mathrm{FC}}\left(P_{\mathrm{DE}}^t\right) + C_{\mathrm{EC}}\left(P_{\mathrm{DE}}^t\right) \\ &\quad = \left(\frac{C_{\mathrm{AIC,DE}}}{E_{\mathrm{APG,DE}}} + K_{\mathrm{MO,DE}}\right) \cdot P_{\mathrm{DE}}^t \\ &\qquad + \left(0.146 + 0.05415 \cdot \frac{P_{\mathrm{DE}}^{\mathrm{RD}}}{P_{\mathrm{DE}}^t}\right) \cdot \left(c_{fp} + \sum_{k=1}^{3} c_{E,k}\right), \end{aligned} \tag{3}$$

where $C_{\mathrm{DC}}(P_{\mathrm{DE}}^t)$, $C_{\mathrm{MO}}(P_{\mathrm{DE}}^t)$, $C_{\mathrm{FC}}(P_{\mathrm{DE}}^t)$, and $C_{\mathrm{EC}}(P_{\mathrm{DE}}^t)$ are the depreciation cost, the M&O cost, the fuel consumption cost, and the emission cost, respectively, for DE to produce P_{DE}^t; $C_{\mathrm{AIC,DE}}$ is the annualized investment cost (¥); $E_{\mathrm{APG,DE}}$ is the estimated total annual power generation produced by DE (kWh); $K_{\mathrm{MO,DE}}$ is the M&O cost coefficient (¥/kWh); $P_{\mathrm{DE}}^{\mathrm{RD}}$ is the nominal power rating of DE (kW); c_{fp} is the fuel price for DE (¥/liter); $c_{E,k}$ is the environmental cost coefficient of pollution emission k of NO_x, SO_2, and CO_2 (¥/kg).

3.1.3. BS Wear Cost. The wear cost of BS is expressed as a proportion of the life cycle cost divided by the total discharging power in the whole life time. It is a simple and effective way to calculate in microgrid economic operation, compared with the method in accordance with the battery cycle times. Thus, the wear cost of BS is expressed as follows:

$$C_{\mathrm{BS}}\left(P_{\mathrm{BS}}^t\right) = \beta_{\mathrm{BS}} \cdot P_{\mathrm{BS,dc}}^t, \tag{4}$$

where $C_{\mathrm{BS}}(P_{\mathrm{BS}}^t)$ is the wear cost of BS to discharge P_{BS}^t at t time; β_{BS} is the unit wear cost coefficients of BS; $P_{\mathrm{BS,dc}}^t$ is the discharging power of BS.

3.2. Constraint Conditions

3.2.1. System Power Balance. The balance between power supply and demand is necessary for islanded microgrid, so the equality constraint is expressed as follows:

$$\sum_{j=1}^{G} P_j^t - P_{\mathrm{excessive}}^t = P_{\mathrm{Load}}^t, \tag{5}$$

where $P_{\mathrm{excessive}}^t$ is the system excessive power beyond the load demand; P_{Load}^t is the total system load demand.

3.2.2. Spinning Reserve Capacity. The necessary spinning reserve capacity of the controllable generation units available is necessary to follow the system net load power fluctuation as to guarantee the secure and stable operation of islanded microgrid; thus it is formulated as follows:

$$\begin{aligned} \sum_{g=1}^{R} P_{\mathrm{CG,SR}}^t &\geq \Delta P_{\mathrm{MG,SR}}^t, \\ \Delta P_{\mathrm{MG,SR}}^t &= e_{\mathrm{MG}} \cdot P_{\mathrm{net\text{-}load}}^t, \end{aligned} \tag{6}$$

where R is the numbers of the controllable generation units; $P_{\mathrm{CG,SR}}^t$ is the spinning reserve of the controllable generation unit (CG) available; $\Delta P_{\mathrm{MG,SR}}^t$ is the required spinning reserve of microgrid (MG); e_{MG} is the deviation ratio between the predictive value and the actual value of system net load; $P_{\mathrm{net\text{-}load}}^t$ is the system net load, which equals the system load minus RES power.

3.2.3. Power Generation Limits of Microsources. As for various generating units such as WT, PV, DE, and BS, their output

power ranges are subject to the functional roles in islanded microgrid.

(1) Master Unit. The master unit is responsible for tracking the power fluctuation as to guarantee the stability of system voltage and frequency in islanded operation of microgrid. Thus, its output powers are subject to the following constraints:

$$P^t_{Mh,\max} \le P^t_{Mh} \le P^t_{Mh,\min}, \tag{7}$$

$$\begin{aligned} P^t_{Mh,\max} &= P^{\max}_{Mh} - \Delta P^t_{\mathrm{MG,SR}}, \\ P^t_{Mh,\min} &= P^{\min}_{Mh} + \Delta P^t_{\mathrm{MG,SR}}, \end{aligned} \tag{8}$$

where P^t_{Mh} is the output power of h master unit (Mh) during tth time interval; $P^t_{Mh,\max}$ and $P^t_{Mh,\min}$ are the output power upper and lower limits, respectively; $P^{\max}_{Mh}$ and $P^{\min}_{Mh}$ are the technical maximum and minimum power limits, respectively.

(2) Slave Unit. The slave unit can operate within its technical maximum and minimum power limits, as it does not need to follow system power fluctuation. Thus, its output power is subject to the following inequality constraint:

$$P^{\min}_{S,l} \le P^t_{S,l} \le P^{\max}_{S,l}, \tag{9}$$

where $P^t_{S,l}$ is the output power of l slave unit during tth time interval; $P^{\min}_{S,l}$ and $P^{\max}_{S,l}$ are the technical operation power upper and lower limits of l slave unit, respectively.

3.2.4. Start-Stop Time Constraints. The minimum start-stop time constraint such as DE or WT cannot be less than a certain number:

$$T_{\mathrm{rs},j} \ge T^{\min}_{\mathrm{rs},j}, \tag{10}$$

where $T_{\mathrm{rs},j}$ is the running/stop (rs) time of j generation unit; $T^{\min}_{\mathrm{rs},j}$ is the minimum continuous running or stop time of j unit.

3.2.5. BS Storage Capacity Limits. In order to avoid any deep discharge to decrease battery lifetime [22], this paper selects four key nodes of $\mathrm{SOC}_{\min}$, $\mathrm{SOC}_{\mathrm{low}}$, $\mathrm{SOC}_{\mathrm{high}}$, and $\mathrm{SOC}_{\max}$ to divide the battery energy storage capacity into 4 interval ranges. The maximum SOC ($\mathrm{SOC}_{\max}$) is fully charged (100% of the SOC), and the minimum SOC ($\mathrm{SOC}_{\min}$) is recommended by the manufacturer below which they should not operate:

$$\mathrm{SOC}_{\min} \le \mathrm{SOC}_{\mathrm{low}} \le \mathrm{SOC}(t) \le \mathrm{SOC}_{\mathrm{high}} \le \mathrm{SOC}_{\max}, \tag{11}$$

where $\mathrm{SOC}_{\min}$ and $\mathrm{SOC}_{\max}$ are the minimum and maximum battery storage capacity limits; $\mathrm{SOC}_{\mathrm{low}}$ and $\mathrm{SOC}_{\mathrm{high}}$ are the low and high storage capacity limits of BS normal operation range.

4. MPSO-Based Dynamic Optimization

4.1. Dynamic Economic Optimization Concept. The economic operation of islanded microgrid is commonly regarded as a discrete cost optimization problem to sum up all minimum generation costs of sampling periods in the time domain [23]. The operation strategy and cost reduction are two main concerns to the economic operation problem of islanded microgrid. The operation strategy is mainly to select the master unit to maintain the system stable operation, as well as pay critical influence on the cost reduction [14, 24]. As shown in Figure 1, there are two different control modes of DE or BS as the master unit. In fact, during the whole dispatch period of islanded microgrid, DE and BS might alternate to operate as the sole master unit to maintain the system frequency and voltage stability. Thus, a novel dynamic economic operation optimization process is proposed based on two potential operation control modes as shown in Figure 2, which adopts a MPSO algorithm to find both optimum operation strategy and optimum cost minimization for the economic operation problem of islanded microgrid. This proposed operation strategy is called the dynamic operation strategy.

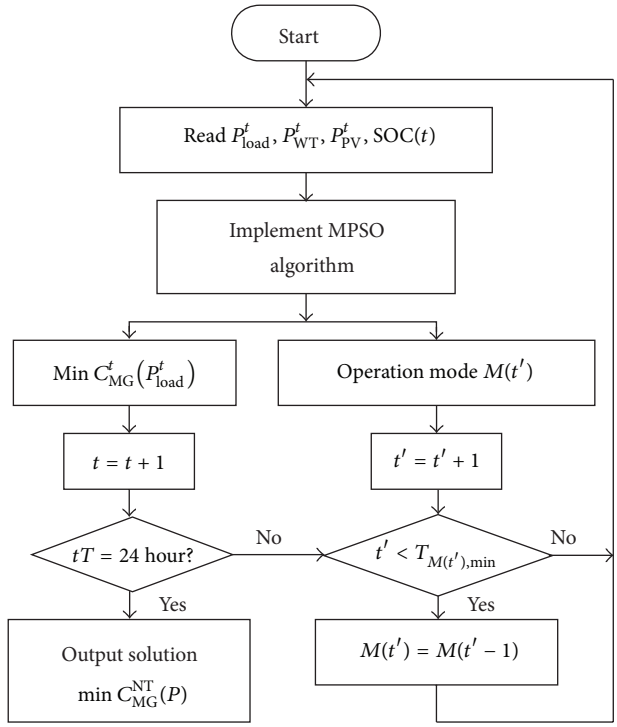


Figure 2: Dynamic economic optimization process.

In the proposed dynamic optimization process as above, there are two different time scales: t is for the cost optimization and t' is for the operation control mode or the type of master unit. It also indicates that the type of master unit will change during dispatch period, so the master unit $M(t')$ needs its own time-domain function ($t' = t' + 1$) to evaluate if its minimum start or stop time constraints are met or not. If the master unit is always fixed as DE or BS over the whole dispatch period or during the initial sampling periods, these two time scales are the same as $t = t'$. Moreover, MPSO algorithm is applied to get the optimal trade-off between the minimum generating cost ($C^t_{\mathrm{MG}}(P^t_{\mathrm{load}})$) for meeting load power (P^t_{load}) and the corresponding operation control mode

that determines the type of master unit ($M(t')$) during tth sampling period.

4.2. MPSO Algorithm. Particle swarm optimization (PSO) is a stochastic population-based algorithm, which was originally introduced by Kennedy and Eberhart [25, 26]. Compared to other optimization techniques such as genetic algorithm (GA), PSO has the most outstanding advantages of few parameters to implement easily and faster convergence speed. In order to find a balance between accelerating the convergence speed and retaining the diversity of the particles, a MPSO with dynamic adaptive inertia weight is adopted to resolve the economic dispatch problem, for it has the improved convergence speed and solution accuracy [27]. In MPSO, a swarm contains a set of population members, which is called the particle. Every particle has both a position and a velocity, and the position represents a candidate solution in the multidimensional solution space, and the velocity moves it from one position to another over the solution space. For the economic operation optimization of islanded microgrid, each particle represents a candidate optimum operation solution.

In a D-dimensional solution space, the position vector of the mth solution is written as $X_m = (x_{m,1}, x_{m,2}, \ldots, x_{m,D})$, and the corresponding velocity vector is given by $V_m = (v_{m,1}, v_{m,2}, \ldots, v_{m,D})$. The position and velocity updating methods for the dth dimension of mth solution at the $(n+1)$th iteration step are expressed as the following equations:

$$x_{m,d}^{n+1} = x_{m,d}^{n} + v_{m,d}^{n+1}, \tag{12}$$

$$v_{m,d}^{n+1} = w \cdot v_{m,d}^{n} + c_1 r_1 \left(pb_{m,d} - x_{m,d}^{n}\right) + c_2 r_2 \left(gb_d - x_{m,d}^{n}\right), \tag{13}$$

$$w = w_0 - h_m^n \cdot w_h + s_m^n \cdot w_s, \tag{14}$$

$$h_m^t = \left| \frac{\min\left(F\left(pb_m^{n-1}\right), F\left(pb_m^n\right)\right)}{\max\left(F\left(pb_m^{n-1}\right), F\left(pb_m^n\right)\right)} \right|, \tag{15}$$

$$s = \left| \frac{\min\left(F_{n\text{best}}^n, \overline{F}_n\right)}{\max\left(F_{n\text{best}}^n, \overline{F}_n\right)} \right|, \tag{16}$$

where x_m^n and v_m^n are vectors representing the position and velocity of the mth particle, respectively; $d \in 1, 2, \ldots, D$ represents the dimension of the particle; w is a dynamically adaptive inertia weight determining how much of particle's previous velocity is preserved; c_1 and c_2 are two positive acceleration constants; r_1 and r_2 are two uniform random sequences sampled from $U(0, 1)$; pb_m is the personal best position found by the mth particle; gb is the best position found by the entire swarm; w_0 is the initial value of w; h_m^n is the speed factor; s is the aggregation degree factor; w_h and w_s are two constants typically within the range $[0, 1]$; $F(pb_m^n)$ is the fitness value of pb_m^n; $\overline{F}_n$ is the mean fitness value of all particles in the swarm at the nth iteration; $F_{n\text{best}}^n$ is the best fitness value achieved by the particles at the nth iteration.

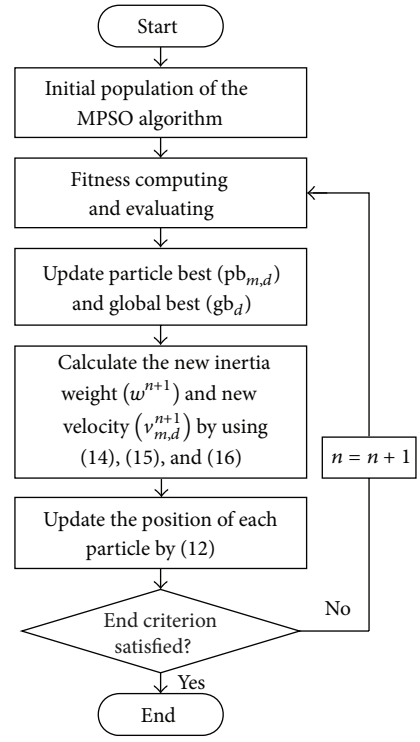


Figure 3: Flowchart of MPSO optimization algorithm.

4.3. The Process of the MPSO Algorithmic Solution. As shown in Figure 2, the MPSO algorithm was adopted to find both optimal operation strategy and optimal minimum generation cost of islanded microgrid. Thus, the algorithm process is shown in Figure 3.

The specific steps are described as follows.

Step 1. The first step includes the following: population initialization of MPSO parameters setting, the total number of the particles, the maximum number of the generations $k_{\max}$, the maximum velocity $v_{\max}$, two positive acceleration coefficients c_1 and c_2, initial inertia weight w_0, and two inertia weight coefficients w_h and w_s.

Step 2. Calculate and evaluate the fitness (1): evaluate the fitness of each particle in the population.

Step 3. Compare the particle's fitness value with $pb_{m,d}$ and gb_d. Update the $pb_{m,d}$ and gb_d according to the comparison result.

Step 4. Calculate the new velocity $v_{m,d}^{n+1}$ by using (15).

Step 5. Update the position of each particle by using (14).

Step 6. Calculate the variance of all particles' fitness functions (1) and check if the converge criterion is met. If it is met, go to Step 7; otherwise, go to Step 2.

Step 7. End and output the optimization result.

Table 1: System configuration of Dongao microgrid.

Generation unit	Power rating	QTY (set)
WT	750 kW	3
PV	250 kW	3
DE	1020 kW/1275 kVA	4
BS	500 kW × 6 h	1

Table 2: Control parameters of DGs.

Type	Parameter (p.u.)	Type	Parameter (p.u.)
P_{MU}^{max}	1 p.u.	SOC_{min}	0.2 p.u.
P_{MU}^{min}	0.2 p.u.	$SOC_{initial}$	1000 kWh
P_{SU}^{max}	1 p.u.	$T_{WT,min}$	30 minutes
P_{SU}^{min}	0.2 p.u.	$T_{PV,min}$	0
SOC_{max}	1 p.u.	$T_{DE,min}$	20 minutes
SOC_{high}	0.90 p.u.	$T_{BS,min}$	0
SOC_{low}	0.3 p.u.		

5. The Case of Study

5.1. The Example System. For the purposes of validating the proposed methodology, the simulations are executed in an actual islanded microgrid project, which was built to supply power for a new economic development experimental region in Dongao Island, China. The system configuration of Dongao microgrid is shown in Table 1, which was optimized based on the full consideration of the load demand and various distributed generation units such as WT, PV, DE, and BS based on the historical data. According to its practical operation performance, the unexpected problems have higher electricity costs and lower utilization of RES generations, when Dongao microgrid adopts an initial static operation strategy that just has the control mode 2 where DE always acts as the master unit, as discussed in Section 2.

In order to reflect the dynamic scheduling of microgrid better, this paper set the calculation cycle as 1 day, setting 5 min as a calculation period; then the whole day could be divided into 288 periods. The related parameters about MPSO were set as follows: particle population size was 50, dimensions were 30, and max iterations were 100 ($c_1 = c_2 = 2$, $w_h = 0.5$, and $w_s = 0.05$). Moreover, based on the developed models, the parameters for each distributed generation unit in Dongao microgrid project are given in Table 2.

5.2. Results and Discussion

5.2.1. Comparative Analysis of the PSO Algorithm. In order to analyze the influence of the MPSO algorithm and the traditional PSO algorithm on the microgrid operation, combined with the proposed dynamic optimization process as shown in Figure 2, the results of adopting these two different algorithms to calculate the optimal models are shown in Table 3, and the convergence curves are shown in Figure 4.

Table 3 shows that the calculation time of the MPSO is shorter than that of the traditional PSO algorithm, because

Table 3: Result comparisons between the traditional PSO and the MPSO algorithm.

Optimization type	PSO	MPSO
Calculation time (second)	35.058	16.235
Average convergence iteration	47	15
Convergence value (¥)	90684.3051	90632.5133

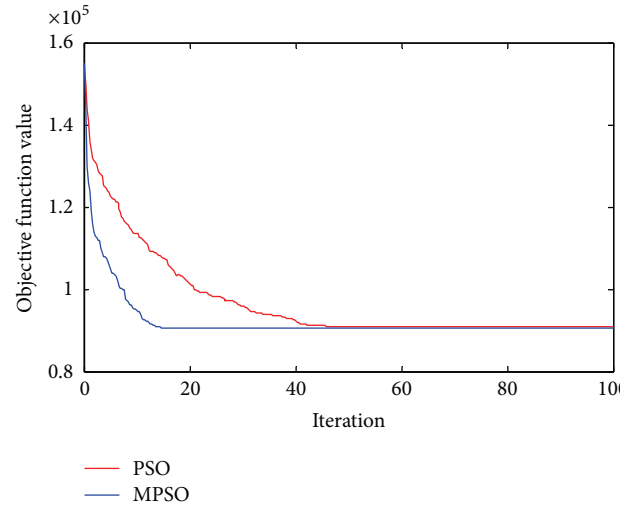


Figure 4: Convergence characteristics of PSO and MPSO.

the adaptive inertia weight effectively improves the convergence speed. It can also be seen from Figure 4 that the convergence rate of the MPSO is faster than the traditional PSO algorithm, and it converges at about 15th generation. Moreover, the MPSO obtained the optimum result compared to the traditional PSO algorithm. In conclusion, the MPSO algorithm has the more rapid convergence rate and more accurate convergence value to resolve the optimum operation of islanded microgrid.

5.2.2. Comparative Analysis of the Operation Strategy. In this paper, we adopt the dynamic optimization strategy with mode 1 and mode 2 where DE and BS alternate as the sole master unit as to minimize the generating cost and improve RES utilization of islanded microgrid over the entire dispatch period. In order to analyze the impact of the proposed dynamic strategy on the economic dispatch of microgrid, the results of adopting both the proposed dynamic strategy and the original static strategy to calculate the model of scheduling are shown in Table 4, and their economic operation performances are demonstrated in Figures 5 and 6, respectively. Please note that the output power curves of WT01, WT02 and 3 sets of PVs are not shown because they are all absorbed or not changed during the economic operation.

We can see from Table 4 that when we adopt the dynamic operation strategy, the total cost is much less than that under the static strategy and the daily cost saving is RMB5047.5858. Thus, the annual cost reduction will be RMB181713.0888 because the ratio of similar weather is approximately 10%

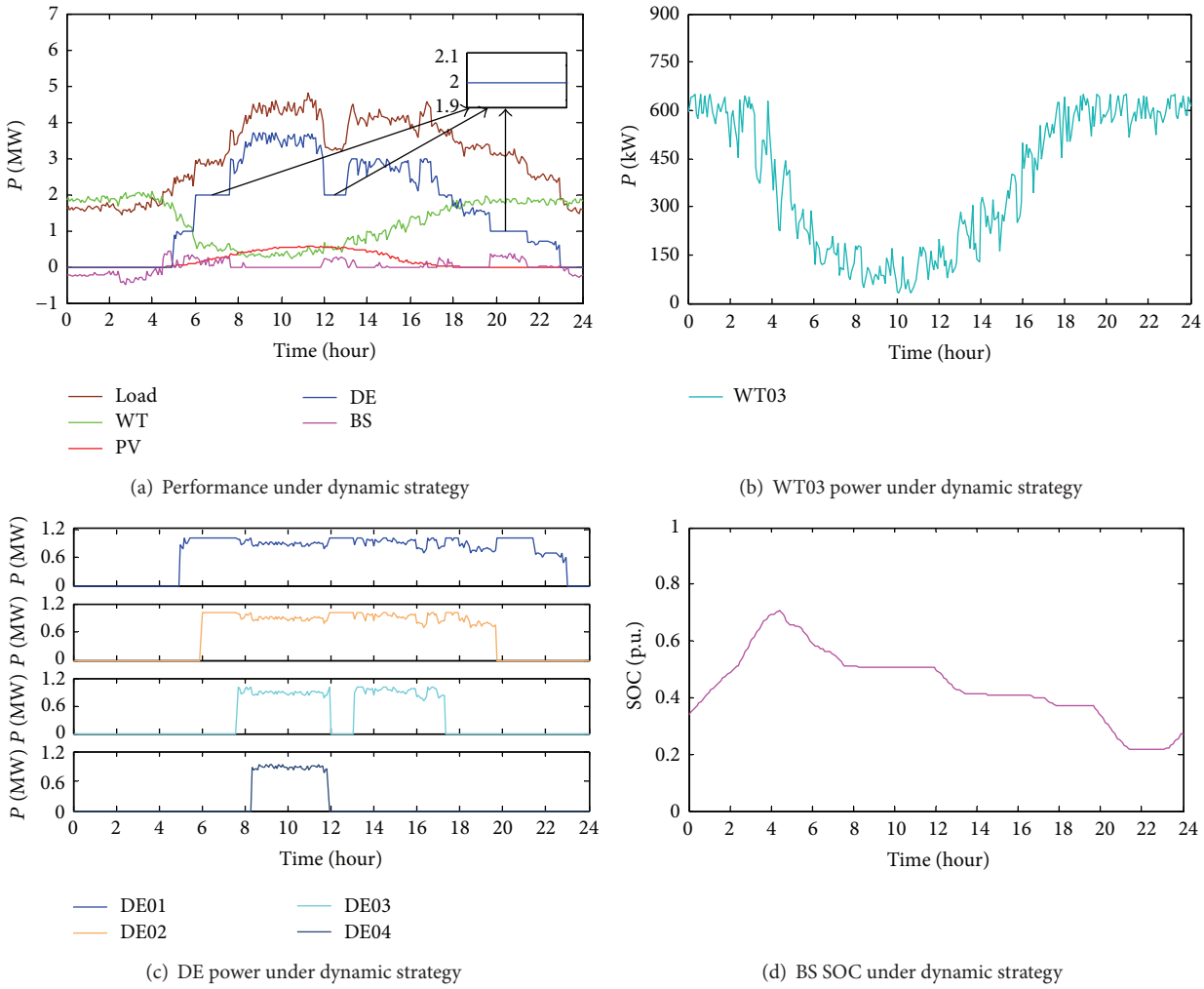


Figure 5: Optimal economic operation under dynamic strategy.

Table 4: Result comparisons between two operation strategies.

Optimization results	Dynamic strategy	Static strategy
Total cost (¥)	90632.5133	95680.099
RES utilization (%)	100%	96.59%
Cost saving (¥)	5047.5858	

calculated based on historical data collected on Dongao Island in 2013. Combined with Figures 5 and 6, it is easy to know the root cause of their different cost levels.

Figure 5 shows detailed operation performances of various distributed generation units under the proposed dynamic operation strategy, in which BS and DE alternate to operate as the master unit frequently according to different operation conditions. Figure 5(b) shows a continuous generation of WT, which means a satisfied RES utilization effect. Combining Figures 5(a) and 5(c), it is easy to know the following. (i) While BS runs as the master unit, it absorbs the surplus RES power when RES power exceeds the load demand during 0:00am–4:55am and 23:05pm–24:00pm, and accordingly DE is off. In addition, BS also runs the peak shaving function to meet buffer instantaneous fluctuations around the net load, and meanwhile DE operates at full power during 5:25am–7:35am, 11:55am–13:00pm, 17:20pm–17:55pm, and 19:45pm–21:25pm. (ii) While DE acts as the master unit during other time periods, it operates at the higher power level as to reduce fuel and emission, and sometimes it should get some necessary power supplement from BS if needed. The SOC curve of BS also shows this clearly in Figure 5(d). In short, the dynamic strategy achieves the higher RES utilization and the higher power level of DE, as well as reduction of DE run-time according to different conditions, which can further reduce the system generating cost.

Figure 6 demonstrates detailed operation performances under the original static strategy, in which DE always operates as the master unit. It is obvious to see that the static strategy causes the system to abandon/dump some wind power and to make DE run at lower power level in some times, which are shown in Figures 6(b) and 6(c). Moreover, combined with Figure 6(a), we also know that BS can assist DE to improve its output power level to a certain extent as to reduce fuel and

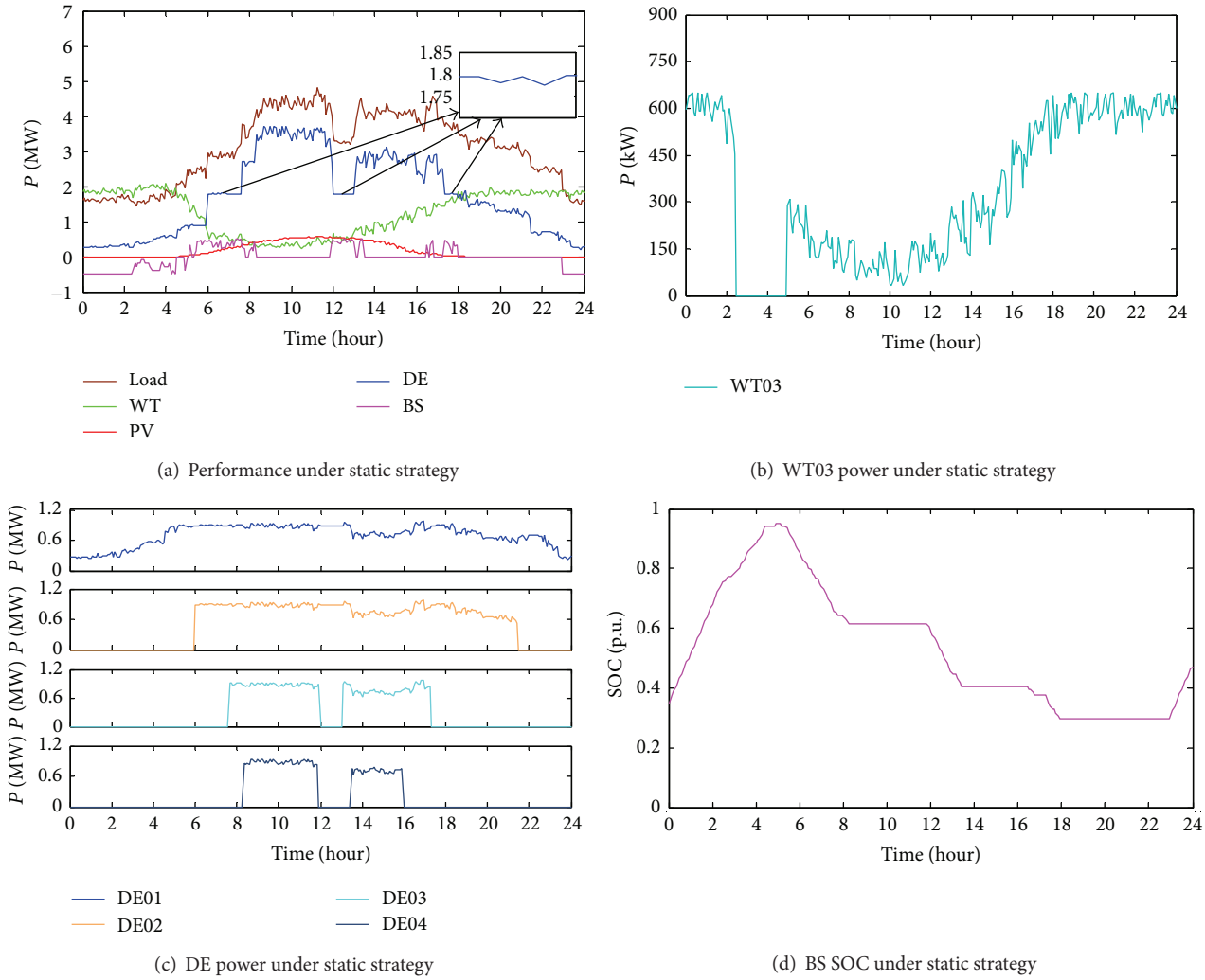


Figure 6: Optimal economic operation under static strategy.

emission. However, DE could not operate at the full power because of the necessary spinning reserve to cover system power fluctuation, indicated in (8). As a result, the higher system generating cost occurs, resulting from the lower RES utilization as well as the more fuel and emission of DE.

Based on the above, it can be seen that the operating strategy has a critical influence upon the actual economic operation performance of islanded microgrid. Compared to the static strategy, the proposed dynamic strategy can maximize BS advantages to improve the RES utilization and help DE reduce run-time and run at higher power level correspondingly, further reducing DE fuel cost and emission cost so as to minimize the system generating cost of microgrid as far as possible. The root cause is to set BS as the master unit to further reduce system generating cost, which can not only just utilize RES power to meet load but also run the system peak shaving function as to minimize DE fuel and emission. In addition, the results show the following. (i) RES units have the lowest generating cost as to be priority scheduling, compared to DE and BS. (ii) There is a set point between the generating cost of DE and the wear cost of BS; when DE output power is higher than this set point, its generating cost is less than the wear cost of BS so as to be scheduled prior to BS. If not, BS has a higher priority than DE.

6. Conclusion

This paper proposes an optimal economic operation method for islanded microgrid to attain a joint-optimization of cost reduction and operation strategy that determine the type of master unit to maintain the stability of system frequency and voltage. The time-series dynamic optimization process is designed according to two different master units of DE and BS, as well as the discrete optimization characteristics of economic operation in islanded microgrid. The MPSO algorithm presented is capable of efficiently searching an optimum trade-off between the minimum generating cost and the corresponding control mode of islanded microgrid. The proposed models and MPSO-based dynamic optimization strategy are verified by case studies based on a real islanded

microgrid in Dongao Island, China. The methodology presented effectively improves the RES generation utilization and the operational life of BS, as well as minimizing the fuel consumption cost and pollution emission cost resulting from DE.

Conflict of Interests

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

Acknowledgments

This study was supported jointly by the National High Technology Research and Development Program (863 Program, 2014AA052001), the Science and Technology Planning Project of Guangdong Province (2012B040303005), the Guangzhou Nansha District Science and Technology Project (2013C006), and the China Southern Power Grid Company Limited Science and Technology Project (K-KY2014-009).

References

[1] N. Hatziargyriou, H. Asano, R. Iravani, and C. Marnay, "Microgrids: an overview of ongoing research, development, and demonstration projects," *IEEE Power and Energy Magazine*, vol. 5, no. 4, pp. 78–94, 2007.

[2] A. N. Celik, "Optimisation and techno-economic analysis of autonomous photovoltaic–wind hybrid energy systems in comparison to single photovoltaic and wind systems," *Energy Conversion and Management*, vol. 43, no. 18, pp. 2453–2468, 2002.

[3] A. M. Eltamaly and M. A. Mohamed, "A novel design and optimization software for autonomous PV/wind/battery hybrid power systems," *Mathematical Problems in Engineering*, vol. 2014, Article ID 637174, 16 pages, 2014.

[4] B. Zhao, X. Zhang, P. Li, K. Wang, M. Xue, and C. Wang, "Optimal sizing, operating strategy and operational experience of a stand-alone microgrid on Dongfushan Island," *Applied Energy*, vol. 113, pp. 1656–1666, 2014.

[5] J. P. S. Catalão, *Electric Power Systems: Advanced Forecasting Techniques and Optimal Generation Scheduling*, CRC Press, New York, NY, USA, 2012.

[6] M. G. Ippolito, M. L. Di Silvestre, E. R. Sanseverino, G. Zizzo, and G. Graditi, "Multi-objective optimized management of electrical energy storage systems in an islanded network with renewable energy sources under different design scenarios," *Energy*, vol. 64, pp. 648–662, 2014.

[7] Y.-H. Chen, S.-Y. Lu, Y.-R. Chang, T.-T. Lee, and M.-C. Hu, "Economic analysis and optimal energy management models for microgrid systems: a case study in Taiwan," *Applied Energy*, vol. 103, pp. 145–154, 2013.

[8] M. Parvizimosaed, F. Farmani, and A. Anvari-Moghaddam, "Optimal energy management of a micro-grid with renewable energy resources and demand response," *Journal of Renewable and Sustainable Energy*, vol. 5, Article ID 053148, pp. 1–15, 2013.

[9] A. Dukpa, I. Duggal, B. Venkatesh, and L. Chang, "Optimal participation and risk mitigation of wind generators in an electricity market," *IET Renewable Power Generation*, vol. 4, no. 2, pp. 165–175, 2010.

[10] H. Ren, W. Zhou, K. Nakagami, W. Gao, and Q. Wu, "Multi-objective optimization for the operation of distributed energy systems considering economic and environmental aspects," *Applied Energy*, vol. 87, no. 12, pp. 3642–3651, 2010.

[11] H. Vahedi, R. Noroozian, and S. H. Hosseini, "Optimal management of MicroGrid using differential evolution approach," in *Proceedings of the 7th International Conference on the European Energy Market (EEM '10)*, pp. 1–6, IEEE, Madrid, Spain, June 2010.

[12] C. D. Barley and C. B. Winn, "Optimal dispatch strategy in remote hybrid power systems," *Solar Energy*, vol. 58, no. 4-6, pp. 165–179, 1996.

[13] M. Ashari and C. V. Nayar, "An optimum dispatch strategy using set points for a photovoltaic (PV)-diesel-battery hybrid power system," *Solar Energy*, vol. 66, no. 1, pp. 1–9, 1999.

[14] A. Gupta, R. P. Saini, and M. P. Sharma, "Modelling of hybrid energy system-Part II: combined dispatch strategies and solution algorithm," *Renewable Energy*, vol. 36, no. 2, pp. 466–473, 2011.

[15] O. Ekren and B. Y. Ekren, "Size optimization of a PV/wind hybrid energy conversion system with battery storage using simulated annealing," *Applied Energy*, vol. 87, no. 2, pp. 592–598, 2010.

[16] H. Yang, W. Zhou, L. Lu, and Z. Fang, "Optimal sizing method for stand-alone hybrid solar-wind system with LPSP technology by using genetic algorithm," *Solar Energy*, vol. 82, no. 4, pp. 354–367, 2008.

[17] N. Cai and J. Mitra, "A multi-level control architecture for master-slave organized microgrids with power electronic interfaces," *Electric Power Systems Research*, vol. 109, pp. 8–19, 2014.

[18] J. A. P. Lopes, C. L. Moreira, and A. G. Madureira, "Defining control strategies for microgrids islanded operation," *IEEE Transactions on Power Systems*, vol. 21, no. 2, pp. 916–924, 2006.

[19] L. Guo, X. Fu, X. Li, and C. Wang, "Coordinated control of battery storage and diesel generators in isolated AC microgrid systems," *Proceedings of the Chinese Society of Electrical Engineering*, vol. 32, no. 25, pp. 70–78, 2012.

[20] O. Skarstein and K. Uhlen, "Design considerations with respect to long-term diesel saving in wind/diesel plants," *Wind Engineering*, vol. 13, no. 2, pp. 72–87, 1989.

[21] W. Morgantown, "Emission rates for new DG technologies," The Regulatory Assistance Project [EB/OL], 2001, http://www.raponline.org/document/download/id/66.

[22] G. Cau, D. Cocco, M. Petrollese, S. Knudsen Kær, and C. Milan, "Energy management strategy based on short-term generation scheduling for a renewable microgrid using a hydrogen storage system," *Energy Conversion and Management*, vol. 87, pp. 820–831, 2014.

[23] X. Xia and A. M. Elaiw, "Optimal dynamic economic dispatch of generation: a review," *Electric Power Systems Research*, vol. 80, no. 8, pp. 975–986, 2010.

[24] R. Dufo-López, J. L. Bernal-Agustín, and J. Contreras, "Optimization of control strategies for stand-alone renewable energy systems with hydrogen storage," *Renewable Energy*, vol. 32, no. 7, pp. 1102–1126, 2007.

[25] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Ausitalia, December 1995.

[26] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on*

World Congress on Computational Intelligence, pp. 69–73, IEEE, Anchorage, Alaska, USA, May 1998.

[27] X. Yang, J. Yuan, J. Yuan, and H. Mao, "A modified particle swarm optimizer with dynamic adaptation," *Applied Mathematics and Computation*, vol. 189, no. 2, pp. 1205–1213, 2007.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 914108, 13 pages
http://dx.doi.org/10.1155/2015/914108

Research Article

A Network-Based Data Envelope Analysis Model in a Dynamic Balanced Score Card

Mojtaba Akbarian,[1] Esmaeil Najafi,[1] Reza Tavakkoli-Moghaddam,[2] and Farhad Hosseinzadeh-Lotfi[3]

[1]*Department of Industrial Engineering, Islamic Azad University, Science and Research Branch, Tehran, Iran*
[2]*School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran*
[3]*Department of Mathematics, Islamic Azad University, Science and Research Branch, Tehran, Iran*

Correspondence should be addressed to Mojtaba Akbarian; mojtabaakbaryan@gmail.com

Received 17 October 2014; Revised 20 December 2014; Accepted 2 March 2015

Academic Editor: Pandian Vasant

Performance assessment during the time and along with strategies is the most important requirements of top managers. To assess the performance, a balanced score card (BSC) along with strategic goals and a data envelopment analysis (DEA) are used as powerful qualitative and quantitative tools, respectively. By integrating these two models, their strengths are used and their weaknesses are removed. In this paper, an integrated framework of the BSC and DEA models is proposed for measuring the efficiency during the time and along with strategies based on the time delay of the lag key performance indicators (KPIs) of the BSC model. The causal relationships during the time among perspectives of the BSC model are drawn as dynamic BSC at first. Then, after identifying the network-DEA structure, a new objective function for measuring the efficiency of nine subsidiary refineries of the National Iranian Oil Refining and Distribution Company (NIORDC) during the time and along with strategies is developed.

1. Introduction

Performance management is one of the most important issues, in which if the organizational vision is used, future decision making of an organization will be achieved along with strategic goals. There are a number of performance assessment tools in the literature; however, the balanced score card (BSC) is used only along with strategic goals. By implementing the BSC, organizations should translate their visions to strategic objectives, link their vision to individual performance, and measure their performance along with their visions [1].

The BSC methodology is a performance management system for today's successful organizations. It indicates the organizational mission and vision in a set of cause-and-effect relationships in four perspectives (i.e., financial, customer, internal processes and learning, and growth) [2]. The BSC innovators introduced cause-and-effect relationships based on key performance indicators (KPIs) to link strategies to vision. The chains of the cause-and-effect relation connect all the factors with performance indicators through the four perspectives of the BSC model, which reflect dynamically the change of strategies and indicate how an organization creates its value [3].

The BSC model controls the vision and strategies by performance indicators. The financial indicators are insufficient in explaining the performance because they only contain the information which has taken place in the past. Then, Kaplan and Norton introduced the BSC system, which integrates the indicators regarding the past performance with the indicators regarding the elements that will bring future performances. By using the BSC, balance between performance drivers, named lead KPIs, and outcome measures, named lag KPIs, is created. The lead KPIs communicate the way to achieve vision and indicate early whether strategies are being implemented successfully. The lag KPIs may enable organization to accomplish long-term operational improvements and enhance financial performance. The ideal BSC should have an appropriate mix of the lead KPIs and the lag KPIs tailored to the business unit's strategy [4].

Continuous performance assessment during the time and along with strategic goals is desirable for top managers of an organization. It is suitable for continuous control of any organization. Performance assessment in the BSC is based on lag and lead KPIs. Due to delay of lag KPIs in the BSC, relationships between four perspectives are not simple and dynamics in the BSC perspectives appeared. Despite the capability of the BSC for evaluating the performance based on KPIs, the lack of quantitative methods for measuring the performance is a major weakness. To tackle this weakness, the DEA model can be considered as one of the best mathematical methods to calculate the performance. Despite its capability for evaluating the performance, inability to determine the input and output variables is a main weakness.

The weakness of input and output variables in the DEA model is removed through KPIs of the BSC model and the lack of the quantitative method for measuring the performance is also removed through the DEA model. By integrating these BSC and DEA models, their weaknesses are removed and the synergy for the performance assessment during the time is achieved. According to the above-mentioned points, integration of two performance assessment models during the time and along with strategic goals is the main purpose of this paper.

Many studies have been done on the simultaneous use of the DEA and BSC models for performance assessment. Rouse et al. [5] first studied a DEA analysis integrated with the BSC model. They used the DEA model to measure the efficiency over time and then used four perspectives of the BSC model as variables of the DEA model. Eilat et al. [6] used mixed DEA, BSC, and branch-and-bound algorithm in order to develop their previous study [7] proposing a framework for selection of the R&D projects. Chen et al. [8] used an integrated DEA and BSC model to measure the selection of KPI results at a credit cooperative bank in Taiwan with four models. The first model includes input and output variables, the second model includes BSC indicators, the third model includes BSC risk indicators, and the fourth model includes traditional financial indicators. García-Valderrama et al. [9] and Chiang and Lin [10] developed a DEA model to compare the tradeoffs between financial and nonfinancial KPIs in the BSC by considering the KPI of the BSC model as a variable in the DEA model.

Maced et al. [11] used an integrated DEA and BSC model with six KPIs based on six BSC perspectives as variables in order to evaluate the performance of bank branches in Brazil. Asosheh et al. [12] used integrated BSC and DEA models to measure the efficiency of projects in Iran Ministry of Science, Research and Technology with the fuzzy data. Amado et al. [13] used DEA and BSC models to measure the performance of a multinational company. Most studies have used the KPIs of BSC model as input and output variables in the classic DEA model.

Because of time delay in the lag KPIs of the BSC model, considering the time factor for performance assessment is essential. To consider the time delay resulting from the lag KPIs, the dynamic cause-and-effect relation of the BSC model in different periods based on judgment of experts must be determined.

Different studies to draw causal relationships between perspectives of BSC based on judgment are done. The DEMATEL model [14–17], DEMATEL model with fuzzy data [18, 19], and the DEMATEL and ANP model [20, 21] are used to identify the causal relationships among strategic objectives. Apart from these, similar studies have been conducted on drawing cause-and-effect relationships. For example, Jeng and Tzeng [22] used the fuzzy DEMATEL model to discover the causal relationship between the important variables. Ren et al. [23] used the DEMATEL model to recognize the cause-and-effect relation to improve the sustainability of a hydrogen supply chain. Horng et al. [24] used the DEMATEL model to identify the critical standard and draw the cause-and-effect relation between them for the future restaurant space design.

By revealing the cause-and-effect relation during the time of the BSC model, the classic DEA model is not suitable quantitative one to measure the efficiency. Thus, a network-based DEA model is needed. Different studies on this network model have been carried out. The reader may read more details in Kao [25]. The basic two-stage is the simplest network structure, in which all inputs from outside are supplied to the first process to produce intermediate products for the second process to produce the final outputs. Each stage is named process and total structure is named system.

Seiford and Zhu [26] measured the performance of US commercial banks by using a basic two-stage structure. Based on this work, a lot of studies have been carried out. Yang [27] used an average model to measure the process efficiencies of insurance companies in Canada. Kao and Hwang [28] proposed a relational model with a product of two processes as objective function to measure the system efficiency of nonlife insurance companies in Taiwan. Chiou et al. [29] considered the average of two processes as objective function to measure the efficiencies of bus companies in Taiwan. Chen et al. [30] used a system distance model with a weighted average approach as objective function considering projection of the intermediate variable to measure the efficiency.

Wang and Chin [31] showed that if the weight of each process is defined as the aggregate output of the process in that of two processes, then the aggregate efficiency is a weighted harmonic average of the process efficiencies. Kao and Hwang [32] used BCC input and output models for efficiency measurement of the first and second process models, respectively. Then, they decomposed the system efficiency as a product of the technical and scale efficiencies. Lewis et al. [33] minimized the input parameter and maximized the output parameter to measure the performance of Major League Baseball teams.

A general two-stage structure allows both stages to consume exogenous inputs supplied from outside. For the first time, Charnes et al. [34] used this structure for measuring the efficiency of each process treated as a decision making unit (DMU). Kao and Hwang [35] extended their previous model [28] for the general two-stage structure that the inputs were shared with the second process. Chen et al. [36] used a weighted average of the two process efficiencies as objective function. Liang et al. [37] developed two models to measure the efficiency. One model is to maximize the average of

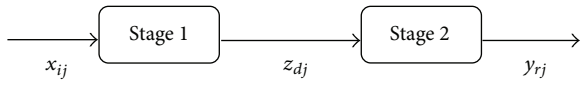


Figure 1: Basic two-stage structure.

the two process efficiencies, and the other model is to give a higher priority to one process. Xie et al. [38] minimized a weighted input process distance function for efficiency measurement of regions in China. Lozano et al. [39] used a distance function for measuring the performance of Spanish airports.

A series structure appeared with generalization of the general two-stage structure considering a number of processes connected in sequence, in which each process consumes the exogenous inputs and intermediate products by the preceding process. Nouri et al. [40] defined series structure with five stages for measuring the system and process efficiencies of plants in Iran. Kao and Liu [41] used multiperiod system without intermediate products for performance measurement of commercial banks in Taiwan with this structure.

Another type of the network is parallel structure that is the same as the multiperiod system; however, input or output variables can be shared. Kao [42] proposed a parallel system to measure the system and process efficiencies with the weighted average of the component process efficiencies as objective function. Bi et al. [43] proposed the parallel structure, in which the inputs were shared and the outputs were the contribution of both processes.

There are network systems whose structures are neither series nor parallel, but a mixture of them, named mixed structure. Chen and Yan [20] defined a mixed three-stage structure for performance measurement of supply chains, in which the system efficiency was the product of the process efficiency. Adler et al. [44] defined a mixed two-stage structure, in which the first and second stages had one and two processes for the performance assessment of airports in European countries, respectively. Wang et al. [45] used a mixed two-stage structure, in which the first stage divides to two parallel processes and the objective function minimizes the weighted average of the input and output distance parameters of the first and second stages for efficiencies measurement of high-tech firms in Taiwan, respectively. Lin and Chiu [46] defined a mixed three-stage structure, in which the second stage is composed of two parallel subprocesses for the Taiwanese domestic bank's performance assessment.

With regard to the above division of the existing network DEA models in the previous section, a basic two-stage structure of the network DEA model as basic model of this paper is described in Figure 1, where all inputs (X_i) from outside are supplied to the first process to produce intermediate products (Z_d) for the second process to produce the final outputs (Y_r).

Chen et al. [30] proposed a weighted additive with the arithmetic mean approach named additive model. Components of a two-stage process and proposed overall efficiency of the two-stage process are given by

$$w_1 \cdot \frac{\sum_{d=1}^{D} \gamma_d z_{dp}}{\sum_{i=1}^{m} v_i x_{ip}} + w_2 \cdot \frac{\sum_{r=1}^{s} u_r y_{rp}}{\sum_{d=1}^{D} \gamma_d z_{dp}}, \tag{1}$$

where $w_1 + w_2 = 1$. These weights are not decision variables, but rather are functions of decision variables. By setting for stage p of DMU_p, w_p is the proportion of the total resources for the process that are devoted to stage p, reflecting the relative size of that stage as follows:

$$w_p = \frac{(\text{component } p \text{ input})}{(\text{total input accross all components})}. \tag{2}$$

Then, for the basic two-stage structure, we have

$$\begin{aligned} w_1 &= \frac{\sum_{i=1}^{m} v_i x_{ip}}{\sum_{i=1}^{m} v_i x_{ip} + \sum_{d=1}^{D} \gamma_d z_{dp}} \\ w_2 &= \frac{\sum_{d=1}^{D} \gamma_d z_{dp}}{\sum_{i=1}^{m} v_i x_{ip} + \sum_{d=1}^{D} \gamma_d z_{dp}}. \end{aligned} \tag{3}$$

The overall efficiency of the process is calculated by solving the following nonlinear problem, named additive model as follows:

$$\begin{aligned} \max z = w_1 \cdot \frac{\sum_{d=1}^{D} \gamma_d z_{dp}}{\sum_{i=1}^{m} v_i x_{ip}} + w_2 \cdot \frac{\sum_{r=1}^{s} u_r y_{rp}}{\sum_{d=1}^{D} \gamma_d z_{dp}} \\ \frac{\sum_{d=1}^{D} \gamma_d z_{dj}}{\sum_{i=1}^{m} v_i x_{ij}} \le 1 \\ \frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{d=1}^{D} \gamma_d z_{dj}} \le 1 \\ \gamma_d, v_i, u_r \ge 0, \quad j = 1, 2, \ldots, n. \end{aligned} \tag{4}$$

In this paper, based on this basic structure of network-DEA, a new model for efficiency measurement is developed.

By considering a time factor resulting from the lag KPIs of the BSC model, the time series analysis can be used. Klepac introduced the REFII model to automate the time series analysis, through a unique transformation model of time series in 2005. It is an authorial mathematical model for time series data mining. Furthermore, he used it to evaluate risk in an insurance company [47].

2. Materials and Methods

2.1. Proposed Framework. Deployment of the BSC model is not the purpose of this paper; however, efficiency measurement during the time and along with strategic goals integrating of the BSC and DEA models is the main purpose of this paper. By the simultaneous use of these two models, the purpose of this paper is followed. Despite various studies on the simultaneous use of the BSC and DEA models, almost

all studies used KPIs of the BSC model as input and output variables in the classic DEA model. These studies did not consider a dynamic factor resulting from the delay of the lag KPIs as dynamic variables with severance of the lead and lag KPIs.

To consider the time factor, in addition to identifying causal relationships of the BSC perspectives at a time period, dynamic causal relationships of BSC perspectives at different periods should also be considered.

After dividing the KPIs into four perspectives of the BSC model and identifying the lead and lag KPIs, dynamic and nondynamic relationships between perspectives of the BSC resulting from lag and lead KPIs are determined based on expert judgment, respectively.

Several studies have been carried out to draw the cause-and-effect relations among perspectives of the BSC model with expert judgment. However, none of them has considered delay of the lag KPIs. In the first stage of the proposed framework, the dynamic and nondynamic cause-and-effect relations considering the lag and lead KPIs between perspectives of the BSC model are drawn with fuzzy DEMATEL model, respectively. For identifying dynamic causal relationships, the relation matrix between perspectives of BSC with attention to the lead and lag KPIs are created in a dynamic pattern as shown in Figure 3. The relationships in the perspectives of the same period and between perspectives of different periods are shown in gray and white cells, respectively. This matrix is created by using expert judgment and considering relationship of the lag and lead KPIs. The linguistic variables are used to decrease obscurity of expert judgment.

In the second stage of the proposed framework, fuzzy DEMATEL model is used to determine the dynamic and nondynamic cause-and-effect relations between perspectives of different and same periods, respectively. Then, after defuzzification of fuzzy DEMATEL calculation, dynamic cause-and-effect relations between perspectives in different periods are drawn as dynamic BSC.

The BSC model evaluates the efficiency along with strategy. Hence, for measuring the efficiency along with a strategy and during the time, a DEA mathematical method should be added to the dynamic BSC. Then, after identifying causal networks of BSC perspectives, a classic DEA model is not suitable. Thus, a network-DEA model is required. Despite several studies have been done on the network DEA as reviewed in the previous section, none of them has not considered network relations of the third level. In the third stage of the proposed framework considering third level, a new network DEA structure for measuring the efficiency is introduced. Then, dynamic and nondynamic variables resulting from the lag and lead KPIs are defined as variables, respectively.

Chen et al. [30] used the weighted additive with the arithmetic mean approach of the second level. However, by defining the third level in the proposed network DEA structure, a new objective function is defined. Then, modeling for optimization of the proposed structure is done. Finally, in the sixth stage, the efficiency of decision making units based on the data set of the KPIs of BSC model as variables is measured. By deploying this proposed framework,

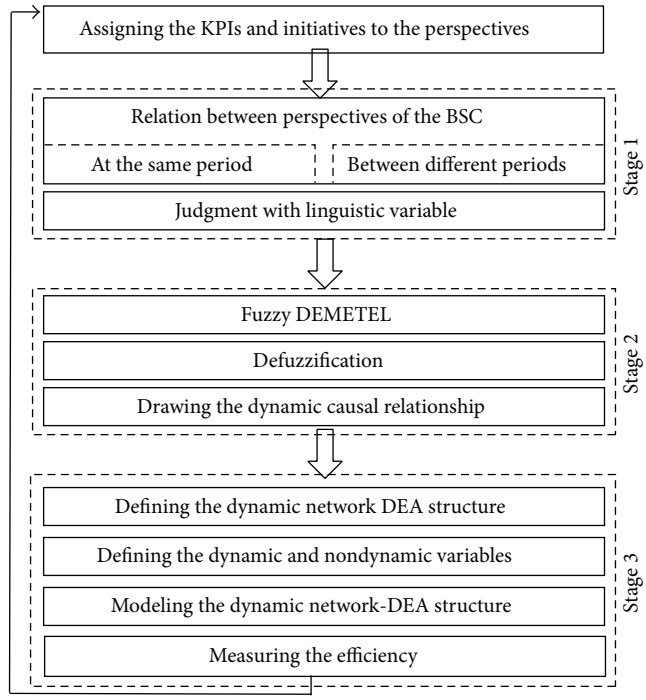


Figure 2: Proposed framework.

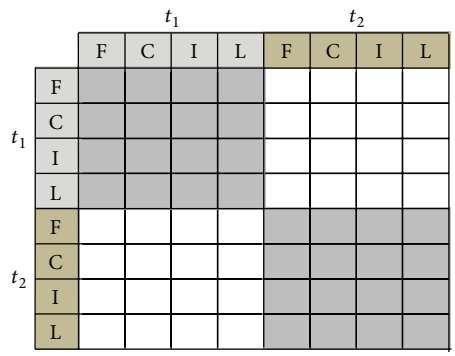


Figure 3: Dynamic relation matrix.

the efficiency of DMUs during the times and along with strategic goals is measured. After identifying the efficiency of DMUs, the strategic initiatives are redefined to improve the efficiency scores with feedback lines as shown in Figure 2.

2.2. Case Analysis. The National Iranian Oil Refining and Distribution Company (NIORDC) was established on the principle of separating upstream activities, which has nine oil existing refineries. These companies have performed National Iranian Oil Company's activities in the area of all activities related to crude oil transportation to the refineries and export jetties, processing, production and distribution of numerous oil main products and byproducts throughout Iran, marketing and exporting special products surplus, construction of refineries, marine platforms, pipelines, and communication networks, and ensuring internal and long

Table 1: KPIs of the BSC model.

Perspectives	Key performance indicators	Type	Strategic initiative
Financial	Net profit margin	Lead	Productivity cycle project
	Annual budget	Lag	Activity based costing
	Strategic plans	Lag	Strategic thinking for managers
Customer	Increase of refinery capacity	Lag	ISO 10004
	Quality of oil products	Lead	ISO 10002
Internal process	Material assigned to refinery	Lead	Total maintenance management
	Gas production in the refinery	Lead	ISO TS29001
	Safety stock for refinery	Lead	
	Outsourcing	Lag	Outsourcing management
Learning and growth	Training	Lead	ISO 10015
	Effectiveness of training	Lag	
	Management systems	Lead	Leadership style management
	Customers portal changes	Lag	Information security system

distance, industrial and official headquarters of Oil Ministry communication with the extreme capability.

In this paper, the efficiency of nine oil existing refineries of the NIORDC is measured as case analysis. The KPIs of the NIORDC is divided into four perspectives of the BSC model based on oil expert comments in Table 1. The strategic initiatives are also assigned to perspectives for improving them. The result of the lag KPIs occurs with time delay because they introduce the past performance. The lead KPIs do not occur with time delay because they introduce elements that will bring future performances. The lead and lag KPIs identified in the Table 1 help oil experts to have judgment. For improving the perspectives, the strategic initiatives are defined for them.

2.2.1. Relation between Perspectives of the BSC. In this stage, a dynamic relation matrix with attention to the lead and lag KPIs between perspectives of a period and different periods is identified, respectively. The lag KPIs occurs with delay. For this reason, the cause-and-effect relations between perspectives in different periods of the BSC model are created with delay. The lead KPIs do not have delay and create a nondynamic cause-and-effect relations between perspectives at the same period without any delay.

According to the pattern presented in Figure 3, three groups of oil experts judge the relationship of the BSC perspectives in different and the same periods with attention to the lag and lead KPIs with linguistic variables, respectively. To ensure a correct judgment of oil experts, the result of the lead and lag KPIs corresponding to the BSC perspectives is presented for them. Then with awareness of the NIORDC lead and lag KPIs, the relationship between the perspectives is judged. Due to space limitations, description of the lead and lag KPIs is ignored. Linguistic judgment of the first oil expert group on a dynamic relation matrix is shown in Table 2. The perspectives of the BSC model are presented as financial (F), customer (C), internal processes (I), and learning and growth (L).

2.2.2. Drawing the Dynamic Causal Relationship. By adopting a fuzzy triangular number and linguistic values as shown in Table 3, a fuzzy DEMATEL model is used by expressing

Table 2: Dynamic linguistic assessment of oil experts.

		$t1$				$t2$			
		F	C	I	L	F	C	I	L
$t1$	F	No	No	L	No	VL	VL	L	H
	C	VH	L	No	No	VH	VL	L	VL
	I	H	VH	No	No	H	VH	L	VL
	L	No	VL	VH	No	VL	L	H	H
$t2$	F	No	No	No	No	No	No	No	No
	C	No	No	No	No	VH	No	No	No
	I	No	No	No	No	H	VH	No	No
	L	No	No	No	No	No	VL	VH	No

Table 3: Linguistic values and corresponding terms.

Linguistic terms	Linguistic value
Very high influence (VH)	(0.75, 1, 1)
High influence (H)	(0.5, 0.75, 1)
Low influence (L)	(0.25, 0.5, 0.75)
Very low influence (VL)	(0, 0.25, 0.5)
No influence (No)	(0, 0, 0.25)

different degrees of influences or causalities with five linguistic terms as {Very high, High, Low, Very low, No} and their corresponding triangular fuzzy numbers [48].

Suppose $\widetilde{Z}_{\widetilde{p}} = (l_{ij}^n, m_{ij}^n, u_{ij}^n)$ are triangular fuzzy numbers and are obtained from assessment of oil experts. The average of these assessments is accounted by using the following equation as shown in Table 4, which is called a dynamic relation fuzzy matrix:

$$\widetilde{Z} = \frac{\widetilde{z_1} \oplus \widetilde{z_2} \cdots \oplus \widetilde{z_p}}{P}. \tag{5}$$

By normalizing a dynamic relation fuzzy matrix, a normalized dynamic relation fuzzy matrix ($\widetilde{X}$) is calculated by

$$\widetilde{x}_{ij} = \frac{\widetilde{Z}_{ij}}{r} = \left[\frac{\widetilde{l}_{ij}}{r}, \frac{\widetilde{m}_{ij}}{r}, \frac{\widetilde{u}_{ij}}{r}\right]$$
$$r = \max_{1 \le i \le n}\left(\sum_{j=1}^{n} U_{ij}\right). \tag{6}$$

Table 4: Dynamic relation fuzzy matrix.

	F1			C1			I1			L1			F2			C2			I2			L2		
	L	M	U	L	M	U	L	M	U	L	M	U	L	M	U	L	M	U	L	M	U	L	M	U
F1	0	0	0.3	0	0	0.3	0.3	0.4	0.7	0	0	0.3	0	0.3	0.5	0	0.3	0.5	0.5	0.8	0.9	0.5	0.8	1
C1	0.8	1	1	0	0	0.3	0	0	0.3	0	0	0.3	0.8	1	1	0	0.3	0.5	0.2	0.4	0.7	0	0.3	0.5
I1	0.5	0.8	1	0.8	1	1	0.2	0.3	0.6	0	0	0.3	0.5	0.8	1	0.8	1	1	0.3	0.6	0.8	0.1	0.3	0.6
L1	0.1	0.2	0.4	0.1	0.3	0.6	0.8	1	1	0	0.1	0.3	0.3	0.5	0.7	0.3	0.5	0.8	0.7	0.9	1	0.5	0.8	1
F2	0	0	0.3	0	0	0.3	0	0	0.3	0	0	0.3	0	0	0.3	0	0	0.3	0.3	0.4	0.7	0	0	0.3
C2	0	0	0.3	0	0	0.3	0	0	0.3	0	0	0.3	0.8	1	1	0	0	0.3	0	0	0.3	0	0	0.3
I2	0	0	0.3	0	0	0.3	0	0	0.3	0	0	0.3	0.5	0.8	1	0.8	1	1	0.2	0.3	0.6	0	0	0.3
L2	0	0	0.3	0	0	0.3	0	0	0.3	0	0	0.3	0.1	0.2	0.4	0.1	0.3	0.6	0.8	1	1	0	0.1	0.3

Table 5: Fuzzy normalized dynamic relation matrix.

	F1			C1			I1			L1			F2			C2			I2			L2		
	L	M	U	L	M	U	L	M	U	L	M	U	L	M	U	L	M	U	L	M	U	L	M	U
F1	0	0	0	0	0	0	0	0.1	0.1	0	0	0	0	0	0.1	0	0	0.1	0.1	0.1	0.1	0.1	0.1	0.2
C1	0.1	0.2	0.2	0	0	0	0	0	0	0	0	0	0.1	0.2	0.2	0	0	0.1	0	0.1	0.1	0	0	0.1
I1	0.1	0.1	0.2	0.1	0.2	0.2	0	0.1	0.1	0	0	0	0.1	0.1	0.2	0.1	0.2	0.2	0.1	0.1	0.1	0	0.1	0.1
L1	0	0	0.1	0	0.1	0.1	0.1	0.2	0.2	0	0	0.1	0	0.1	0.1	0	0.1	0.1	0.1	0.1	0.2	0.1	0.1	0.2
F2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0.1	0	0	0
C2	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0.2	0.2	0	0	0	0	0	0	0	0	0
I2	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0.1	0.2	0.1	0.2	0.2	0	0.1	0.1	0	0	0
L2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0	0.1	0.1	0.1	0.2	0.2	0	0	0.1

It is assumed that at least one i exists such that $\sum_{j=1}^{n} U_{ij} < 1$ [48]. According to Table 4 and considering the mean of a fuzzy relation matrix and judgment of oil experts, 6.25 for r-value is defined. The fuzzy normalized dynamic relation matrix is illustrated in Tables 4 and 5. Next, the total fuzzy relation matrix is calculated by using (7) and the total dynamic fuzzy relation matrix is illustrated in Table 6:

$$\begin{gathered}
\widetilde{T} = \lim_{k \to \infty} \left(\widetilde{X}^1 + \widetilde{X}^2 + \cdots \widetilde{X}^k\right)^n \\
\widetilde{T} = \begin{bmatrix} \widetilde{t}_{11} & \cdots & \widetilde{t}_{1n} \\ \vdots & \ddots & \vdots \\ \widetilde{t}_{n1} & \cdots & \widetilde{t}_{nn} \end{bmatrix}, \qquad \widetilde{t}_{ij} = \left(l''_{ij}, m''_{ij}, u''_{ij}\right) \\
\left[l''_{ij}\right] = X_l * \left(I - X_l\right)^{-1} \\
\left[m''_{ij}\right] = X_m * \left(I - X_m\right)^{-1} \\
\left[u''_{ij}\right] = X_u * \left(I - X_u\right)^{-1}.
\end{gathered} \tag{7}$$

By producing the total dynamic fuzzy relation matrix, $\widetilde{D}_i + \widetilde{R}_i$ and $\widetilde{D}_i - \widetilde{R}_i$ are calculated, where $\widetilde{R}_i$ and $\widetilde{D}_i$ are the sum of rows and columns, respectively. To access the casual dynamic relationships between perspectives, $\widetilde{D}_i + \widetilde{R}_i$ and $\widetilde{D}_i - \widetilde{R}_i$ are calculated in our partial results in Table 6. To finalize the procedure, all calculated $\widetilde{D}_i + \widetilde{R}_i$ and $\widetilde{D}_i - \widetilde{R}_i$ and total dynamic fuzzy relation matrix are defuzzified through simple equation (8) as shown in Tables 7 and 8:

$$\text{Crisp}\left(\widetilde{T}\right) = \frac{(l + 2m + u)}{4}. \tag{8}$$

After it becomes clear that how perspectives of BSC impact each other during the time, identification of threshold to remove the minor impact is essential. This threshold is determined by using expert opinion and in some cases by mean of the total relation matrix elements [49].

In this paper, after getting out by the mean of the total relation matrix elements of Table 8 as a mental assistance to oil experts, the threshold value of 0.45 based on oil experts' consensus is considered. The elements that have relationship more than 0.45 are considered to be drawn as causal relationship. By considering dynamic cause-and-effect relations between four perspectives of the BSC model, network relationships are identified as dynamic BSC depicted in Figure 4.

2.2.3. Defining the Dynamic Network DEA Structure. According to the dynamic cause-and-effect relations of perspectives as shown in Figure 4, a dynamic network-DEA structure with dynamic and nondynamic variables for one DMU is depicted in Figure 5. According to this structure, efficiencies are measured in three levels. In the first level, named system, the overall efficiency is measured. In the second level, named process, the efficiency of each period is measured. In the third level, named activity, the efficiency of each perspective in a period is measured. The dynamic and nondynamic relations

Table 6: Total dynamic fuzzy relation matrix.

	F1			C1			I1			L1			F2			C2			I2			L2		
	L	M	U	L	M	U	L	M	U	L1	M	U	L	M	U	L	M	U	L	M	U	L	M	U
F1	0	0	0	0	0	0	0	0.1	0.1	0	0	0	0	0	0.1	0	0	0.1	0.1	0.1	0.2	0.1	0.1	0.2
C1	0.1	0.2	0.2	0	0	0	0	0	0	0	0	0	0.1	0.2	0.2	0	0	0.1	0	0.1	0.1	0	0	0.1
I1	0.1	0.1	0.2	0.1	0.2	0.2	0	0.1	0.1	0	0	0	0.1	0.1	0.2	0.1	0.2	0.2	0.1	0.1	0.2	0	0.1	0.1
L1	0	0	0.1	0	0.1	0.1	0.1	0.2	0.2	0	0	0.1	0	0.1	0.1	0	0.1	0.1	0.1	0.2	0.2	0.1	0.1	0.2
F2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0.1	0	0	0
C2	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0.2	0.2	0	0	0	0	0	0	0	0	0
I2	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0.1	0.2	0.1	0.2	0.2	0	0.1	0.1	0	0	0
L2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0	0.1	0.1	0.1	0.2	0.2	0	0	0.1

Table 7: Calculation for dynamic relationships between perspectives.

	$\widetilde{D}_i$			$\widetilde{R}$			$\widetilde{D}_i - \widetilde{R}$			$\widetilde{D}_i + \widetilde{R}_i$			$D + R$	$D - R$
	L	M	U	L1	M	U	L	M	U	L	M	U		
F1	0.2	0.4	0.8	0.2	0.4	0.7	−0	0.1	0.1	0.5	0.4	1.4	0.90094	0.39171
C1	0.3	0.5	0.8	0.1	0.2	0.5	0.2	0.3	0.3	0.5	0.5	1.3	0.98389	1.5617
I1	0.5	0.9	1.2	0.2	0.3	0.6	0.3	0.6	0.5	0.7	0.9	1.8	1.50274	3.08482
L1	0.5	0.8	1.1	0	0	0.3	0.5	0.8	0.7	0.5	0.8	1.4	1.23462	4.18843
F2	0	0.1	0.4	0.5	0.8	1.1	−0	−1	−1	0.5	0.1	1.5	0.58041	−4.0835
C2	0.1	0.2	0.5	0.3	0.6	0.9	−0	−0	−0	0.5	0.2	1.4	0.64646	−2.2553
I2	0.3	0.4	0.7	0.5	0.8	1.1	−0	−0	−0	0.7	0.4	1.8	1.01532	−2.3134
L2	0.2	0.3	0.6	0.2	0.4	0.8	−0	−0	−0	0.4	0.3	1.3	0.71044	−0.5745

Table 8: Total dynamic relation matrix.

	F1	C1	I1	L1	F2	C2	I2	L2
F1	0.042	0.042	0.447	0.042	0.254	0.254	0.804	0.823
C1	1.089	0.042	0.042	0.042	1.089	0.254	0.433	0.254
I1	0.823	1.089	0.356	0.042	0.823	1.089	0.622	0.342
L1	0.195	0.342	1.089	0.11	0.509	0.526	0.997	0.823
F2	0.042	0.042	0.042	0.042	0.042	0.042	0.447	0.042
C2	0.042	0.042	0.042	0.042	1.089	0.042	0.042	0.042
I2	0.042	0.042	0.042	0.042	0.823	1.089	0.356	0.042
L2	0.042	0.042	0.042	0.042	0.195	0.342	1.089	0.11

shown as dash and continuous lines in the dynamic network-DEA structure are the lag and lead KPIs of the BSC model, respectively.

By attention to dynamic network structure as shown in Figure 5, the first level is similar to the basic two-stage structure. It measures the overall efficiency. The external relations of perspectives among different periods are identified as dynamic relation in the second level. It is similar to general two-stage structure because it allows both processes to consume exogenous inputs and outputs from outside. The internal relations between perspectives in each period are shown in third level. It is similar to the mixed structure because of being connected by both series and parallel network relations.

2.2.4. Defining the Dynamic and Nondynamic Variables. The dynamic and nondynamic relations shown as dash and continuous lines are the lag and lead KPIs of the BSC model, respectively. Then, the data set of nine refineries KPIs as the DMU of the NOIRDC are illustrated in Table 9.

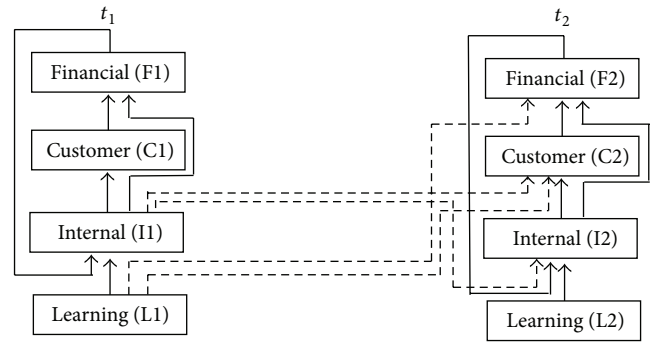


Figure 4: Cause-and-effect relations in the dynamic BSC.

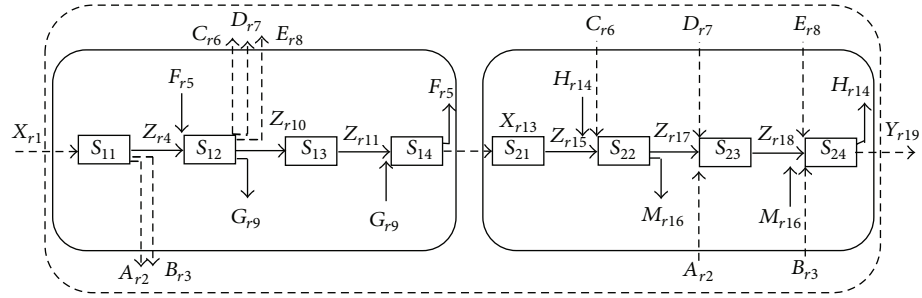


Figure 5: Dynamic network-DEA structure in the NIORDC.

Table 9: Data set of the NIORDC.

Key performance indicators	Variables	DMU1	DMU2	DMU3	DMU4	DMU5	DMU6	DMU7	DMU8	DMU9
Annual budget	Xr1	94	99	58	78	98	99	74	70	92
Customers portal changes	Ar2	100	100	100	68	91	30	40	30	85
Effective measuring of training	Br3	90	88	83	68	90	31	42	29	80
Training in first period	Zr4	97	92	97	93	97	92	89	91	99
Management systems in first period	Fr5	73	40	47	87	53	80	40	40	53
Increase refinery capacity	Cr6	94	99	58	78	98	99	74	70	92
Strategic plans	Dr7	100	100	100	68	91	30	40	30	85
Outsourcing	Er8	90	88	83	68	90	31	42	29	80
Material assigned to refinery in first period	Gr9	90	20	156	20	98	5	8	8	9
Gas production in the refinery in first period	Zr10	75	59	15	89	95	167	42	220	127
Quality of oil products in first period	Zr11	98	94	90	93	90	95	89	87	83
Safety stock for refinery	Xr13	54	44	20	72	44	35	76	46	20
Management systems in second period	Hr14	83	44	49	90	60	85	43	44	56
Training in second period	Zr15	100	100	100	68	91	96	98	79	98
Material assigned to refinery in second period	Mr16	93	24	165	26	102	10	13	11	12
Gas production in refinery in second period	Zr17	77	63	18	90	97	180	47	233	130
Quality of oil products in second period	Zr18	100	96	92	97	93	97	92	89	91
Net profit margin	Yr19	4.4	6.11	8.76	6.77	5.9	9.35	1	1	9.4

2.2.5. Modeling the Dynamic Network-DEA Structure. The activities in the third level are perspectives of the BSC model. Based on Chen et al. [30], the efficiency of each activity is measured as follows.

For activities of Process 1,

$$\theta_{11}(j) = \frac{W_{r4} \cdot Z_{r4j} + W_{r2} \cdot A_{r2j} + W_{r3} \cdot B_{r3j}}{W_{r1} \cdot Z_{r1j}}$$

$$\theta_{12}(j) = \left(W_{r6} \cdot C_{r6j} + W_{r7} \cdot D_{r7j} + W_{r8} \cdot E_{r8j} + W_{r9} \cdot G_{r9j} + W_{r10} \cdot Z_{r10j}\right) \cdot \left(W_{r4} \cdot Z_{r4j} + W_{r5} \cdot F_{r5j}\right)^{-1}$$

$$\theta_{13}(j) = \frac{W_{r11} \cdot Z_{r11j}}{W_{r10} \cdot Z_{r10j}}$$

$$\theta_{14}(j) = \frac{W_{r5} \cdot F_{r5j} + W_{r13} \cdot X_{r13j}}{W_{r11} \cdot Z_{r11j} + W_{r9} \cdot G_{r9j}}. \quad (9)$$

For activities of Process 2,

$$\theta_{21}(j) = \frac{W_{r15} \cdot Z_{r15j}}{W_{r13} \cdot X_{r13j}}$$

$$\theta_{22}(j) = \frac{W_{r17} \cdot Z_{r17j} + W_{r16} \cdot M_{r16j}}{W_{r15} \cdot Z_{r15j} + W_{r14} \cdot H_{r14j} + W_{r6} \cdot C_{r6j}}$$

$$\theta_{23}(j) = \frac{W_{r18} \cdot Z_{r18j}}{W_{r17} \cdot Z_{r17j} + W_{r2} \cdot A_{r2j} + W_{r7} \cdot D_{r7j}}$$

$$\theta_{24}(j) = \left(W_{r14} \cdot H_{r14j} + W_{r19j} \cdot Y_{r19j}\right) \cdot \left(W_{r8} \cdot E_{r8j} + W_{r18} \cdot Z_{r18j} + W_{r16} \cdot M_{r16j} + W_{r3} \cdot B_{r3j}\right)^{-1}. \quad (10)$$

Each period is a process in the second level. The efficiency of each process is measured by the following.

For Process 1,

$$\theta_{t1}(j) = \left(W_{r13} \cdot X_{r13j} + W_{r2} \cdot A_{r2j} + W_{r3} \cdot B_{r3j} + W_{r6} \cdot C_{r6j} + W_{r7} \cdot D_{r7j} + W_{r8} \cdot E_{r8j}\right) \cdot \left(W_{r1} \cdot X_{r1j}\right)^{-1}. \quad (11)$$

For Process 2,

$$\theta_{t2}(j) = \left(W_{r19} \cdot Y_{r19j}\right) \cdot \left(W_{r13} \cdot X_{r13j} + W_{r2} \cdot A_{r2j} + W_{r3} \cdot B_{r3j} + W_{r6} \cdot C_{r6j} + W_{r7} \cdot D_{r7j} + W_{r8} \cdot E_{r8j}\right)^{-1}. \quad (12)$$

For the first level, the overall efficiency of system by considering the system as black box is measured by

$$\theta_T(j) = \frac{W_{r19} \cdot Y_{r19j}}{W_{r1} \cdot X_{r1j}}. \quad (13)$$

The black-box approach in the DEA was originally developed to measure the efficiency of a DMU as a whole unit by ignoring internal relationships. The calculated efficiency with this approach may not be true because the operations of the component processes are ignored [34]. It may overestimate the efficiency of ignoring the intermediate input/output measures [45]. The black-box objective function is the simplest approach that comes to mind for a dynamic network in the Figure 5. But because it ignores the operations of the component processes and activities in second and third level, it is not an appropriate approach. Hence, for considering the network and dynamic operations of the component in a sublevel, the additive model based on Chen et al. [30] is used.

The weighted average of the efficiencies of the individual components in the activity level treated as perspectives of the dynamic BSC is measured by

$$\beta_p = \frac{(\text{component } p \text{ input})}{(\text{total input accross all components})}. \quad (14)$$

For example the weight of first activity in first period is modeled by

$$\beta_{11} = \left(W_{r1} \cdot X_{r1j}\right) \cdot \left(W_{r1} \cdot X_{r1j} + W_{r4} Z_{r4j} + W_{r5} \cdot F_{r5j} + W_{r10} \cdot Z_{r10j} + W_{r11} \cdot Z_{r11j} + W_{r9} \cdot G_{r9j}\right)^{-1}. \quad (15)$$

Similarly, for the other activities the weights are calculated. The above coefficients are the weights of perspectives of the BSC in the third level named activities. The weights of each period in the second level, named processes, are measured as follows.

For Process 1,

$$\beta_{t1} = \left(W_{r1} \cdot X_{r1j}\right) \cdot \left(W_{r1} \cdot X_{r1j} + W_{r13} \cdot X_{r13j} + W_{r2} \cdot A_{r2j} + W_{r3} \cdot B_{r3j} + W_{r6} \cdot C_{r6j} + W_{r7} \cdot D_{r7j} + W_{r8} \cdot E_{r8j}\right)^{-1}. \quad (16)$$

For Process 2,

$$\beta_{t2} = \left(W_{r13} \cdot X_{r13j} + W_{r2} \cdot A_{r2j} + W_{r3} \cdot B_{r3j} + W_{r6} \cdot C_{r6j} + W_{r7} \cdot D_{r7j} + W_{r8} \cdot E_{r8j}\right) \cdot \left(W_{r1} \cdot X_{r1j} + W_{r13} \cdot X_{r13j} + W_{r2} \cdot A_{r2j} + W_{r3} \cdot B_{r3j} + W_{r6} \cdot C_{r6j} + W_{r7} \cdot D_{r7j} + W_{r8} \cdot E_{r8j}\right)^{-1}. \quad (17)$$

In the studies carried out so far in the network DEA model, the weighted average of processes is considered as objective function. In the structure of this paper in Figure 5, the third level with network relationships is added. Then, the classic weighted average of processes as objective function is not appropriate and needs to be defined as a new objective function. Considering a network relationship in the third level between activities, the weighted average of activities in each process in the third level measure is considered for measuring the weighted average of the process in the second level as follows:

$$\text{Max } \theta: \ \beta_{t1} \cdot (\beta_{11} \cdot \theta_{11} + \beta_{12} \cdot \theta_{12} + \beta_{13} \cdot \theta_{13} + \beta_{14} \cdot \theta_{14}) + \beta_{t2} \cdot (\beta_{21} \cdot \theta_{21} + \beta_{22} \cdot \theta_{22} + \beta_{23} \cdot \theta_{23} + \beta_{24} \cdot \theta_{24}). \quad (18)$$

In (18), the processes efficiency is a convex combination of activities efficiency and the system efficiency is a convex combination of the processes efficiency because

$$\beta_{11} + \beta_{12} + \beta_{13} + \beta_{14} = 1$$
$$\beta_{21} + \beta_{22} + \beta_{23} + \beta_{24} = 1 \quad (19)$$
$$\beta_{t1} + \beta_{t2} = 1.$$

By adding constrains to the objective function of (18) based on Chen et al. [30], the model is completed.

Constrains of the third level are

$$\begin{aligned} &\theta_{11} \le 1; \quad \theta_{12} \le 1; \\ &\theta_{13} \le 1; \quad \theta_{14} \le 1; \\ &\theta_{21} \le 1; \quad \theta_{22} \le 1; \\ &\theta_{23} \le 1; \quad \theta_{24} \le 1. \end{aligned} \tag{20}$$

Constrains of the second level are

$$\theta_{t1} \le 1; \quad \theta_{t2} \le 1. \tag{21}$$

Constrain of the first level is

$$\begin{aligned} &\theta_T \le 1 \\ &W_{r1}, W_{r2}, W_{r3}, W_{r4}, W_{r5}, W_{r6}, W_{r7}, W_{r8}, W_{r9}, W_{r10}, W_{r11}, \\ &W_{r13}, W_{r14}, W_{r15}, W_{r16}, W_{r17}, W_{r18}, W_{r19} \ge 0 \\ &j = 1, \ldots, n. \end{aligned} \tag{22}$$

2.2.6. Measuring the Efficiency. To measure the efficiency based on the proposed model in the previous section, information of the KPIs of the NIORDC in Table 8 is used. The efficiency of nine refineries in three levels and the objective function of the model are illustrated in Table 9. By this calculation, the efficiency of each perspective and efficiency of each period and overall efficiency along with strategic goals and during the time are measured.

3. Result and Discussion

The integrated BSC and DEA model proposed in this paper to measure the efficiency during the time and along with strategic goals have been the main purpose. To follow this purpose, it should be understood how a time factor can be added to the cause-and-effect relation between perspectives of the BSC model. No study using the BSC and DEA models simultaneously has considered the time factor resulting from the lag KPIs as dynamic variables. In addition, no study has considered dynamics with attention to delay of the lag KPIs to draw the cause-and-effect relations of the BSC model reviewed so far. By deploying a proposed framework of this paper, the cause-and-effect relations of perspectives during the time with attention to delay of the lag KPIs between different periods and the lead KPIs in the same period have been created.

As shown in Figure 4, in addition to the classic relations from the "Learning and Growth" perspective to the "Financial" perspective, there are a recursive dynamic relation from the "Financial" perspective to "Internal processes" perspective and a circle loop relation from the "Internal processes" perspective to the "Financial" perspective in a period. There also are dynamic relationships as the dash lines

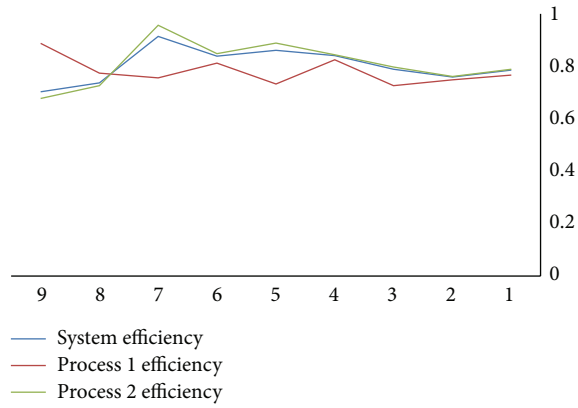


Figure 6: Efficiency score of nine refineries.

from perspectives of the first period to perspectives of the second period. Then, considering the time delay of the lag KPIs in different periods, the dynamic BSC and time factor are considered.

By considering a network relation of the dynamic BSC, the perspectives of each period and different periods have internal and external dynamic network relations, respectively. None of the network DEA structure reviewed so far does not consider the network relations in the third level and the relation of processes stages only is modeled. In the proposed network DEA structure, these relationships are considered.

By deploying the proposed framework, the efficiencies of each perspective and each period as well as the overall efficiency along with strategic goals and during the time considering delay of the lag KPIs are measured. As shown in Figure 6, the system efficiency is a convex combination of processes. It means that, for each DMU, the overall efficiency (θ) is between the maximum and minimum of each period efficiency (θ_{t1}, θ_{t2}). The processes efficiency also is a convex combination of the activities efficiency. It also means that for each DMU, each period efficiency is between the maximum and minimum of perspectives efficiency. These convex relations indicate accuracy of the proposed formulation.

By considering Table 10, the 7th refinery has the best efficiency because of the good efficiency score in the "Financial" and "Internal processes" perspectives in the second period. The ninth refinery has the worth efficiency because of the bad efficiency score in the "customer" perspective in the second period. Similarly, for other refineries, the causes of good or bad efficiency scores are analyzed. Then, the root causes of these scores are cleared with attention to causal relations of perspectives. For example, considering the dynamic BSC as shown in Figure 3, the "Internal processes" perspective in the first period has causal relationships with the "Customer" perspective in the second period. Hence, the root cause of the bad efficiency score of 9th refinery is the bad performance of "Internal process." After identifying inefficient units, by considering the cause-and-effect relations of the dynamic BSC, the root cause of the bad efficiency scores is identified. Then, the strategic initiatives are revised to improve efficiency

Table 10: Calculated efficiency of nine refineries of NIORDC in three levels.

	System	Processes		Activity of period 1				Activity of period 2			
	θ	θ_{t1}	θ_{t2}	θ_{13}	θ_{14}	θ_{13}	θ_{14}	θ_{21}	θ_{22}	θ_{23}	θ_{24}
DMU1	0.786	0.766	0.789	0.624	1	0.553	0.727	0.7	0.762	0.705	0.879
DMU2	0.759	0.748	0.761	0.561	0.998	0.574	0.731	0.699	0.89	0.719	0.717
DMU3	0.789	0.726	0.797	1	0.628	0.565	0.78	0.683	0.976	0.721	0.76
DMU4	0.841	0.825	0.844	0.728	0.892	0.436	1	0.597	0.766	0.756	1
DMU5	0.861	0.733	0.888	0.586	1	0.414	0.756	1	1	0.625	1
DMU6	0.839	0.812	0.848	0.504	0.924	0.523	1	0.483	0.703	1	1
DMU7	0.914	0.756	0.956	0.688	1	0.496	0.72	0.642	1	0.902	1
DMU8	0.737	0.774	0.726	0.29	0.924	0.462	1	0.505	1	0.918	0.414
DMU9	0.703	0.887	0.677	0.609	0.968	1	0.94	0.621	0.897	0.331	0.703

scores. This also is a good method for making a decision on which improvement projects should be started or revised.

4. Conclusion

In this paper, the time factor has been considered with severance of the lead and lag KPIs for drawing the casual relationships during the time with the fuzzy DEMATEL model. By considering the network relations of the dynamic BSC, a new two-stage DEA structure with three levels has been proposed. For formulation of this structure, the weighted average of activities efficiency (i.e., efficiency of perspectives) in the third level has produced weights of the processes efficiency (i.e., weights of periods) as objective function. Then, dynamic and nondynamic variables based on the lag and lead KPIs have been considered, respectively, for the efficiency measurement. In this new objective function, the dynamic network relations of the third level among perspectives have been considered for measuring the efficiency, and the weakness of ignoring these relations have been removed. Based on the proposed framework, the efficiency of nine subsidiary refineries of the NIORDC during the time and along with strategic goals has been measured. By reporting the relationship of the dynamic BSC and these efficiency scores to the NIORDC managers, the results of the proposed framework have been confirmed and validated in experimental space. It can be interesting to use time series based methods (e.g., REF-II model) to predict the efficiency score of DMUs with the data set of the last KPIs for future research.

Conflict of Interests

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

Acknowledgments

The authors would like to thank anonymous reviewers for their valuable comments. Additionally, the authors thank the National Iranian Oil Refining & Distribution Company for providing the real data.

References

[1] R. S. Kaplan and D. P. Norton, "Using the balanced scorecard as a strategic management system," *Harvard Business Review*, vol. 74, no. 2, pp. 70–82, 1996.

[2] V. Nissen, "Modeling corporate strategy with the fuzzy balanced scorecard," in *Proceedings of the Symposium on Fuzzy Systems in Computer Science (FSCS '06)*, E. Hüllermeier, R. Kruse, A. Nürnberger, and J. Strackeljan, Eds., pp. 121–138, Magdeburg, Germany, 2006.

[3] R. S. Kaplan and D. P. Norton, "The strategy map: guide to aligning intangible assets," *Strategy & Leadership*, vol. 32, no. 5, pp. 10–17, 2004.

[4] M. L. Frigo, P. G. Pustorino, and G. W. Krull, "The balanced scorecard for community banks: translating strategy into action," *Bank Accounting and Finance*, vol. 13, no. 3, pp. 17–29, 2000.

[5] P. Rouse, M. Putterill, and D. Ryan, "Integrated performance measurement design: insights from an application in aircraft maintenance," *Management Accounting Research*, vol. 13, no. 2, pp. 229–248, 2002.

[6] H. Eilat, B. Golany, and A. Shtub, "R&D project evaluation: an integrated DEA and balanced scorecard approach," *Omega*, vol. 36, no. 5, pp. 895–912, 2008.

[7] H. Eilat, B. Golany, and A. Shtub, "Constructing and evaluating balanced portfolios of R&D projects with interactions: a DEA based methodology," *European Journal of Operational Research*, vol. 172, no. 3, pp. 1018–1039, 2006.

[8] T.-Y. Chen, C.-B. Chen, and S.-Y. Peng, "Firm operation performance analysis using data envelopment analysis and balanced scorecard: a case study of a credit cooperative bank," *International Journal of Productivity and Performance Management*, vol. 57, no. 7, pp. 523–539, 2008.

[9] T. García-Valderrama, E. Mulero-Mendigorri, and D. Revuelta-Bordoy, "Relating the perspectives of the balanced scorecard for R&D by means of DEA," *European Journal of Operational Research*, vol. 196, no. 3, pp. 1177–1189, 2009.

[10] C. Y. Chiang and B. Lin, "An integration of balanced scorecards and data envelopment analysis for firm's benchmarking management," *Total Quality Management and Business Excellence*, vol. 20, no. 11, pp. 1153–1172, 2009.

[11] M. A. Maced, A. C. Barbosa, and G. T. Cavalcante, "Performance of bank branches in Brazil: applying data envelopment analysis (DEA) to indicators related to the BSC perspectives," *E&G—Revista Economia e Gestão*, vol. 19, no. 19, pp. 65–84, 2009.

[12] A. Asosheh, S. Nalchigar, and M. Jamporazmey, "Information technology project evaluation: an integrated data envelopment analysis and balanced scorecard approach," *Expert Systems with Applications*, vol. 37, no. 8, pp. 5931–5938, 2010.

[13] C. A. F. Amado, S. P. Santos, and P. M. Marques, "Integrating the data envelopment analysis and the balanced scorecard approaches for enhanced performance assessment," *Omega*, vol. 40, no. 3, pp. 390–403, 2012.

[14] S. M. Seyedhosseini, A. E. Taleghani, A. Bakhsha, and S. Partovi, "Extracting leanness criteria by employing the concept of balanced scorecard," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10454–10461, 2011.

[15] E. Falatoonitoosi, Z. Leman, and S. Sorooshian, "Casual strategy mapping using integrated BSC and MCDM DEMATEL," *Journal of American Science*, vol. 8, no. 1, pp. 125–155, 2012.

[16] J.-S. Horng, C.-H. Liu, S.-F. Chou, and C.-Y. Tsai, "Creativity as a critical criterion for future restaurant space design: developing a novel model with DEMATEL application," *International Journal of Hospitality Management*, vol. 33, no. 1, pp. 96–105, 2013.

[17] M. N. Shaik and W. Abdul-Kader, "Comprehensive performance measurement and causal-effect decision making model for reverse logistics enterprise," *Computers and Industrial Engineering*, vol. 68, no. 1, pp. 87–103, 2014.

[18] J. Jassbi, F. Mohamadnejad, and H. Nasrollahzadeh, "A fuzzy DEMATEL framework for modeling cause and effect relationships of strategy map," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5967–5973, 2011.

[19] S. A. Heydariyeh, M. Javidnia, and A. Mehdiabadi, "A new approach to analyze strategy map using an integrated BSC and FUZZY DEMATEL," *Management Science Letters*, vol. 2, no. 1, pp. 161–170, 2012.

[20] C. Chen and H. Yan, "Network DEA model for supply chain performance evaluation," *European Journal of Operational Research*, vol. 213, no. 1, pp. 147–155, 2011.

[21] M. Alvandi, S. Fazli, L. Yazdani, and M. Aghaee, "An integrated MCDM method in ranking BSC perspectives and key performance indicators (KPIs)," *Management Science Letters*, vol. 2, no. 3, pp. 995–1004, 2012.

[22] D. J.-F. Jeng and G.-H. Tzeng, "Social influence on the use of clinical decision support systems: revisiting the unified theory of acceptance and use of technology by the fuzzy DEMATEL technique," *Computers and Industrial Engineering*, vol. 62, no. 3, pp. 819–828, 2012.

[23] J. Ren, A. Manzardo, S. Toniolo, and A. Scipioni, "Sustainability of hydrogen supply chain. Part I: identification of critical criteria and cause-effect analysis for enhancing the sustainability using DEMATEL," *International Journal of Hydrogen Energy*, vol. 38, no. 33, pp. 14159–14171, 2013.

[24] J. S. Horng, C. H. Liu, S. F. Chou, and C. Y. Tsai, "Creativity as a critical criterion for future restaurant space design: developing a novel model with DEMATEL application," *International Journal of Hospitality Management*, vol. 33, no. 1, pp. 96–105, 2013.

[25] C. Kao, "Network data envelopment analysis: a review," *European Journal of Operational Research*, vol. 239, no. 1, pp. 1–16, 2014.

[26] L. M. Seiford and J. Zhu, "Profitability and marketability of the top 55 U.S. commercial banks," *Management Science*, vol. 45, no. 9, pp. 1270–1288, 1999.

[27] Z. Yang, "A two-stage DEA model to evaluate the overall performance of Canadian life and health insurance companies," *Mathematical and Computer Modelling*, vol. 43, no. 7-8, pp. 910–919, 2006.

[28] C. Kao and S.-N. Hwang, "Efficiency decomposition in two-stage data envelopment analysis: an application to non-life insurance companies in Taiwan," *European Journal of Operational Research*, vol. 185, no. 1, pp. 418–429, 2008.

[29] Y.-C. Chiou, L. W. Lan, and B. T. H. Yen, "A joint measurement of efficiency and effectiveness for non-storable commodities: integrated data envelopment analysis approaches," *European Journal of Operational Research*, vol. 201, no. 2, pp. 477–489, 2010.

[30] Y. Chen, W. D. Cook, and J. Zhu, "Deriving the DEA frontier for two-stage processes," *European Journal of Operational Research*, vol. 202, no. 1, pp. 138–142, 2010.

[31] Y.-M. Wang and K.-S. Chin, "Some alternative DEA models for two-stage process," *Expert Systems with Applications*, vol. 37, no. 12, pp. 8799–8808, 2010.

[32] C. Kao and S.-N. Hwang, "Decomposition of technical and scale efficiencies in two-stage production systems," *European Journal of Operational Research*, vol. 211, no. 3, pp. 515–519, 2011.

[33] H. F. Lewis, S. Mallikarjun, and T. R. Sexton, "Unoriented two-stage DEA: the case of the oscillating intermediate products," *European Journal of Operational Research*, vol. 229, no. 2, pp. 529–539, 2013.

[34] A. Charnes, W. W. Cooper, B. Golany, R. Halek, G. Klopp, and E. Schmitz, "Two-phase data envelopment analysis approaches to policy evaluation and management of army recruiting activities: tradeoffs between joint services and army advertising," Research Report CCS #532, Center for Cybernetic Studies. University of Texas-Austin, Austin, Tex, USA, 1986.

[35] C. Kao and S.-N. Hwang, "Efficiency measurement for network systems: IT impact on firm performance," *Decision Support Systems*, vol. 48, no. 3, pp. 437–446, 2010.

[36] Y. Chen, J. Du, H. D. Sherman, and J. Zhu, "DEA model with shared resources and efficiency decomposition," *European Journal of Operational Research*, vol. 207, no. 1, pp. 339–349, 2010.

[37] L. Liang, Z.-Q. Li, W. D. Cook, and J. Zhu, "Data envelopment analysis efficiency in two-stage networks with feedback," *IIE Transactions*, vol. 43, no. 5, pp. 309–322, 2011.

[38] B. C. Xie, Y. Fan, and Q. Q. Qu, "Does generation form influence environmental efficiency performance? An analysis of China's power system," *Applied Energy*, vol. 96, pp. 261–271, 2012.

[39] S. Lozano, E. Gutiérrez, and P. Moreno, "Network DEA approach to airports performance assessment considering undesirable outputs," *Applied Mathematical Modelling*, vol. 37, no. 4, pp. 1665–1676, 2013.

[40] J. Nouri, F. H. Lotfi, H. Borgheipour, F. Atabi, S. M. Sadeghzadeh, and Z. Moghaddas, "An analysis of the implementation of energy efficiency measures in the vegetable oil industry of Iran: a data envelopment analysis approach," *Journal of Cleaner Production*, vol. 52, pp. 84–93, 2013.

[41] C. Kao and S.-T. Liu, "Multi-period efficiency measurement in data envelopment analysis: the case of Taiwanese commercial banks," *Omega*, vol. 47, pp. 90–98, 2014.

[42] C. Kao, "Efficiency measurement for parallel production systems," *European Journal of Operational Research*, vol. 196, no. 3, pp. 1107–1112, 2009.

[43] G. Bi, C. Feng, J. Ding, and M. R. Khan, "Estimating relative efficiency of DMU: Pareto principle and Monte Carlo oriented DEA approach," *Information Systems and Operational Research*, vol. 50, no. 1, pp. 44–57, 2012.

[44] N. Adler, V. Liebert, and E. Yazhemsky, "Benchmarking airports from a managerial perspective," *Omega*, vol. 41, no. 2, pp. 442–458, 2013.

[45] C.-H. Wang, Y.-H. Lu, C.-W. Huang, and J.-Y. Lee, "R&D, productivity, and market value: an empirical study from high-technology firms," *Omega*, vol. 41, no. 1, pp. 143–155, 2013.

[46] T.-Y. Lin and S.-H. Chiu, "Using independent component analysis and network DEA to improve bank performance evaluation," *Economic Modelling*, vol. 32, no. 1, pp. 608–616, 2013.

[47] G. Klepac, "Risk evaluation in the insurance company using REFII model," *Intelligent techniques in recommendation systems: contextual advancements and new methods*, pp. 84–104, 2012.

[48] C. L. Lin and W. W. Wu, "A fuzzy extension of the DEMATEL method for group decision making," *European Journal of Operational Research*, vol. 156, Article ID 445455, pp. 445–455, 2004.

[49] W.-S. Lee, A. Y. Huang, Y.-Y. Chang, and C.-M. Cheng, "Analysis of decision making factors for equity investment by DEMATEL and analytic network process," *Expert Systems with Applications*, vol. 38, no. 7, pp. 8375–8383, 2011.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 351354, 13 pages
http://dx.doi.org/10.1155/2015/351354

Research Article

Short-Term Wind Speed Hybrid Forecasting Model Based on Bias Correcting Study and Its Application

Mingfei Niu, Shaolong Sun, Jie Wu, and Yuanlei Zhang

School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, China

Correspondence should be addressed to Shaolong Sun; sunshl13@lzu.edu.cn

Received 27 October 2014; Revised 7 December 2014; Accepted 8 December 2014

Academic Editor: Pandian Vasant

The accuracy of wind speed forecasting is becoming increasingly important to improve and optimize renewable wind power generation. In particular, reliable short-term wind speed forecasting can enable model predictive control of wind turbines and real-time optimization of wind farm operation. However, due to the strong stochastic nature and dynamic uncertainty of wind speed, the forecasting of wind speed data using different patterns is difficult. This paper proposes a novel combination bias correcting forecasting method, which includes the combination forecasting method and forecasting bias correcting model. The forecasting result shows that the combination bias correcting forecasting method can more accurately forecast the trend of wind speed and has a good robustness.

1. Introduction

Because of the global energy shortage, renewable energy has received increasing attention, just now. Wind power is one of the cleanest renewable energy sources that produces no greenhouse gases, has no effect on climate change, and produces little environmental impacts, and the energy generated from the wind has been well recognized as environmentally friendly, socially beneficial, and economically competitive for many applications [1]. As of now the effectiveness of wind speed forecasting is an important role in the scheduling of wind power. At present, these methods can be divided into two categories: statistical models and machine-learning models. Statistical models primarily use a time series approach and have been successfully applied for forecasting [2–7]. These models are based on the assumption that a linear correlation structure exists among time series values. Therefore, nonlinear patterns cannot be captured using these models. To overcome this limitation, machine-learning models have been used to improve nonlinear time series predictions (which primarily include artificial neural networks, support vector machines, heuristic algorithm, and fuzzy logic methods) [8–34].

In a nutshell, in the past decades, many computational intelligence techniques have been developed for short-term wind speed forecasting, for instance, support vector regression [15, 27, 35], support vector machine [26, 31, 33, 36], fuzzy model [21, 27], artificial neural networks [8, 9, 11, 13, 14, 20, 23–25, 28, 29], wavelet method [9, 34, 37], and heuristic intelligence algorithm: particle swarm optimization [37, 38], adaptive particle swarm optimization [17, 36, 39], chaotic particle swarm optimization [31], biogeography-based optimization [10], coral reefs optimization [16], gravitational search algorithm [19], and harmony search algorithm [16]. In the next section, we will detail the explanation of the previous work in the short-term wind speed prediction.

The remaining sections are arranged as follows. The related work will be brief description in Section 2. The preparation methods and main modeling process are described from Section 3 to Section 6. Section 7 forecasts the wind speed of Penglai using three wind farms and provides the forecasting results and analyses. Finally, the conclusion is presented in Section 8 and the future research in Section 9.

2. Related Work

In the above references, Song et al. [5] employ a discrete-state Markov chain to model the nonlinear characteristic of the wind speed time series, and a Bayesian inference is applied

to evaluate the parameters of the Markov-switching model. Finally, by comparison with other methods, this proposed method outperforms them. Liu et al. [9] present four important decomposing algorithms including wavelet decomposition, wavelet packet decomposition, empirical mode decomposition, and fast ensemble empirical mode decomposition, which are all adopted to realize the wind speed high-precision predictions. Salcedo-Sanz et al. [16] introduce a new hybrid coral reefs optimization and harmony search algorithm; this novel approach is utilized to obtain the best set of meteorological variables in the context of short-term wind speed forecasting, and the selection variable will be input to an extreme learning machine network. Experimental result shows that these proposed methods have good results when compared to other approaches. Wang et al. [17] proposed an optimization model to decide the rated power system and the capacity of a compressed air energy storage system in a power system with high wind power penetration. Moreno et al. [18] proposed a strategy including the uncertainty of involving market and wind power. Mondal et al. [19] solved economic dispatch problem in wind generation. In a nutshell, as the randomness of wind speed distribution, every prediction model owns some limitations.

In the short-term wind speed forecasting, because of ignoring of the secondary influence factors and correlations, every prediction model can generate prediction errors, which are the difference between the predicted value and the actual value, the main causes that forecasting method just considers the main factors, and many of the secondary factors are ignored. However, as the effect of the secondary factors, the forecasting bias may form a certain trend. Making allowance for these minor influence factors, the bias correction becomes important. The basic idea of forecasting bias correction is following. After forecasting by the prediction model and comparing with the actual wind speed, forecasting error is generated. Using suitable prediction model to forecast error, error correction can be got, which is used to modify the original forecasting result. The error correction prediction model expression is as follows: $Y = Y_c + \widehat{e}_c$. Y is final forecasting value, Y_c is combination forecasting value, and $\widehat{e}_c$ is bias correction.

Nowadays, there are some error correction models [36, 40–42], such as the periodic extrapolation, vector error correction model, partial simulation approximate value, and Bayesian error correction model. But the relevant researches about the short-term wind speed and wind power error correction models are very rare. So this paper quotes bias correction model in short-term wind speed forecasting, thus, making wind power scheduling reasonable.

Due to the randomness of wind speed, the former forecasting models have their own limitations. It is because of the volatility of the wind speed; firstly, this paper proposes a combination of a linear model and two nonlinear models: double exponential smoothing method (*DES*), backpropagation of particle swarm optimization artificial neural network (*PSO-BPANN*), and Elman artificial neural network (*Elman-ANN*). The inputs of *PSO-BPANN* and *Elman-ANN* consist of historical wind speed data and residual errors of the *DES* model. Then the combined weight will use adaptive particle swarm optimization algorithm (*APSO*) to optimize. The combination model can more accurately forecast short-time wind speed. The reasons of wind speed forecasting errors are analyzed; then, the empirical orthogonal function model will be error correction. Making use of this model, main variables can be extracted, and error correction model is built by the empirical orthogonal function regression method. Some advantages are that the main variables are determined by the properties of wind speed series itself, but not prior artificial regulation, and can reflect the actual wind speed data basic structure, and expansion equation converges fast. Finally, combination bias correcting forecasting model is presented. In order to check the validity of the model, the case study will be analyzed in detail.

Generally speaking, in this research, our main contribution is that we set up the combination bias correction forecasting model in the short-term wind speed forecasting, which consists of double exponential smoothing (*DES*), *PSO-BP* artificial neural network, and *Elman* artificial neural network and, finally, adaptive particle swarm optimization algorithm (*APSO*) to optimize the combination weights. Forecasting error can be corrected by the empirical orthogonal function, which can be used in variables analysis and regression forecasting for wind speed prediction bias and correcting wind speed prediction result. In this paper, the ten-minute wind speed data from three wind farms in Penglai of Shandong province in China were used as examples to evaluate the performance of the proposed approach. To avoid volatility due to the *PSO-BP*, Elman, and *PSO* optimization algorithm, all of the simulations were repeated 50 times prior to averaging. As time goes on, more wind speed information will be obtained, more accurate wind speed characteristic will be derived by forecasting models, and the new information on the wind speed is absorbed by this combination bias correction model. Therefore, the performance of this combination bias correction model will be accurate and stable.

3. Combination Forecasting Model

Due to the random features of wind speed, the nonlinear characteristics are significant. So the combination forecasting consists of a linear model and two nonlinear models; then the combined weight will use adaptive particle swarm optimization algorithm (*APSO*) to optimize. Three models theories are as follows.

3.1. Wind Speed DES Model. Exponential smoothing technique used to forecast wind speed is comparatively simple. It only needs a single wind speed series. It can be divided into single exponential smoothing method (*SES*), double exponential smoothing method (*DES*), cubic exponential smoothing method (*CES*), and so on. The main steps of exponential smoothing method used to forecast wind speed include modeling and calculating the consequence of exponential smoothing, determining the smoothing coefficient of α, and forecasting. Because *DES* is based on single exponential smoothing values and is more accurate than the single exponential smoothing method, it can better reflect the

linear characteristic of wind speed. So this paper uses *DES*. The procedure is as follows.

3.1.1. Modeling and Calculating the Consequence of Exponential Smoothing. The *DES* is modeled by the wind speed data, and the expression is as follows:

$$\hat{x}_{l+T} = \hat{a}_T + \hat{b}_T l. \tag{1}$$

The *DES* model is similar to the double moving average method (*DMA*); first of all, on the base of the sequence of the single exponential smoothing (smoothing coefficient $0 < \alpha < 1$),

$$S_t^{(1)} = \alpha x_t + (1-\alpha) S_{t-1}^{(1)}, \quad t = 1, 2, \ldots, T, \tag{2}$$

and calculating the consequence of second exponential smoothing

$$S_t^{(2)} = \alpha S_t^{(1)} + (1-\alpha) S_{t-1}^{(2)}, \quad t = 1, 2, \ldots, T, \tag{3}$$

where the initial value $S_0^{(2)} = S_0^{(1)}$ and $S_0^{(1)} = x_1$, $S_1^{(1)} = \alpha x_1 + (1-\alpha) S_0^{(1)} = x_1$, $S_0^{(2)} = S_1^{(1)} = x_1$.

3.1.2. Determining the Smoothing Coefficient α and Forecasting. A sequence of the single exponential smoothing $S_t^{(1)}$ is relative to data x_t migration or lag effect, and a sequence of the double exponential smoothing $S_t^{(2)}$ is relative to $S_t^{(1)}$ lag effect, otherwise. Under certain conditions, such as t large enough, especially α is close to 1, and two lags are equal. Among them, the smoothing coefficient α can be optimized via analyzing the wind speed forecasting error. Just for the sake of smooth sequence, α can be small enough. If it is used to predict, when the original wind speed volatility is not obvious, α can get smaller or get bigger, in order to make the smoothing sequence reflect the changes of the wind speed data.

Through calculation we get the intercept and slope of the prediction linear

$$\begin{aligned} \hat{a} &= 2S_t^{(1)} - S_t^{(2)} \\ \hat{b} &= \frac{\alpha}{1-\alpha}\left[S_t^{(1)} - S_t^{(2)}\right], \quad t = 1, 2, \ldots, T. \end{aligned} \tag{4}$$

Finally, prediction model is as follows:

$$\begin{aligned} \overline{x}_{t+1} &= \hat{a}_t + \hat{b}_t = \frac{2-\alpha}{1-\alpha} S_t^{(1)} - \frac{1}{1-\alpha} S_t^{(2)}, \quad t = 1, 2, \ldots, T-1 \\ \hat{x}_{T+l} &= \hat{a}_T + \hat{b}_T l, \quad l = 1, 2, \ldots. \end{aligned} \tag{5}$$

3.2. Wind Speed PSO-BP Neural Network Prediction Model. In order to solve the nonlinear features of wind speed, artificial neural network (*ANN*) methods have been proposed. *ANN* is able to give better performance in dealing with the nonlinear relationships among their input variables [35]. The conventional backpropagation algorithm (*BP*) is successfully applied to complex nonlinear problems. However, using BP method needs the following; the transfer function of each neuron must be different. Moreover, it has been proven that gradient techniques are slow to train and are sensitive to the initial guess which can possibly be trapped in a local minimum [43].

To overcome these shortcomings, the paper introduces particle swarm optimization algorithm (*PSO*) to optimize the *BP* network to solve the wind speed forecasting problem. The *PSO* algorithm is applied to the neural network in the training phase, to obtain a set of weights that will minimize the error function in competitive time. Weights are progressively updated until the convergence criterion is satisfied. The objective function to be minimized by the *PSO* algorithm is the predicted error function [37].

Figure 1 shows the flow process of *PSO-BPANN* forecasting model [38].

In Figure 1, the normalized formula and fitness function are as follows:

$$y_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \tag{6}$$

where x_i is defined as an initial group of data among the collected wind speed data groups. $x_{\max}$ and $x_{\min}$ express the maximum and minimum data group among the collected data groups, respectively; consider

$$f = \frac{1}{n} \sum_{j=1}^{n} \sum_{k=1}^{m} \left(y_k - t_k\right)^2, \tag{7}$$

where y_k is the actual forecasting output value, t_k is the target value, n is the numbers of training sample, m is the nodes of output.

In short-term wind speed forecasting, the sample data is normalized and smoothly processed. Firstly, the inputs of *PSO-BPANN* are made of historical wind speed and residual errors calculated by *DES* model. The target output is original value of wind speed next ten-minute data.

3.3. Wind Speed Elman Artificial Neural Network Prediction Model. This paper has proposed another neural network: *Elman* artificial neural network. The *Elman* neural network is proposed firstly by Elman in 1990 [44, 45]. It is a form of recurrent neural network (*RNN*) by adding recurrent links into hidden layer as a feedback connection which allows the network to learn to recognize and generate temporal patterns. However, *RNN* has some merits for nonlinear system modeling and forecasting while the order of systems under consideration is unknown or with uncertainty. Generally, comparing with the *BP* neural networks, the training for the Elman recursion neural networks is faster than for the *BP* neural networks.

The researches on *Elman-ANN* have been developed with nonlinear modeling, transfer function, and field of application [46, 47], such as nonlinear stable adaptive control and solar activity forecasting.

The flow chart of *Elman-ANN* prediction model is shown in Figure 2 [48].

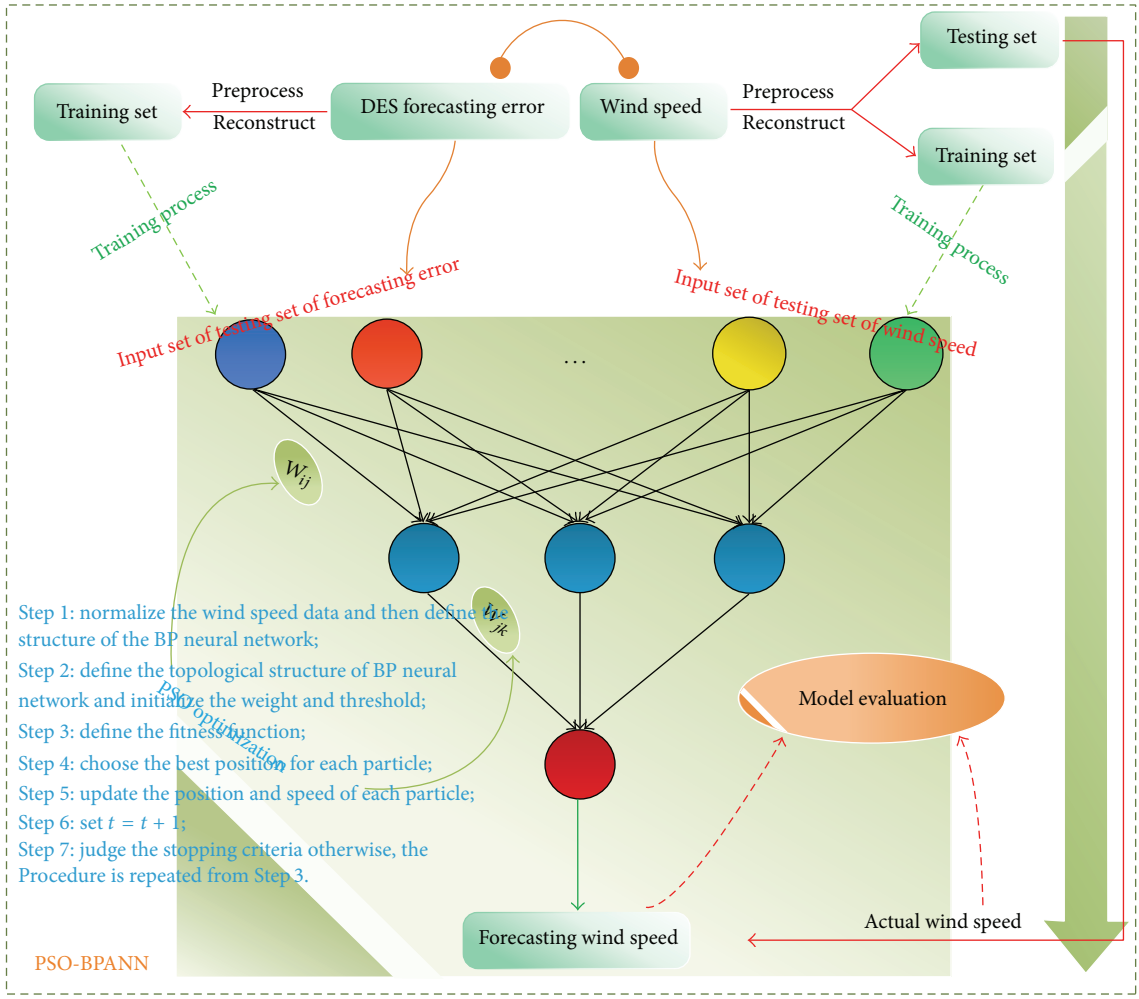


Figure 1: The process of *PSO-BPANN* model.

In Figure 2 the correction formula is as follows:

$$\omega_{ij}^{(l+1)} = \omega_{ij}^{(l)} - \eta \frac{\partial E_k}{\partial \omega_{ij}^{(l)}}, \quad \eta > 0, \tag{8}$$

wherein $\omega_{ij}^{(l)}$ is the weight coefficient between j neuron in l layer and i neuron in $l + 1$ layer; η is gain.

In this paper, we have processed the wind speed data sample according to neighboring three ten-minute time periods and residual errors calculated by *DES* model before the training of the network, aiming to forecast one-step ahead wind speed by using previous three ten-minute data and residual errors.

4. Combination Forecasting Model

DES prediction is y_1; *PSO-BPANN* prediction is y_2; *Elman-ANN* prediction is y_3; weighted average prediction is y_c. Combination forecasting model is as follows:

$$y_c = \omega_1 y_1 + \omega_2 y_2 + \omega_3 y_3, \tag{9}$$

wherein ω_1, ω_2, and ω_3 are weights, respectively. And the following is satisfied:

$$\omega_1 + \omega_2 + \omega_3 = 1. \tag{10}$$

In order to improve the precision of the combination forecasting model, the paper proposed a novel adaptive particle swarm optimization (*APSO*) to optimize the combination weights. As the change of particle fitness function value, inertia weight will be automatically adjusted, which makes the particle search direction illuminating enhancement. The *APSO* algorithm not only converges fast, but also does not fall into local extreme points easily. The specific formula of adaptive adjustment is as follows [49]:

$$\begin{aligned} v_i(t+1) = {} & \omega(t)\, v_i(t) + c_1 r_1 \left[p_i(t) - x_i(t)\right] \\ & + c_2 r_2 \left[p_g(t) - x_i(t)\right], \end{aligned} \tag{11}$$

$$x_i(t+1) = x_i(t) + v_i(t+1), \tag{12}$$

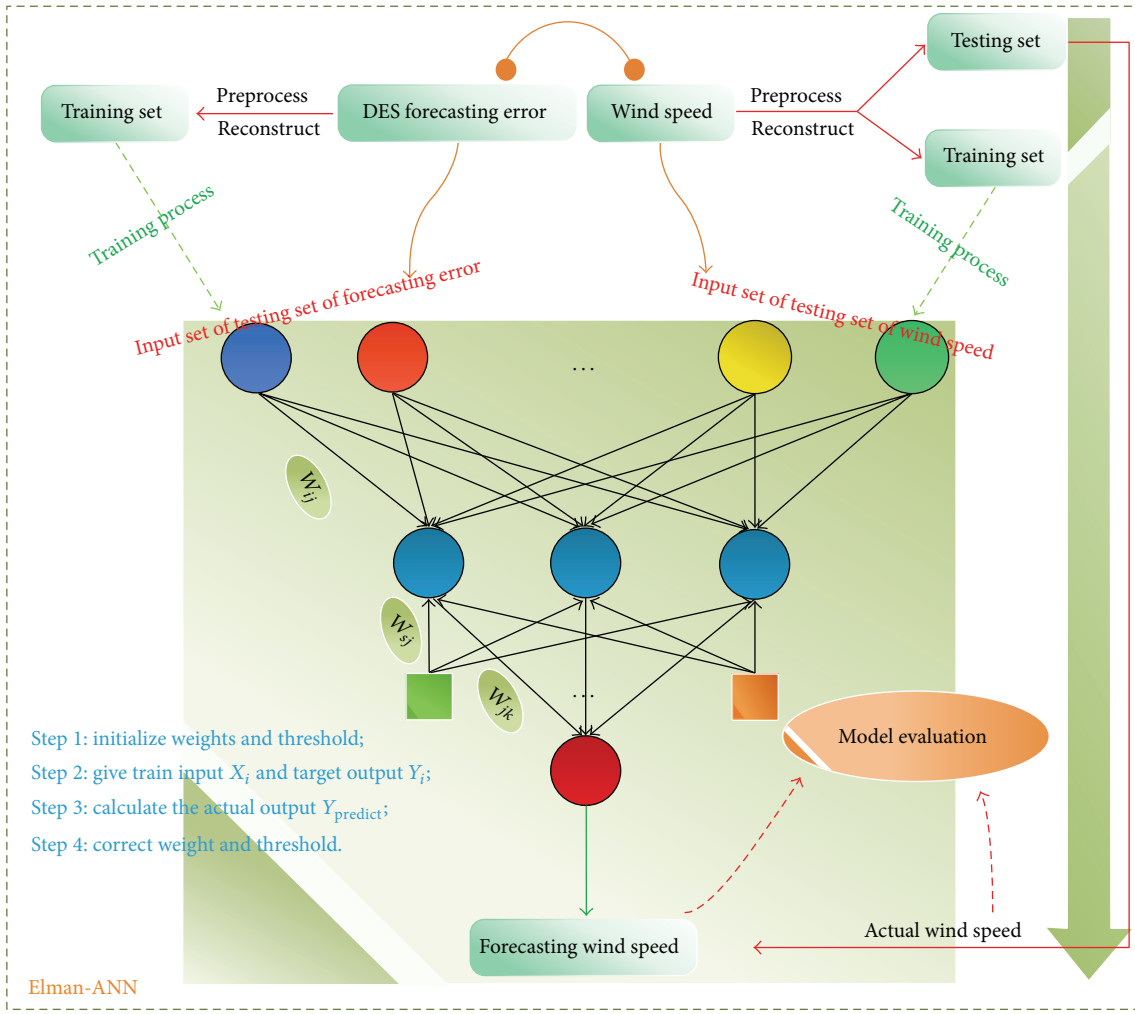


Figure 2: The process of *Elman-ANN* model.

$$\omega(t) = \begin{cases} \lambda \cdot \omega(t-1) \\ \quad + \theta \dfrac{f\left(p_g(t)\right) - f\left(x_i(t)\right)}{f\left(p_g(t)\right) - f\left(x_{\min}(t)\right)} & \text{if } \omega(t) \geq \omega_{\min} \\ \omega_{\min} & \text{otherwise,} \end{cases} \tag{13}$$

where c_1 and c_2 are positive constants, which are called acceleration coefficients. r_1 and r_2 are two random numbers in the range $[0, 1]$. $x_i = (x_i(1), x_i(2), \ldots, x_i(D))$ represents the ith particle. $p_i = (p_i(1), p_i(2), \ldots, p_i(D))$ represents the best previous position (the position giving the best fitness value) of the ith particle. The symbol g represents the index of the best particle among all the particles in the population. $v_i = (v_i(1), v_i(2), \ldots, v_i(D))$ represents the rate of the position change for particle i and D represent the search space dimension. $\omega(t)$ is the adaptive inertia weight, λ and θ are constraint factors in the range $[0, 1]$, $\omega_{\min}$ is the minimum inertia weight, and $f(\bullet)$ is the fitness function. In this paper, the fitness function is defined as follows:

$$f = -\frac{1}{N} \sum_{i=1}^{N} (y_c - \hat{y}_i)^2, \tag{14}$$

where y_c and $\hat{y}_i$ are the output value of combination model and the actual wind speed value; N is the sample size.

The process of *APSO* to optimize the combination weights is in Figure 3 [39].

5. Prediction Error Correction Model-*EOF* Model

Empirical orthogonal function (*EOF*) has recently become popular tool in the atmospheric science decomposition and expansion since its introduction several decades ago [50, 51]. It is equivalent to the principal component used in the multivariate statistics and is close relatives to the bases used in factor analysis. It is characterized by the fast convergence

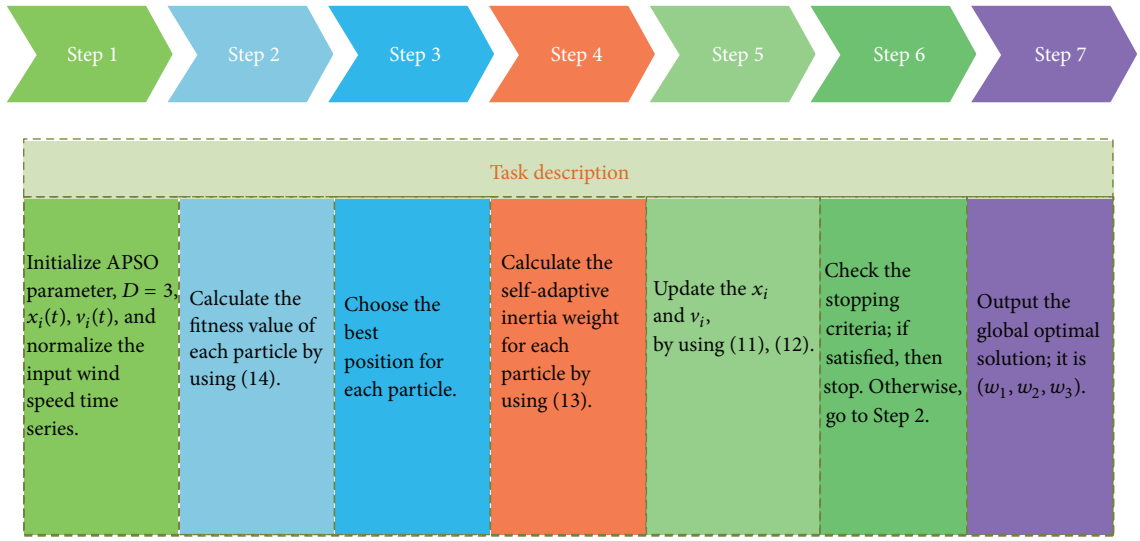


Figure 3: The process of *APSO* optimization combination weights.

of expansion and can approach variable field condition by a few items. It was first proposed by Pearson in 1902 [52], and Lorenz first introduced it to the weather and climate research [53]. At present, *EOF* has been widely used in the ocean and other fields. Probably the most important thing is that this method often enables a description of the variations of a complex geophysical field with a relatively small number of functions and associated time coefficients. This property is especially important in the development of statistical prediction schemes which rely on multiple linear regression, since the skill and statistical confidence of those schemes depend heavily upon a priori methods of reducing the number of available predictors [54, 55].

In this paper, we will describe and illustrate the *EOF* technique, and the forecasting errors have been corrected with *EOF* decomposition and regression.

5.1. EOF Decomposition. *EOF* is also called empirical orthogonal decomposition or natural orthogonal decomposition. It is similar to principal component analysis but different from meanings and analysis methods [48]. The basic idea is to decompose data that contain m variables in n time observation field. That is to say, the wind speed forecasting error e_{ij} ($i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$ that contain m variables in n time observation field of a region) is decomposed into the product of orthogonally the sum of space function s_{ip} and the sum of function t_{pj}. The form is as follows [56]:

$$e_{ij} = \sum_{p=1}^{m} s_{ip} t_{pj} = s_{i1} t_{1j} + s_{i2} t_{2j} + \cdots + s_{im} t_{mj}, \quad (i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, n). \tag{15}$$

It is written in the matrix form:

$$\mathbf{E} = \mathbf{ST}. \tag{16}$$

According to the orthogonality, the decomposition should be satisfied:

$$s_i s_j = \sum_{p=1}^{m} s_{ip} s_{pj} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$
$$t_i t_j = \sum_{p=1}^{m} t_{ip} t_{pj} = \begin{cases} 0 & i \neq j \\ 1 & i = j. \end{cases} \tag{17}$$

Formula (15) is right multiplied $\mathbf{E}^T$:

$$\mathbf{E}\mathbf{E}^T = \mathbf{S}\mathbf{T}\mathbf{T}^T\mathbf{S}^T. \tag{18}$$

Denote

$$\mathbf{A} = \mathbf{E}\mathbf{E}^T. \tag{19}$$

$\mathbf{A}$ is a $m \times m$ symmetric matrix, According to the principle of symmetric matrix decomposition, the following must be satisfied:

$$\mathbf{S}^T\mathbf{A}\mathbf{S} = \mathbf{\Lambda} \quad \text{or} \quad \mathbf{A} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^T, \tag{20}$$

wherein $\mathbf{\Lambda}$ is a diagonal matrix made up of $\mathbf{A}$ eigenvalues. $\mathbf{S}$ is a matrix made up of the feature vector. So

$$\mathbf{T}\mathbf{T}^T = \mathbf{\Lambda}$$
$$\mathbf{S}^T\mathbf{S} = \mathbf{S}\mathbf{S}^T = \mathbf{I}, \tag{21}$$

where $\mathbf{I}$ is a unit matrix. $\mathbf{S}$ can be got from $\mathbf{A}$. The time function can be got by (15) left multiplied $\mathbf{S}^T$:

$$\mathbf{T} = \mathbf{S}^T\mathbf{E} \quad \text{or} \quad t_{ij} = \sum_{p=1}^{m} s_{pi} e_{pj}. \tag{22}$$

From (15)–(21) completed the EOF decomposition as like the equation (15).

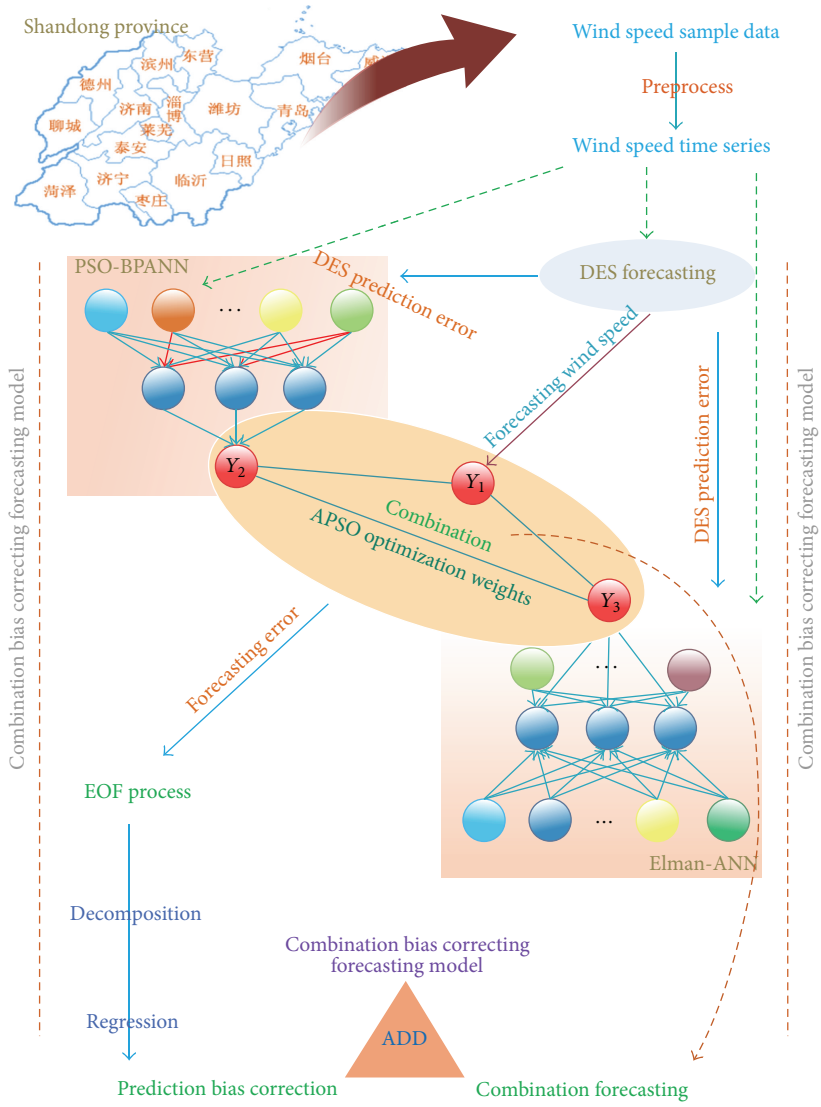


Figure 4: The process of combination bias correcting forecasting model.

5.2. EOF Regression. The main idea of *EOF* regression is that, decomposed into orthogonal principal component, *EOF* model completes regression with a small amount of principal components and changes into orthogonal variables. There is no large error on regression equation [48].

With the *EOF* decomposition of **E**, according to the precision requirement, the former p variance components can be extracted $X_1, X_2, \ldots, X_p$. The regression equation is as follows:

$$e_{ij} = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p. \tag{23}$$

In the regression equation (23), the selection of p is very important. The characteristics of λ_i ($i = 1, 2, \ldots, n$, n is the numbers of eigenvalue) are calculated by the wind speed forecasting error covariance matrix $\mathbf{EE}^T$; let the former p eigenvalue be in descending order, and calculation formula is as follows:

$$\varepsilon^2 = \frac{\sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{m} \lambda_i}. \tag{24}$$

According to the precision requirement, p can be selected, usually $\varepsilon^2 \geq 80\%$.

6. Combination Bias Correcting Forecasting Model

Combination bias correcting forecasting model consists of the combination forecasting model and the forecasting error correcting model. Figure 4 shows the flow chart of modeling.

Table 1: Models evaluation of simulation.

Wind farm	Evaluation criteria	Simulation				
		DES	PSO-BPNN	Elman	Hybrid	Hybrid bias correcting
1	MAPE (%)	6.45	3.07	2.96	2.01	1.26
	MSE	0.511	0.056	0.040	0.056	0.019
2	MAPE (%)	7.12	3.14	2.30	2.05	0.97
	MSE	0.575	0.062	0.059	0.051	0.027
5	MAPE (%)	7.50	3.15	3.29	2.14	1.14
	MSE	0.512	0.054	0.051	0.042	0.021

7. Case Study

7.1. Collection of Data. The data of three farms in Penglai of China are collected to examine the combination bias correcting model. In particular, firstly, the data of wind farm 1 are used to witness the whole process of the proposed method. In the same way, the corresponding forecasting results of wind farm 2 and wind farm 5 are shown and further verify the validity of the method, respectively.

In this paper, the 4500 observations of ten-minute wind speed are used for training, and the remaining 6 observations are used to test the effectiveness of the models. It is used to forecast next one hour; the wind speed times series sampled each hour consisting of 6 data points.

7.2. Evaluation Criteria of Forecast Performance. In order to evaluate the forecast effectiveness of the model, two indices for error forecast serve as the criteria to evaluate the forecasting performance. They are mean square error (*MSE*) and mean absolute percent error (*MAPE*). The values of the indices are smaller and the forecast performance is better. The indices are as follows [57]:

$$\begin{aligned} \mathrm{MSE} &= \frac{1}{n}\sum_{t=1}^{n}\left|Y_t - \widehat{Y}_t\right| \\ \mathrm{MAPE} &= \frac{1}{n}\sum_{t=1}^{n}\frac{\left|Y_t - \widehat{Y}_t\right|}{Y_i}, \end{aligned} \tag{25}$$

wherein Y_t is the value of actual wind speed for a time period t and $\widehat{Y}_t$ is the forecast value for the same period.

7.3. Combination Bias Correcting Forecasting Model. To verify the proposed combination bias correction method, firstly, we conducted a simulation experiment with the different wind farm. In this experiment, we compared our model with the *DES, PSO-BPNN, Elman,* and combined model, which were established using the original data without error correction. The comparisons of wind speed values simulate using the *DES*, the *PSO-BPNN, Elman*, and the combined model with the actual models being shown in Figure 5. Specific process is as follows.

DES prediction is y_1; *PSO-BPANN prediction is* y_2; *Elman-ANN* prediction is y_3; weighted average prediction is y_c. Prediction errors are e_1, e_2, e_3, and e_c. Combination forecasting model is $y_c = \omega_1 y_1 + \omega_2 y_2 + \omega_3 y_3$, where *APSO* optimization weights are ω_1, ω_2, and ω_3, and $\omega_1 + \omega_2 + \omega_3 = 1$.

Finally, combination bias correcting forecasting is $\widehat{y}_c$, *EOF* decomposition and regression are $\widehat{e}_c$, and $\widehat{y}_c = y_c + \widehat{e}_c$.

To avoid randomness due to the *PSO-BP*, *Elman*, and *PSO* optimization algorithm, all of the simulations were repeated 50 times prior to averaging. For more instinctive to compare, Figure 5 shows the *DES* model, *PSO-BPANN* model, *Elman-ANN*, combination forecasting model, and combination bias correcting simulation forecasting graphics at the same time. By the error graphics it can be seen that the combination bias correcting forecasting method is better than combination forecasting method, and combination method is better than single forecasting method.

In order to accurately describe the simulation effect of this proposed model. From Table 1, it can be seen that the combination bias correcting model performs much better than combination forecasting model and single forecasting model. More precisely, in wind farm 1, comparing with DES, PSO-BPANN, Elman-ANN, and combination model, MAPE and MSE of the proposed model are reduced to 1.26% and 0.019. However, in order to reflect the robustness of this model, wind farm 2 and wind farm 5 are also analyzed; comparing with *DES*, *PSO-BPANN*, *Elman-ANN*, and combination model, *MAPE* and *MSE* of the proposed model are reduced to 0.97% and 0.027, 1.41% and 0.021, respectively. Therefore, using this model to forecast is very feasible.

7.4. The Analysis of Forecasting Results. Based on the historical data of three wind farms' simulation training, the short-term wind speed forecasting is reasonable. To avoid volatility due to the *PSO-BP*, Elman, and *PSO* optimization algorithm, all of the forecasting was repeated 50 times prior to averaging.

Figure 6 shows the short-term wind speed forecasting graphics of three wind farms and the covariance matrix table of the *EOF* decomposition. In the covariance matrix table, P_1, P_2, P_3, and P_4 are components. The variance contribution rate is defined $\varphi = \lambda_i / \sum_{j=1}^{m} \lambda_j$, and the cumulative variance contribution rate is defined $\varepsilon^2 = \sum_{i=1}^{p} \lambda_i / \sum_{i=1}^{m} \lambda_i$, and the number of eigenvalues is m.

According to the precision requirement, p can be selected, usually $\varepsilon^2 \geq 80\%$; in wind farm 1, the rate of λ_1 and λ_2 is 99.22%, and P_3 is ignored; then $p = 2$. With this similarity, the rate of λ_1 is 81.57% and 93.57%, respectively; then $p = 1$ of wind farm 2 and wind farm 5.

Form Table 2 and Figure 6 it is clear that the combination bias correcting model performs much better than

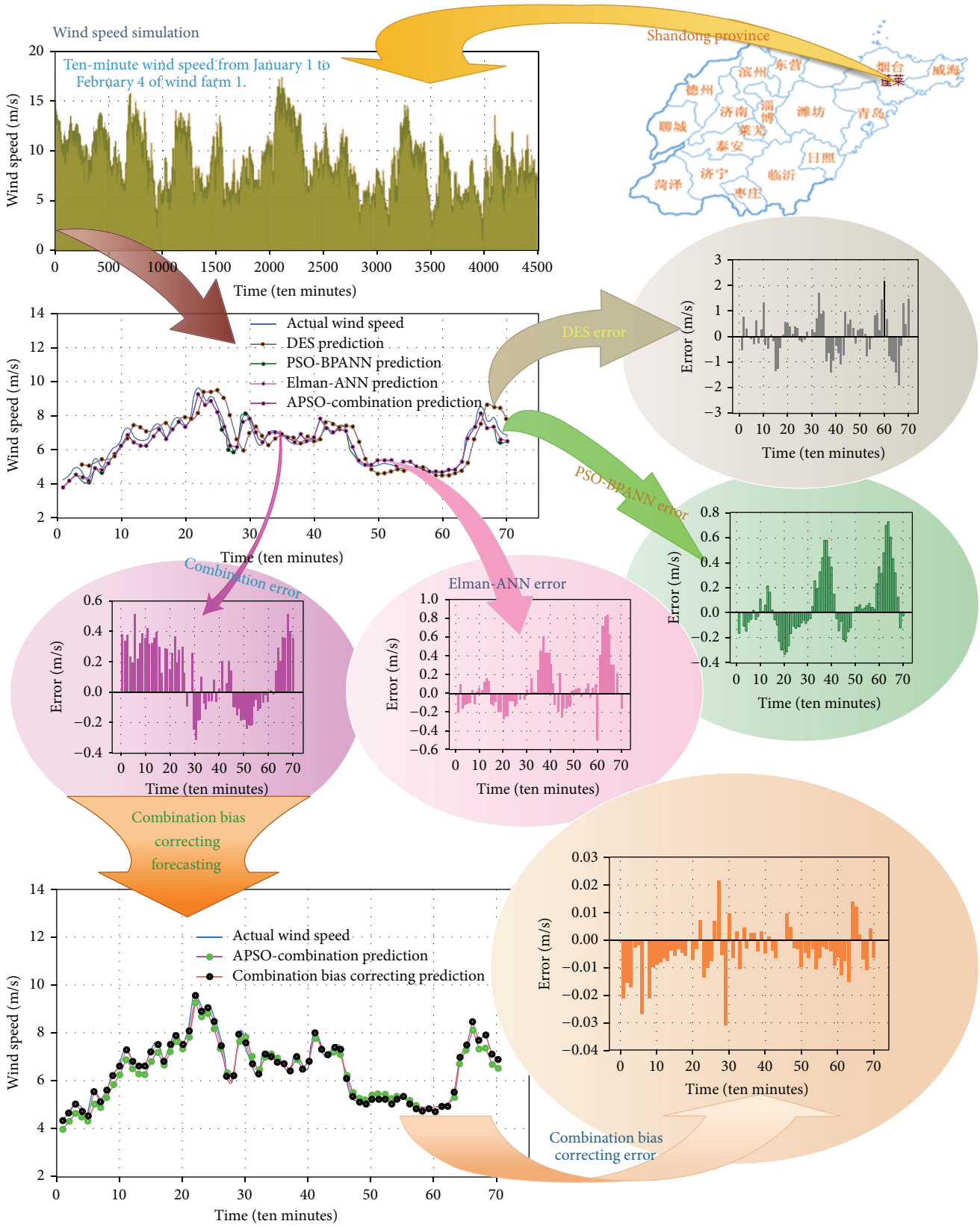
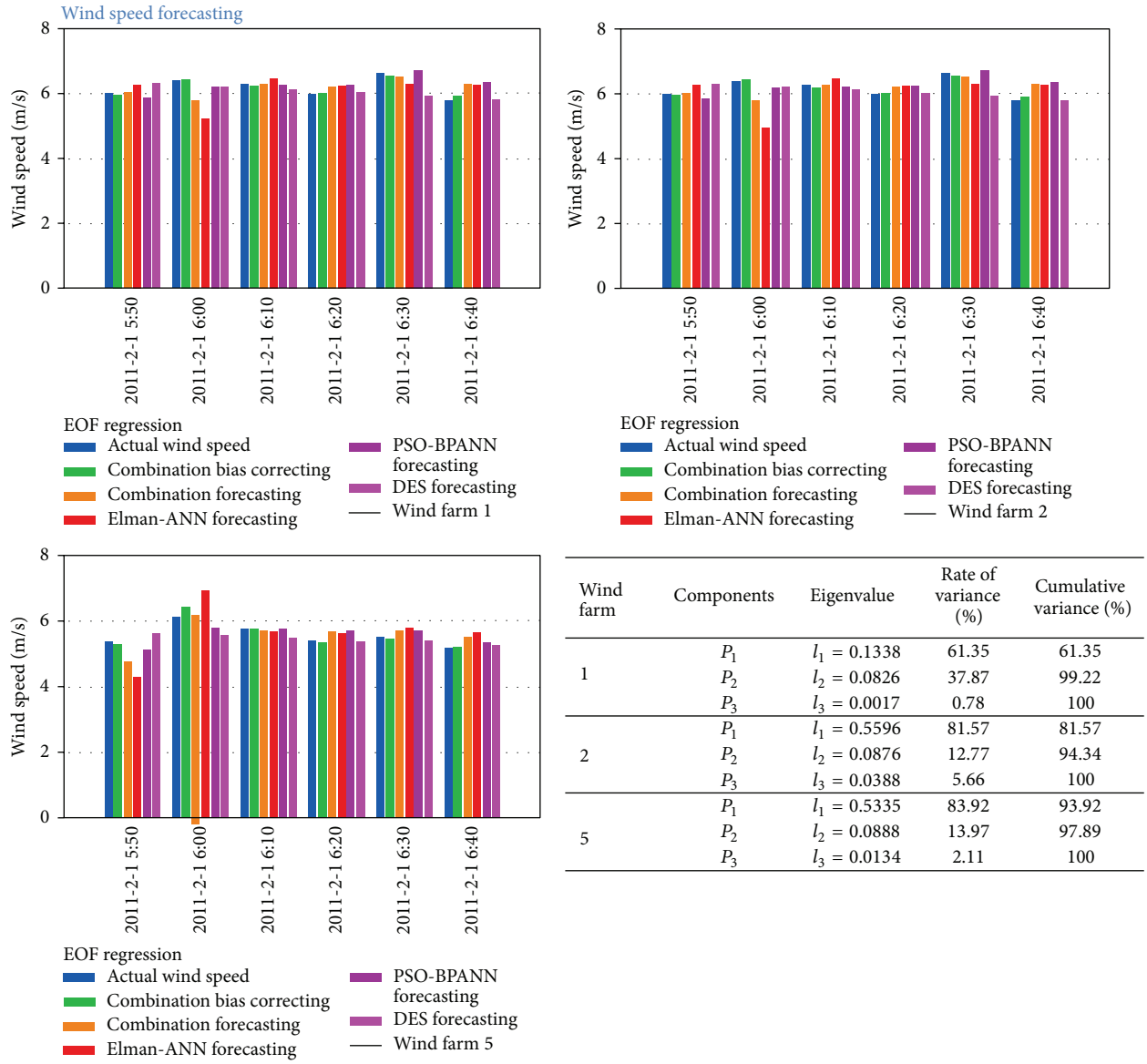


Figure 5: Simulation chart of wind farm 1.



Wind farm	Components	Eigenvalue	Rate of variance (%)	Cumulative variance (%)
1	P_1	$l_1 = 0.1338$	61.35	61.35
	P_2	$l_2 = 0.0826$	37.87	99.22
	P_3	$l_3 = 0.0017$	0.78	100
2	P_1	$l_1 = 0.5596$	81.57	81.57
	P_2	$l_2 = 0.0876$	12.77	94.34
	P_3	$l_3 = 0.0388$	5.66	100
5	P_1	$l_1 = 0.5335$	83.92	93.92
	P_2	$l_2 = 0.0888$	13.97	97.89
	P_3	$l_3 = 0.0134$	2.11	100

Figure 6: Short-term wind speed forecasting.

Table 2: Models evaluation of forecasting.

Wind farm	Evaluation criteria	Forecasting				
		DES	PSO-BPNN	Elman	Hybrid	Hybrid bias correcting
1	MAPE (%)	6.61	3.72	3.08	2.75	1.27
	MSE	0.111	0.050	0.056	0.043	0.137
2	MAPE (%)	3.89	3.58	7.67	3.84	2.12
	MSE	0.107	0.072	0.429	0.112	0.043
5	MAPE (%)	3.98	4.52	9.35	5.81	1.19
	MSE	0.079	0.065	0.402	0.139	0.035

combination forecasting model and single forecasting model. More precisely, in wind farm 1, comparing with *DES*, *PSO-BPANN*, *Elman-ANN*, and combination model, *MAPE* and *MSE* of the proposed model are reduced to 1.27% and 0.137. By comparing the *MAPE* and *MSE* of the combination bias correction forecasting model they are also the smallest. But it can be shown that this combination bias correction model can be just right for wind farm 1. In order to eliminate the randomness, this paper also forecast wind farm 2 and wind farm 5.

Similarly, in order to reflect the steadiness of this combination bias correction forecasting model, we analyze the forecasting results of wind farm 2 and wind farm 5 in detail. For wind farm 2, comparing with *DES*, *PSO-BPANN*, *Elman-ANN*, the combination model, and the proposed model, *MAPE* is 3.89%, 3.58%, 7.67%, 3.84%, and 2.12%, respectively, and *MSE* is 0.107, 0.072, 0.429, 0.112, and 0.043, respectively. In a similar way, comparing with *DES*, *PSO-BPANN*, *Elman-ANN*, the combination model and this proposed model for wind farm 5, *MAPE* is 3.98%, 4.52%, 9.35%, 5.81%, and 1.19%, respectively, and *MSE* is 0.079, 0.065, 0.402, 0.139, and 0.035, respectively.

Through the above analysis, in detail, we used this test to evaluate the predictive performances of the five models. In the three wind farms, it is shown that the combination bias correcting forecasting method is better than combination forecasting method, and combination method is better than single forecasting method; the *MAPE* and *MSE* of the optimized hybrid bias correction model are also the smallest. Hence, all of the indices imply that the optimized hybrid bias correction model can effectively decrease the error of the forecasting values compared to the other four forecasting methods.

In a word, by the detailed analysis of three wind farms, the combination bias correcting forecasting method can more accurately forecast the trend of wind speed; *EOF* decomposition and regression are an effective bias correction tool that may be combined with other forecasting methods, such as statistical and other artificial intelligence models, and have a good performance, which can be applied to the nonstationary wind speed forecasting.

8. Conclusion

Due to the importance of the wind speed forecasting error, this paper proposes a novel combination bias correcting forecasting method, which includes the combination forecasting method and forecasting bias correcting model. The method can improve the precision of forecasting. The main conclusions are as follows. (1) The combination forecasting model consists of *DES*, *PSO-BP* artificial neural network model, and *Elman* artificial neural network model for short-term wind speed forecasting which are proposed and *APSO* to optimize the combination weights. Forecasting results are more satisfactory than the single forecasting model. (2) *EOF* forecasting bias correction method can be used in variables analysis and regression forecasting for wind speed prediction bias and correcting wind speed prediction result. (3) By the detailed analysis of the forecasting results of three wind farms, the combination bias correcting forecasting method can more accurately forecast the trend of wind speed and has a good robustness.

9. Future Research

In the short-term wind speed prediction, because of the wind speed nonlinear and chaotic characteristics we will be preprocessing the original data. Using wavelet transform, empirical mode decomposition, or singular spectrum analysis, removes white noise of the original wind speed series, and the filtered series will be modeling. However, the parameters of this preprocessing method can be optimized by the novel heuristic intelligence algorithm; for example, the parameters (windows length L and component r) of the singular spectrum analysis are optimized by the gravitational search algorithm, firefly algorithm or shuffled leapfrog algorithm, and so on. In addition, this hybrid method also can be applied in other fields, such as power load forecasting, stock index forecasting, air transport demand forecasting, accidental deaths series forecasting, and exchange rates forecasting in the future.

Conflict of Interests

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

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant no. 71171102). The authors are grateful to the editors and reviewers for their valuable suggestions.

References

[1] M. Monfared, H. Rastegar, and H. M. Kojabadi, "A new strategy for wind speed forecasting using artificial intelligent methods," *Renewable Energy*, vol. 34, no. 3, pp. 845–848, 2009.

[2] G. D'Amico, F. Petroni, and F. Prattico, "Wind speed and energy forecasting at different time scales: a nonparametric approach," *Physica A*, vol. 406, pp. 59–66, 2014.

[3] H. Liu, H.-Q. Tian, and Y.-F. Li, "Comparison of two new *ARIMA-ANN* and *ARIMA-Kalman* hybrid methods for wind speed prediction," *Applied Energy*, vol. 98, pp. 415–424, 2012.

[4] A. Tascikaraoglu and M. Uzunoglu, "A review of combined approaches for prediction of short-term wind speed and power," *Renewable and Sustainable Energy Reviews*, vol. 34, pp. 243–254, 2014.

[5] Z. Song, Y. Jiang, and Z. Zhang, "Short-term wind speed forecasting with Markov-switching model," *Applied Energy*, vol. 130, pp. 103–112, 2014.

[6] M. Zhou, Z. Yan, Y. X. Ni et al., "A novel ARIMA approach on electricity price forecasting with the improvement of predicted error," *Proceeding of the CSEE*, vol. 24, no. 12, pp. 63–68, 2004.

[7] J. Aghaei, T. Niknam, R. Azizipanah-Abarghooee, and J. M. Arroyo, "Scenario-based dynamic economic emission dispatch considering load and wind power uncertainties," *International Journal of Electrical Power and Energy Systems*, vol. 47, no. 1, pp. 351–367, 2013.

[8] Y. Jiang, Z. Song, and A. Kusiak, "Very short-term wind speed forecasting with Bayesian structural break model," *Renewable Energy*, vol. 50, pp. 637–647, 2013.

[9] H. Liu, H.-Q. Tian, and Y.-F. Li, "Comparison of new hybrid FEEMD-MLP, FEEMD-ANFIS, Wavelet Packet-MLP and Wavelet Packet-ANFIS for wind speed predictions," *Energy Conversion and Management*, vol. 89, pp. 1–11, 2015.

[10] R. A. Gupta, R. Kumar, and A. K. Bansal, "*BBO*-based small autonomous hybrid power system optimization incorporating

wind speed and solar radiation forecasting," *Renewable and Sustainable Energy Reviews*, vol. 41, pp. 1366–1375, 2015.

[11] J.-Z. Wang, J.-J. Wang, Z.-G. Zhang, and S.-P. Guo, "Forecasting stock indices with back propagation neural network," *Expert Systems with Applications*, vol. 38, no. 11, pp. 14346–14355, 2011.

[12] H. Liu, H.-Q. Tian, C. Chen, and Y.-F. Li, "A hybrid statistical method to predict wind speed and wind power," *Renewable Energy*, vol. 35, no. 8, pp. 1857–1861, 2010.

[13] J. Hu, J. Wang, and G. Zeng, "A hybrid forecasting approach applied to wind speed time series," *Renewable Energy*, vol. 60, pp. 185–194, 2013.

[14] J.-J. Wang, J.-Z. Wang, Z.-G. Zhang, and S.-P. Guo, "Stock index forecasting based on a hybrid model," *Omega*, vol. 40, no. 6, pp. 758–766, 2012.

[15] J. Wang, W. Zhu, W. Zhang, and D. Sun, "A trend fixed on firstly and seasonal adjustment model combined with the ε-SVR for short-term forecasting of electricity demand," *Energy Policy*, vol. 37, no. 11, pp. 4901–4909, 2009.

[16] S. Salcedo-Sanz, A. Pastor-Sánchez, J. Del Ser, L. Prieto, and Z. W. Geem, "A coral reefs optimization algorithm with harmony search operatorsfor accurate wind speed prediction," *Renewable Energy*, vol. 75, pp. 93–101, 2015.

[17] J. Wang, S. Zhu, W. Zhang, and H. Lu, "Combined modeling for electric load forecasting with adaptive particle swarm optimization," *Energy*, vol. 35, no. 4, pp. 1671–1678, 2010.

[18] M. Á. Moreno, M. Bueno, and J. Usaola, "Evaluating risk-constrained bidding strategies in adjustment spot markets for wind power producers," *International Journal of Electrical Power and Energy Systems*, vol. 43, no. 1, pp. 703–711, 2012.

[19] S. Mondal, A. Bhattacharya, and S. H. Nee Dey, "Multi-objective economic emission load dispatch solution using gravitational search algorithm and considering wind power penetration," *International Journal of Electrical Power and Energy Systems*, vol. 44, no. 1, pp. 282–292, 2013.

[20] K. W. Chau, "Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River," *Journal of Hydrology*, vol. 329, no. 3-4, pp. 363–367, 2006.

[21] C. T. Cheng, C. P. Ou, and K. W. Chau, "Combining a fuzzy optimal model with a genetic algorithm to solve multi-objective rainfall-runoff model calibration," *Journal of Hydrology*, vol. 268, no. 1–4, pp. 72–86, 2002.

[22] P. Flores, A. Tapia, and G. Tapia, "Application of a control algorithm for wind speed prediction and active power generation," *Renewable Energy*, vol. 30, no. 4, pp. 523–536, 2005.

[23] Z. X. Guo, W. K. Wong, and M. Li, "Sparsely connected neural network-based time series forecasting," *Information Sciences*, vol. 193, pp. 54–71, 2012.

[24] C. Hamzaçebi, "Improving artificial neural networks' performance in seasonal time series forecasting," *Information Sciences*, vol. 178, no. 23, pp. 4550–4559, 2008.

[25] S. A. Kalogirou, "Artificial neural networks in renewable energy systems applications: a review," *Renewable and Sustainable Energy Reviews*, vol. 5, no. 4, pp. 373–401, 2000.

[26] J.-Y. Lin, C.-T. Cheng, and K.-W. Chau, "Using support vector machines for long-term discharge prediction," *Hydrological Sciences Journal*, vol. 51, no. 4, pp. 599–612, 2006.

[27] K.-P. Lin, P.-F. Pai, Y.-M. Lu, and P.-T. Chang, "Revenue forecasting using a least-squares support vector regression model in a fuzzy environment," *Information Sciences*, vol. 220, pp. 196–209, 2013.

[28] M. Carolin Mabel and E. Fernandez, "Analysis of wind power generation and prediction using *ANN*: a case study," *Renewable Energy*, vol. 33, no. 5, pp. 986–992, 2008.

[29] N. Muttil and K.-W. Chau, "Neural network and genetic programming for modelling coastal algal blooms," *International Journal of Environment and Pollution*, vol. 28, no. 3-4, pp. 223–238, 2006.

[30] A. Sfetsos, "A comparison of various forecasting techniques applied to mean hourly wind speed time series," *Renewable Energy*, vol. 21, no. 1, pp. 23–35, 2000.

[31] Q. Wu, R. Law, E. Wu, and J. X. Lin, "A hybrid-forecasting model reducing Gaussian noise based on the Gaussian support vector regression machine and chaotic particle swarm optimization," *Information Sciences*, vol. 238, pp. 96–110, 2013.

[32] J.-X. Xie, C.-T. Cheng, K.-W. Chau, and Y.-Z. Pei, "A hybrid adaptive time-delay neural network model for multi-step-ahead prediction of sunspot activity," *International Journal of Environment and Pollution*, vol. 28, no. 3-4, pp. 364–381, 2006.

[33] J. Y. Zhou, J. Shi, and G. Li, "Fine tuning support vector machines for short-term wind speed forecasting," *Energy Conversion and Management*, vol. 52, no. 4, pp. 1990–1998, 2011.

[34] L.-J. Wang, L. Dong, X.-Z. Liao, and Y. Gao, "Short-term power prediction of a wind farm based on wavelet analysis," *Proceedings of the Chinese Society of Electrical Engineering*, vol. 29, no. 28, pp. 30–33, 2009.

[35] K. Chen and J. Yu, "Short-term wind speed prediction using an unscented Kalman filter based state-space support vector regression approach," *Applied Energy*, vol. 113, pp. 690–705, 2014.

[36] Y.-S. Huang, J.-J. Deng, and Z.-Z. Yuan, "*SVM* short-term load forecasting based on *ARMA* error calibration and the adaptive particle swarm optimization," *Power System Protection and Control*, vol. 39, no. 14, pp. 26–32, 2011.

[37] Z. A. Bashir and M. E. El-Hawary, "Applying wavelets to short-term load forecasting using PSO-based neural networks," *IEEE Transactions on Power Systems*, vol. 24, no. 1, pp. 20–27, 2009.

[38] D. Xian and X. Bingji, "Telecommunication traffic forecasting based on BP neural network trained by PSO," *Journal of Central South University*, vol. 42, no. 1, pp. 24–25, 2011.

[39] C. Dong, G. Wang, Z. Chen, and Z. Yu, "A method of self-adaptive inertia weight for PSO," in *Proceedings of the International Conference on Computer Science and Software Engineering (CSSE '08)*, vol. 1, pp. 1195–1198, IEEE, Hubei, China, December 2008.

[40] C. Xiu, T. Wang, M. Tian, Y. Li, and Y. Cheng, "Short-term prediction method of wind speed series basedon fractal interpolation," *Chaos, Solitons & Fractals*, vol. 68, pp. 89–97, 2014.

[41] K. G. Sheela and S. N. Deepa, "Neural network based hybrid computing model for wind speed prediction," *Neurocomputing*, vol. 122, pp. 425–429, 2013.

[42] M. Xu, Y. Qiao, and Z. Lu, "A comprehensive error evaluation method for short-term wind power prediction," *Automation of Electric Power Systems*, vol. 35, no. 12, pp. 20–26, 2011.

[43] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[44] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[45] H.-T. He and X. Tian, "An improved Elman network and its application in flatness prediction modeling," in *Proceedings*

of the 2nd International Conference on Innovative Computing, Information and Control (ICICIC '07), vol. 9, p. 552, September 2007.

[46] S. Marra and F. C. Morabito, "A new technique for solar activity forecasting using recurrent Elman networks," *World Academy of Science, Engineering and Technology*, vol. 7, no. 8, pp. 68–73, 2005.

[47] X. Li, Z. Chen, and Z. Yuan, "Nonlinear stable adaptive control based upon Elman networks," *Applied Mathematics—A Journal of Chinese Universities*, vol. 15, no. 3, pp. 332–340, 2000.

[48] X. Nan, Q. Li, D. Qiu, Y. Zhao, and X. Guo, "Short-term wind speed syntheses correcting forecasting model and its application," *International Journal of Electrical Power and Energy Systems*, vol. 49, no. 1, pp. 264–268, 2013.

[49] J. Wang, S. Zhu, W. Zhang, and H. Lu, "Combined modeling for electric load forecasting with adaptive particle swarm optimization," *Energy*, vol. 35, no. 4, pp. 1671–1678, 2010.

[50] A. M. Obukhov, "Statistically homogeneous fields on a sphere," *Uspekhi Matematicheskikh Nauk*, vol. 2, pp. 196–198, 1947.

[51] J. E. Kutzbach, "Empirical eigenvectors of sea-level pressure, surface temperature and precipitation complexes over North America," *Journal of Applied Meteorology*, vol. 6, no. 5, pp. 791–802, 1967.

[52] K. Pearson, "On lines and plans of closest fit to system of points in space philosophical," *Magazine*, vol. 6, pp. 559–572, 1902.

[53] E. N. Lorenz, "Empirical orthogonal functions and statistical weather prediction," Statistical Forecasting Project 1, MIT, Cambridge, Mass, USA, 1956.

[54] R. E. Davis, "Predictability of sea surface temperature and sea level pressure anomalies over the North Pacific Ocean," *Journal of Physical Oceanography*, vol. 6, no. 3, pp. 249–266, 1976.

[55] T. P. Barnett and K. Hasselmann, "Techniques of linear prediction, with application to oceanic and atmospheric fields in the tropical Pacific," *Reviews of Geophysics & Space Physics*, vol. 17, no. 5, pp. 949–968, 1979.

[56] B. Wang and H. Yue, "EOF model analysis of place names landscape in Guangdong province on GIS," *Scientia Geographica Sinica*, vol. 27, no. 2, pp. 282–288, 2007.

[57] Z. Zhao, J. Wang, J. Zhao, and Z. Su, "Using a Grey model optimized by Differential Evolution algorithm to forecast the per capita annual net income of rural households in China," *Omega*, vol. 40, no. 5, pp. 525–532, 2012.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 320130, 7 pages
http://dx.doi.org/10.1155/2015/320130

Research Article

Parameters Optimization and Application to Glutamate Fermentation Model Using SVM

Xiangsheng Zhang and Feng Pan

Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University, Wuxi 214122, China

Correspondence should be addressed to Xiangsheng Zhang; zxs@jiangnan.edu.cn and Feng Pan; pan_feng_63@163.com

Received 10 September 2014; Revised 23 December 2014; Accepted 23 December 2014

Academic Editor: Pandian Vasant

Aimed at the parameters optimization in support vector machine (SVM) for glutamate fermentation modelling, a new method is developed. It optimizes the SVM parameters via an improved particle swarm optimization (IPSO) algorithm which has better global searching ability. The algorithm includes detecting and handling the local convergence and exhibits strong ability to avoid being trapped in local minima. The material step of the method was shown. Simulation experiments demonstrate the effectiveness of the proposed algorithm.

1. Introduction

Glutamate fermentation is a complex microbial growth process. Glutamate bacterium draws raw material of nutrition and produces complex biochemical reactions in vivo specific enzyme [1]. The reaction process is highly nonlinear, time-variant, and uncertain. It is very difficult to establish dynamic model in the application of fermentation control [2, 3].

Modeling the glutamate fermentation process under normal or abnormal conditions is major challenge as an optimization strategy is applied from the initial stage to the final stage. It is difficult to measure online substrate, biomass, and product concentrations; hence pH, dissolved oxygen (DO) concentration, and CO_2 production are usually utilized in the bioprocess analysis. The process variables provide indications of the bioprocess condition. However, the information density of these data is usually low, and the multidimensional nature of the data usually makes it difficult to understand.

Support vector machines (SVMs), developed by Boster, Guyon, and Vapnik (1992), are a kind of machine learning method based on statistical learning theory [4–7]. It has become a hot research in the field of machine learning. SVM overcomes commendably such defects as dimensionality curse and overfitting that are apt to appear in some other conventional algorithms, such as neural networks [8, 9].

But there exists a problem in the practical application of SVM. This problem is how to select some SVM parameters so that the performance of SVM can be maximized. These SVM parameters mainly include the penalty constant C, the relaxation factor ξ, and the parameters in kernel function (e.g., the width in the RBF kernel function), and they affect the SVM performance more or less. Most existing approaches use leave-one-out (LOO) [10], gradient descent (GD) [11], genetic algorithm (GA) [12], particle swarm optimization (PSO) [13], and related parameter estimators. There does not exist a single universally effective method to select these SVM parameters. Generally, cross verification trial is used, but it involves *ad hoc* user interference factors or requires that the kernel function should be continuously differentiable, and the resulting SVM classifiers are prone to falling into local minima.

In order to overcome the shortcomings of the existing parameter selection approaches mentioned above, an attempt is made to jointly optimize the feature selection and the SVM parameters with some evolutionary algorithms such as improved particle swarm optimization (IPSO) algorithm, in hopes of improving the SVM performance in glutamate fermentation. This paper describes an IPSO algorithm which has better global search ability and then focuses on optimizing the parameters which can be used in the prediction of SVM modeling and state of glutamate fermentation.

The paper is organized as follows. In Section 2, the SVM with mixed kernel function is briefly reviewed. Section 3 describes the standard PSO algorithm. Then in Section 4 we present the new IPSO technique. The simulations results are provided in Section 4. Finally, concluding remarks are drawn in Section 5.

2. Mixed Kernel Function

The aim of this section is to introduce the notations and to review the concepts that are relevant to the development of the proposed model parameter selection method.

The development of SVM starts from the simplest case of two classes that is linearly separable. Its basic mathematical model can be given using the following training sample set:

$$T = \{x_i, y_i,\ i = 1, 2, \ldots, l\}, \quad x_i \in R^N,\ y_i \in R,$$
$$f(x) = \sum_{i=1}^{l} w_i \phi_i(x) + b, \tag{1}$$

where $\{\phi_i(x)\}_{i=1}^{l}$ is the data in features space and $\{w_i(x)\}_{i=1}^{l}$ and b are coefficients. They can be estimated by minimizing the regularized risk function

$$R(C) = C\frac{1}{l}\sum_{i=1}^{l} L_\varepsilon(y, f(x_i)) + \frac{1}{2}\|w\|^2, \tag{2}$$

where $L_\varepsilon(y, f(x_i))$ is the so-called loss function measuring the approximate errors between the expected output y_i and the calculated output $f(x_i)$ and C is a regularization constant determining the trade-off between the training error and the generalization performance. The second term, $(1/2)\|w\|^2$, is used as a measurement of function flatness. Introducing slack variables ξ, ξ^* transforms (2) into the following constrained function:

$$\begin{aligned} \min_{w,b,\xi} \quad & J = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ \text{s.t.} \quad & y_i - w\phi(x_i) - b \le \varepsilon + \xi_i \\ & w\phi(x_i) + b - y_i \le \varepsilon + \xi_i^* \\ & \xi, \xi_i^* \ge 0, \quad i = 1, 2, \ldots, l. \end{aligned} \tag{3}$$

Using the duality principle, (3) could be changed to

$$\begin{aligned} \max_{\alpha,\alpha^*} \quad & \left\{ L_D = -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle x_i \cdot x_j \rangle \right. \\ & \left. - \varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*) \right\} \\ \text{s.t.} \quad & \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \\ & \alpha_i, \alpha_i^* \in [0, C], \quad i = 1, 2, \ldots, l. \end{aligned} \tag{4}$$

Although the nonlinear function ϕ is usually unknown, all computations related to ϕ can be reduced to the form $\phi(x)^T\phi(y)$, which can be replaced with a so-called kernel function $K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$ that satisfies Mercer's condition. Then, (1) becomes

$$f(x, \alpha_i, \alpha_i^*) = \sum_{x_i \in \mathrm{SV}} (\alpha_i - \alpha_i^*) K(x_i, x) + b. \tag{5}$$

In (5), Lagrange multipliers α_i and α_i^* satisfy the equalities:

$$\alpha_i \times \alpha_i^* = 0, \quad \alpha_i \ge 0,\ \alpha_i^* \ge 0,\ i = 1, \ldots, l. \tag{6}$$

Those vectors with $\alpha_i \ne 0$ are called support vectors, which contribute to the final solution.

The kernel functions are used to project the sample data into a high-dimensional feature space and then to find the optimal separation plane in it. Kernel functions used by SVM can be divided into global and local kernels. Smits and Jordaan studied representative mapping properties of global kernel function (polynomial kernel function) and local kernel function (RBF kernel) and proposed a mixed kernel function. It was shown that the mixed kernel function-based SVM has strong learning and generalization ability.

This paper selects two polynomial kernel functions and RBF kernel function to produce a hybrid kernel function given by

$$K_{\mathrm{mix}} = \rho K_{\mathrm{poly}} + (1 - \rho) K_{\mathrm{RBF}}, \tag{7}$$

where $K_{\mathrm{poly}} = (x \cdot x_i + 1)^2$ is the polynomial kernel function; $K_{\mathrm{RBF}} = \exp(-\|x - x_i\|^2/\sigma^2)$ is the RBF kernel function; ρ $(0 \le \rho \le 1)$ adjusts the sizes of two kernel functions.

The parameters in the mixed kernel function SVM such as C and the width coefficient in the kernel function $K(x, x_i)$ exert a considerable influence on the performance of SVM. A large or small value of C may lead to degraded generalization ability of SVM. The value of C indicates the error expectation in the classification process of the sample data, and it affects the number of support vectors generated by the classifier, thereby affecting the generalization error of the classifier. If the value of C is too big, the separating error is high, the number of support vectors is small, and vice versa [14]. The parameters in the kernel function reflect the characteristics of the training data, and they also affect the generalization of SVM. Therefore, only after the choice of all these parameters is correctly made can the SVM achieve its best possible performance.

3. Improved Particle Swarm Optimization (IPSO)

The standard PSO algorithm has fast convergence speed, but the speed of particle gets more and more slowly later [15, 16]. A particle updates its velocity and location according to the following formula:

$$V_i^{(t+1)} = \omega V_i^{(t)} + c_1 r_1 \left(P_{i\mathrm{best}}^{(t)} - X_i^{(t)}\right) + c_2 r_2 \left(P_{g\mathrm{best}}^{(t)} - X_i^{(t)}\right)$$
$$X_i^{(t+1)} = X_i^{(t)} + V_i^{(t+1)}, \tag{8}$$

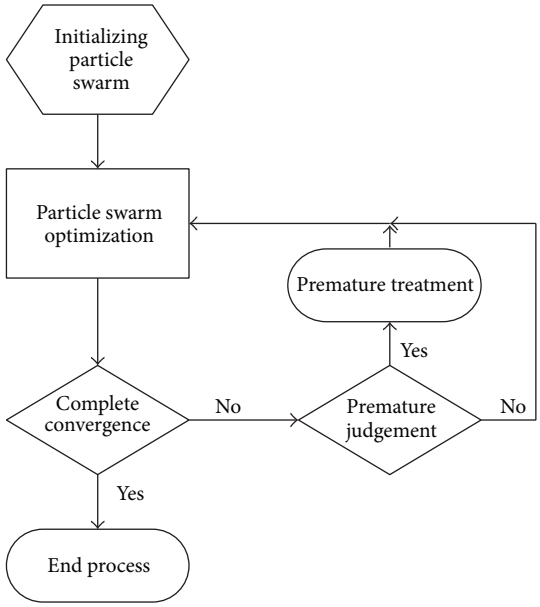


Figure 1: Flowchart of IPSO.

where $V_i^{(t)}$ and $V_i^{(t+1)}$ represent the present and next velocity of particle i and $X_i^{(t)}$ and $X_i^{(t+1)}$ are the present and next position of particle i.

In order to reduce the possibility in the evolutionary process of particles leaving the search space, the particle velocity is usually limited to a certain range $V_i \in [-V_{\max}, V_{\max}]$; P_i is the best previous position of particle i; P_g is the best position among all particles in the population $X = [X_1, X_2, \ldots, X_N]$; r_1 and r_2 are random real numbers in the range of $[0, 1]$; c_1 and c_2 are acceleration constants; ω is the internal weight coefficient (i.e., the impact of the previous velocity of particle on its current one).

Because it is easy for the standard PSO to fall into local minima, which is called the phenomenon of being premature [17–19], improved particle swarm optimization algorithm is proposed which is the theory of detecting premature convergence. This algorithm combines the method of analyzing and treating premature convergence to avoid the premature phenomenon throughout the whole algorithm. The whole algorithm process is shown in Figure 1.

3.1. Premature Detection. The literature [20] pointed out that when the particle swarm has premature convergence, the particles in the swarm will appear as "aggregation" phenomenon, and the position of the particles determines the adaptability of particles, so the state of particle swarm can be tracked by the overall changes in the fitness of all particles.

The number of particles in the particle swarm is assumed to be m, F_i is the fitness of i particle, F_{avg} is the mean fitness, and δ^2 is the particle swarm colony fitness variance defined as

$$\delta^2 = \sum_{i=1}^{m} \frac{F_i - F_{\text{avg}}}{F}, \tag{9}$$

where F is a normalized scaling factor, and it can restrict the size of δ^2. The value of the F is determined using the following formula:

$$F = \begin{cases} \max\limits_{1\le i\le m} \left|F_i - F_{\text{avg}}\right|, & \max\limits_{1\le i\le m} \left|F_i - F_{\text{avg}}\right| > 1 \\ 1, & \text{other.} \end{cases} \tag{10}$$

Equation (9) shows that the variance of the colony fitness is the reflection of all particles in the swarm "aggregation" degree. The smaller the δ^2, the greater "aggregates" level particle swarm is. If the algorithm does not meet the termination condition, the "aggregation" will enable the group to lose diversity in early state. A premature detection can be made when $\delta^2 < H$ (H is a given constant).

3.2. Premature Treatment. For premature state of the particle, the method combined by chaos algorithm and particle swarm algorithm can set up the position and velocity of particles and make it jump out of local minima. Use the classical logistic equation to achieve chaos sequence via

$$y_i' = \lambda y_i (1 - y_i), \tag{11}$$

where y_i is the random number uniformly sampled from $(0, 1)$; λ is control parameter, when it takes the value of 4, the Logistic equation is in complete chaos [13]. Then, it will be introduced into the optimization space by the following formula:

$$x_i = a_i + y_i' (b_i - a_i), \tag{12}$$

where $[a_i, b_i]$ is the range of variables. A global optimal solution has no practical significance to the algorithm when particle swarm runs into local optimum. We can put forward the renewal equation of particle premature condition after combining (11) and (12):

$$V_i^{(t+1)} = \omega V_i^{(t)} + c_1 r_1 \left(P_{i\text{best}}^{(t)} - X_i^{(t)}\right) + \left(-V_{\max} + 2y_i' V_{\max}\right),$$
$$X_i^{(t+1)} = X_i^{(t)} + V_i^{(t+1)}, \tag{13}$$

where $[-V_{\max}, V_{\max}]$ is the range of particle velocity. Thus, the particle can jump out of local optimum and back into the particle swarm optimization iteration.

3.3. Optimization Algorithm. In order to improve the precision and ability of generalization of the mixed kernel function, the following variance function can be used for the fitness function of IPSO algorithm which can respond to SVM regression performance directly. Consider

$$F_{\text{fitness}} = \frac{1}{n} \sum_{i=1}^{n} (f_i - y_i)^2, \tag{14}$$

where f_i is the predictive value, y_i is the measured value, and n is the number of samples. The flow diagram of optimization algorithm is shown in Figure 2. The detailed process of modeling is as follows.

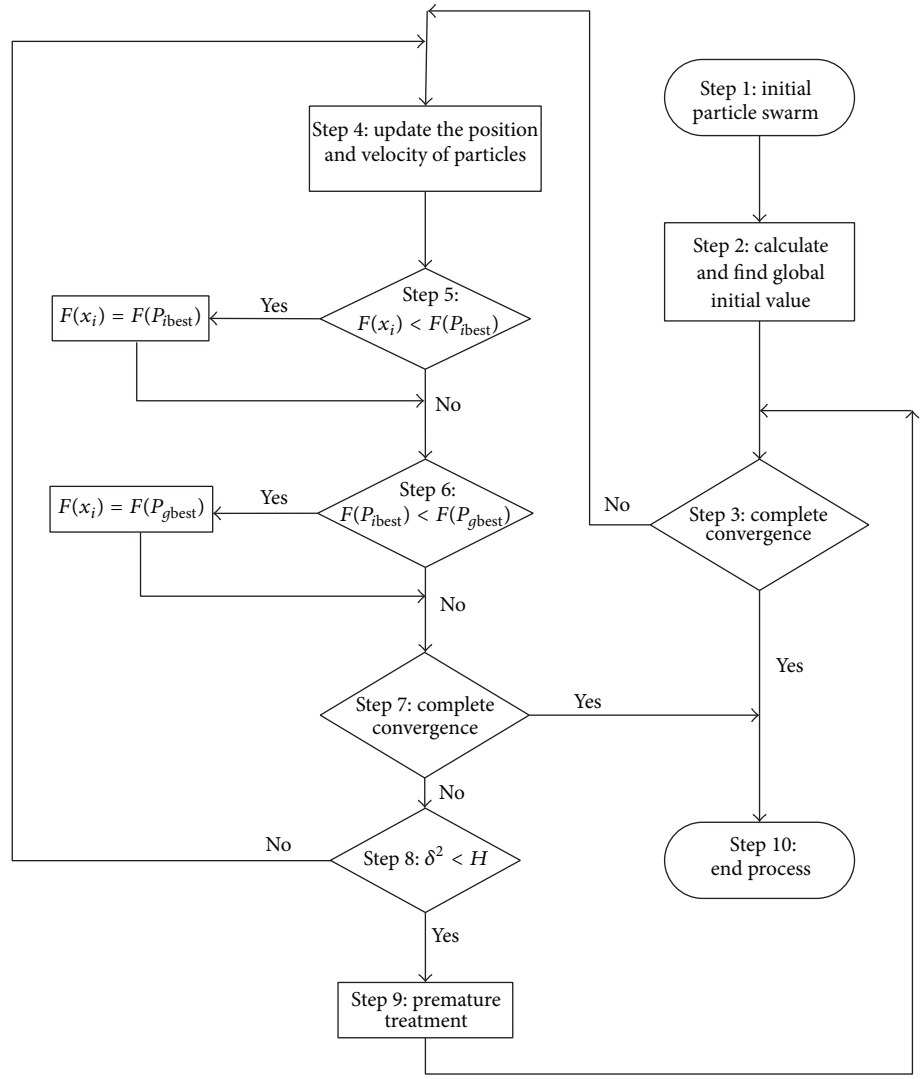


Figure 2: Flowchart of optimization algorithm.

Step 1. Initialize particle swarm parameters (ρ, σ, C), swarm size, and the maximum number of iterations and determine the weight factor ω and particle swarm flight speed range $[-V_{\max}, V_{\max}]$. Set a loss function parameter ε of SVM and judgment criterion of global convergence and premature convergence.

Step 2. The individual value of each particle $P_{i\text{best}}$ is set to the current position, and then fitness is calculated for each particle, the fitness value of individual extreme corresponding to the best particle as the global extreme initial $P_{g\text{best}}$.

Step 3. Judging the convergence criterion is satisfied. If the algorithm is satisfied, then execute Step 10; otherwise go to Step 4.

Step 4. Use (2) (the standard PSO algorithm) to execute the iterative calculation and update the position and velocity of particles.

Step 5. Compare $F(X_i)$ and $F(P_{i\text{best}})$, and if the $F(X_i) < F(P_{i\text{best}})$, update $P_{i\text{best}}$.

Step 6. Compare the updated $F(P_{i\text{best}})$ and $F(P_{g\text{best}})$, and if $F(P_{i\text{best}}) < F(P_{g\text{best}})$ then update $P_{g\text{best}}$.

Step 7. Determine whether the convergence criterion is satisfied. If it is satisfied, then execute Step 10; otherwise go to Step 8.

Step 8. Evaluate (3) and (4) to calculate the variance δ^2 of the population's fitness and then judge whether the establishment of $\delta^2 < H$, and if established turn to Step 9 for premature treatment; otherwise turn to Step 4.

Step 9. According to (7) for premature treatment of falling into local optimal particle, the particle swarm escapes from local optima; then turn to Step 3.

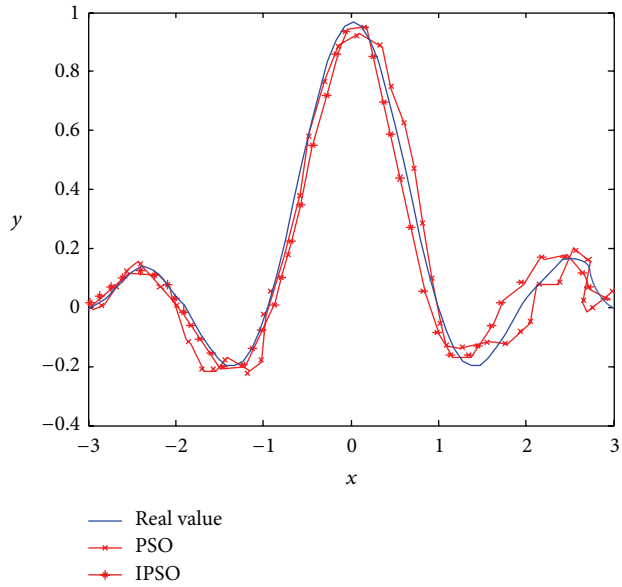


Figure 3: Simulation results of $f(x)$.

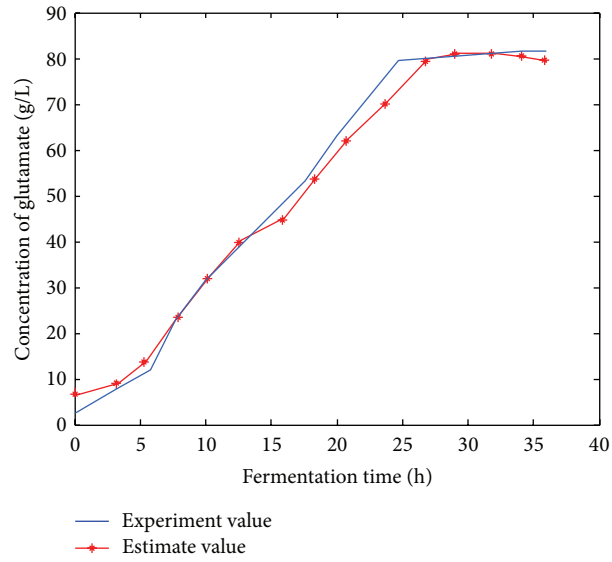


Figure 4: Predictive result of glutamate concentration.

Table 1: Performance comparison of parameters selection of SVM by different methods.

Method	ρ	σ	C	Mean error
Particle swarm optimization	0.0102	0.2015	756	0.0306
Improved particle swarm optimization	0.0100	0.2836	884	0.0201

Step 10. Output the particle swarm optimal value, and the algorithm terminates.

4. Modeling and Simulation

4.1. Function Fitting Simulation. In order to test the effectiveness of the algorithm, one-dimensional function is selected to simulate

$$f(x) = \frac{\sin(x)}{x} + \xi, \quad x \in [-3, 3], \tag{15}$$

where ξ is the Gaussian noise with zero mean and variance of 0.1. In order to minimize the fitness function and optimize the mixed kernel function of SVM parameters using IPSO algorithm, we take 50 sets of data to constitute a hybrid kernel function training sample of SVM in the input variable domain. Among them, mixed kernel function SVM uses an insensitive loss function which is $\varepsilon = 0.1$; ρ, σ and C are initialized, respectively, in $[0, 1]$, $[0.01, 1.0]$, and $[1, 1000]$; the population size is 20, and the premature judgment constant H is 1. The maximum number of iterations is 50 times, and $F(P_{g\text{best}}) \leq 10^{-3}$ which is the fitness value as global convergence. The simulation results are given in Figure 3 and Table 1.

The simulation results show that the mixed kernel function of SVM model has a high accuracy constructed by IPSO high parameter optimization. It shows the feasibility and effectiveness of the improved particle swarm optimization algorithm.

4.2. Modeling and Application Based on IPSO. In this paper, a forecast model of glutamate fermentation concentration can be built by a mixed kernel function of SVM based on IPSO; the input which is the current time in fermentation process (H), dissolved oxygen (DO), concentration of glutamate (g/L), cell concentration (g/L), and residual sugar concentration (g/L) can estimate next moment of glutamate and residual sugar concentration.

In the experiment, in the 6 batches of production data, each number represents a complete fermentation process, in which the 4 batches of data are for training, and the other 2, used as the test sample to test its generalization ability. Modeling mixed kernel function SVM using an insensitive loss function $\varepsilon = 0.08$, the general range of initializing particle swarm parameters is given: $\sigma = [0, 1]$, $\rho = [0, 1]$, $C = [100, 1000]$. Based on the training data for learning and the optimization and adjustment for mixed kernel function of SVM parameter by IPSO algorithm, it can get $\sigma = 0.4862$, $\rho = 0.0421$, $C = 861$. Using the parameter set of glutamate fermentation model trained to predict test data, the results are as shown in Figures 4 and 5.

Figure 4 shows the changing behavior in glutamate production from 6 glutamate fermentation experiments. The production of glutamate increased in a nonlinear way over the fermentation period from 2 h to 38 h; glutamate increased very slowly in the early period during 2–5 h. It increased quickly from 20 to 60 g/L in 9–20 h and then increased slowly after 20 h. After 34 h, the production of glutamate kept stable at 70–75 g/L. Figure 5 shows the changing behavior of residual sugar concentration in glutamate fermentation experiments. The early period (usually the former 5 hours) is bacteria growth stage and the sugar concentration is

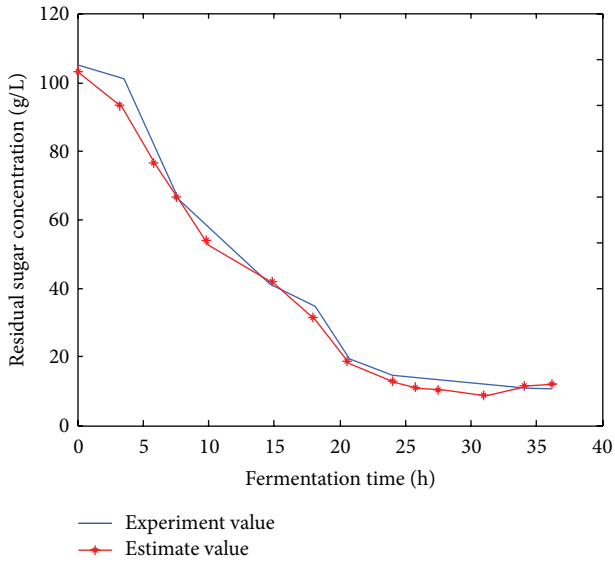


Figure 5: Predictive result of residual sugar concentration.

higher. About 9 hours later fermentation turned into acid producing period, bacteria growth slowing down and a large number of glutamic acids accumulating. Most of sugar in the culture medium has been burned off, requiring a continuous consumption of nutrients.

In order to maintain stable environment for the growth of bacteria and prevent severe changes in sugar concentration which can disturb bacteria physiological metabolism, we can take a fed-batch flow rate strategy and keep the glucose concentration on a constant level, referring to the estimate results.

The average prediction error for glutamate concentration is 2.61%. The average prediction error residual sugar concentration is 3.82%. It can then be seen that this method has good modeling effect.

5. Conclusion and Future Work

In this paper, a new IPSO-based optimization algorithm was used to select the optimal parameters for the mixed kernel function of SVM. The simulation results show that the mixed kernel function-based SVM with optimally chosen parameters is accurate and reliable, and the optimum mixed kernel function of SVM model based on structural parameters has better learning precision.

Nomenclature

SVM: Support vector machine
IPSO: Improved particle swarm optimization
DO: Dissolved oxygen
LOO: Leave-one-out
GD: Gradient descent
GA: Genetic algorithm
PSO: Particle swarm optimization
RBF: Radial basis function
T: Set of training data
$R(C)$: Regularized risk function
C: Regularization constant
K_{mix}: Hybrid kernel function
K_{poly}: Polynomial kernel function
K_{RBF}: RBF kernel function
$V_i^{(t)}$: Present velocity of particle i
$V_i^{(t+1)}$: Next velocity of particle i
$X_i^{(t)}$: Present position of particle i
$X_i^{(t+1)}$: Next position of particle i
P_i: Best previous position of particle i
P_g: Best position of particle i among all particles
c_1, c_2: Acceleration constants
ω: Internal weight coefficient
m: Number of particles in particle swarm
F_i: The fitness of i particle
F_{avg}: The mean fitness
δ^2: Particle swarm colony fitness variance
F: Normalized scaling factor
y_i: Random number
λ: Control parameter
F_{fitness}: Fitness function
f_i: Predictive value
y_i: Measured value
n: Number of samples
ξ: Gaussian noise.

Conflict of Interests

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61273131) and the Fundamental Research Funds for the Central Universities (JUDCF13040).

References

[1] J. Xiao, Z. P. Shi, P. Gao, H. Feng, Z. Duan, and Z. Mao, "Online optimization of glutamate production based on balanced metabolic control by RQ," *Bioprocess and Biosystems Engineering*, vol. 29, no. 2, pp. 109–117, 2006.

[2] X. Feng, T. Yu, and J. Wang, "Nonlinear GPC with in-place trained RLS-SVM model for DOC control in a fed-batch bioreactor," *Chinese Journal of Chemical Engineering*, vol. 20, no. 5, pp. 988–994, 2012.

[3] X. J. Gao, P. Wang, Y. S. Qi, Y. Zhang, H. Q. Zhang, and A. J. Yan, "An optimal control strategy combining SVM with RGA for improving fermentation titer," *Chinese Journal of Chemical Engineering*, vol. 18, no. 1, pp. 95–101, 2010.

[4] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, Berlin, Germany, 1995.

[5] Y. F. Li, Z. F. Wang, and J. Q. Yuan, "On-line fault detection using SVM-based dynamic MPLS for batch processes," *Chinese Journal of Chemical Engineering*, vol. 14, no. 6, pp. 754–758, 2006.

[6] Z. W. Liu, H. R. Cao, X. F. Chen, Z. J. He, and Z. J. Shen, "Multi-fault classification based on wavelet SVM with PSO algorithm to analyze vibration signals from rolling element bearings," *Neurocomputing*, vol. 99, pp. 399–410, 2013.

[7] N. Cristianmini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, Cambridge, UK, 2000.

[8] G. F. Smits and E. M. Jordaan, "Improved SVM regression using mixtures of kernels," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '02)*, pp. 2785–2790, May 2002.

[9] S.-X. Du and T.-J. Wu, "Support vector machines for regression," *Journal of System Simulation*, vol. 15, no. 11, pp. 1580–1585, 1633, 2003.

[10] J. H. Lee and C. J. Lin, *Automatic Model Selection for Support Vector Machines*, Taiwan University, Taipei, Taiwan, 2000.

[11] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, no. 1–3, pp. 131–159, 2002.

[12] C. L. Zheng and L. C. Jiao, "Automatic parameters Selection for SVM based on GA," in *Proceedings of the 5th World Congress on Intelligent Control and Automation (WCICA '04)*, pp. 1869–1872, Piscataway, NJ, USA, June 2004.

[13] A. P. Mariano, C. B. B. Costa, D. D. F. de Angelis et al., "Optimisation of a fermentation process for butanol production by particle swarm optimisation (PSO)," *Journal of Chemical Technology and Biotechnology*, vol. 85, no. 7, pp. 934–949, 2010.

[14] W. Wang, X. M. Guo, S. Y. Wang et al., "On the method of Kernel function selection in support vector machine," *Journal of Liaoning Normal University (Natural Science Edition)*, vol. 31, no. 1, pp. 1–4, 2008.

[15] Q. Wu, "A self-adaptive embedded chaotic particle swarm optimization for parameters selection of Wv-SVM," *Expert Systems with Applications*, vol. 38, no. 1, pp. 184–192, 2011.

[16] X. Li, S.-D. Yang, and J.-X. Qi, "New support vector machine optimized by improved particle swarm optimization and its application," *Journal of Central South University of Technology*, vol. 13, no. 5, pp. 568–572, 2006.

[17] Y. J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Austrilia, December 1995.

[18] R. Suzuki, F. Kawai, C. Nakazawa, T. Matsui, and E. Aiyoshi, "Parameter optimization of model predictive control by PSO," *Electrical Engineering in Japan*, vol. 178, no. 1, pp. 40–49, 2012.

[19] Y.-B. Li, N. Zhang, and C.-B. Li, "Support vector machine forecasting method improved by chaotic particle swarm optimization and its application," *Journal of Central South University of Technology*, vol. 16, no. 3, pp. 478–481, 2009.

[20] H. Y. Liu, Y. Lin, and J. S. Zhang, "A hybrid particle swarm optimization based on chaos strategy to handle local convergence," *Computer Engineering and Applications*, no. 13, pp. 77–79, 2006.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 814084, 5 pages
http://dx.doi.org/10.1155/2015/814084

Research Article

Structure Optimal Design of Electromagnetic Levitation Load Reduction Device for Hydroturbine Generator Set

Qingyan Wang,[1,2] Hongzhong Ma,[1,2] Shengrang Cao,[1,2] and Bingyan Chen[1,3,4]

[1] *College of Energy and Electrical Engineering, Hohai University, Nanjing, China*
[2] *Research Center for Renewable Energy Generation Engineering, Ministry of Education, Beijing, China*
[3] *Hohai University Nantong Institute of Marine and Offshore Engineering, Nantong, China*
[4] *Jiangsu Province Key Laboratory of Environmental Engineering, Nanjing, China*

Correspondence should be addressed to Qingyan Wang; wqy6990@foxmail.com

Received 8 October 2014; Revised 8 February 2015; Accepted 20 February 2015

Academic Editor: Gerhard-Wilhelm Weber

Thrust bearing is one part with the highest failure rate in hydroturbine generator set, which is primarily due to heavy axial load. Such heavy load often makes oil film destruction, bearing friction, and even burning. It is necessary to study the load and the reduction method. The dynamic thrust is an important factor to influence the axial load and reduction design of electromagnetic device. Therefore, in the paper, combined with the structure features of vertical turbine, the hydraulic thrust is analyzed accurately. Then, take the turbine model HL-220-LT-550, for instance; the electromagnetic levitation load reduction device is designed, and its mathematical model is built, whose purpose is to minimize excitation loss and total quality under the constraints of installation space, connection layout, and heat dissipation. Particle swarm optimization (PSO) is employed to search for the optimum solution; finally, the result is verified by finite element method (FEM), which demonstrates that the optimized structure is more effective.

1. Introduction

For vertical shaft hydroturbine generator set, the axial load is mainly composed of two parts, total gravity of rotating parts and hydraulic axial thrust. In large and medium hydropower units, it could be up to thousands of tons. With the development of unit capacity, axial load shows the growing tendency, and some in actual operations have exceeded rated loads, which would lead to serious deformations of thrust tiles, oil film breakdown, and mechanical strength aging. In particular, under long-term loading conditions, oil film breakdown would cause in future mechanical abrasion and heat problems [1–3]; thrust bearing has been regarded as one of the most complex components in hydropower plant [3–5]. In recent years, bearing faults happen all the time, such as in Gezhouba, Baishan, and Wujiangdu, whose characteristics mainly show high temperature in certain areas of pads or burning-out. Reference [2] indicated that the bearing failure rate takes up almost 60% of the total mechanical faults in the whole unit. With the increasing proportion of hydropower in power system, the reliability and stability of unit operation become much more pressing concerns to deal with. How to develop new technical methods to reduce axial load and enhance reliability have become a key technology in the design of hydropower generator unit and capacity-increasing transformation.

Ma et al. proposed a new method by using permanent magnetic and electromagnetic levitation devices to reduce the axial load [2–4]. In electromagnetic part, the main idea is to use the electromagnetic towing force to counteract most of the downward axial load, about 80%; in this way, thrust bearing just needs to bear about 20% of the total load, the frictional loss of oil films could be declined, and failure rate is accordingly reduced. References [3, 4] have analyzed the structure, feasibility, and predomination when compared with traditional supporting forms. However, hydraulic axial thrust has a big impact on axial load and reduction design; it is necessary to be analyzed accurately. Besides, the structure parameters of the device are interactional, which determine

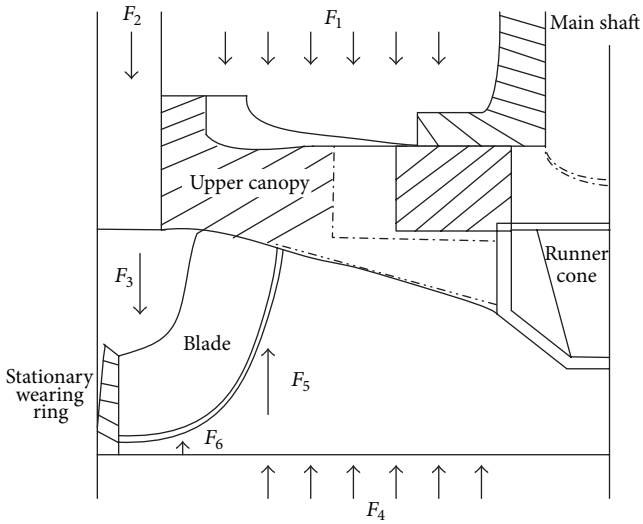


Figure 1: Structure of turbine runner.

the performance of the device. Therefore, to improve the efficiency, combined with the thrust analysis and original structure, a mathematical model of electromagnetic device is established and PSO is employed to search for the optimum relation.

2. Hydraulic Axial Thrust of Turbine

According to stress and structure analysis of the runner (shown in Figure 1), the hydraulic thrust is composed of water pressure on upper canopy, runner band, ingress, exit, the inner surface of the cavity flow path, and the buoyancy of runner [6, 7]. If there is no outlet entrance on the surface of upper canopy, the force F_1 on the upper canopy is associated with static pressure of upper wearing ring ingress, which can be described as follows:

$$F_1 = \gamma\pi \left[\frac{2H_1 g - \left(K_0 \pi n R_1/30\right)^2}{2g} \left(R_1^2 - R_0^2\right) + \left(\frac{K_0 \pi n}{30}\right)^2 \left(\frac{R_1^4 - R_0^4}{4g}\right) \right], \tag{1}$$

where γ is the unit weight of water (N/m^3); g is acceleration of gravity (N/s^2); K_0 is angular velocity coefficient of water on upper canopy; generally, $K_0 = 0.5$ [6]; R_1 is radius of upper wearing ring ingress (m); H_1 is static head of ingress of upper wearing ring (m) [7], whose specific expression is

$$H_1 = H_0 - \frac{1}{4\pi g R_1} \sqrt{\frac{Q^2}{b^2} + \frac{3600\eta_t^2 g^2 H^2}{n_r^2}}, \tag{2}$$

where H_0 is rated head (m); Q is hydraulic turbine discharge (m^3/s); b is turbine blade height (m); η_t is the efficiency of hydraulic turbine; H is working head (m); n_r is turbine rated speed.

The force F_2 on runner band depends on the static head of the lower wearing ring ingress H_2 (m), the diameter of upper wearing ring ingress D_1 (m), and the diameter of lower wearing ring ingress D_2 (m), and it is often expressed as

$$F_2 = \frac{\gamma\pi\left(H_1 + H_2\right)\left(D_2^2 - D_1^2\right)}{8}. \tag{3}$$

On turbine blade inlet, the force F_3 is given in (4), where $V_{1r} = Q/(\pi D_1 b)$ is the axial component of the absolute speed on the upper wearing ring ingress [7] and φ is the inclination angle between runner inlet section and main shaft:

$$F_3 = \frac{\gamma Q V_{1r} \sin\varphi}{g}. \tag{4}$$

On blade outlet, the force F_4 is in connection with draft height H_s and exit velocity V_2:

$$F_4 = \gamma\left(H_s - \frac{V_2^2}{2g}\right)\left(\frac{\pi D_2^2}{4}\right). \tag{5}$$

According to the flow momentum theorem on runner inlet and outlet, the force F_5 on runner cavity and the upward buoyant force on the runner submerged in the water F_6 can be described as follows:

$$F_5 = \frac{\gamma Q V_2}{g}, \qquad F_6 = \gamma\frac{G_r}{\gamma_t}, \tag{6}$$

where G_r is the runner weight (kg) and γ_t is the specific gravity of runner. Combined with structure parameters of the turbine model HL-220-LT-550 in Ankang hydropower station, each force component is calculated and shown in Table 1.

Table 1: Hydraulic thrust of turbine.

Parameter	Data
Rated output N (MW)	204.1
Rated flow Q (m^3/s)	304
Rated head H (m)	76.2
Rated speed n_r (r/min)	107
Draft height H_s (m)	−4.5
Gravity of rotating parts G (kN)	8722
Force component F_1 (kN)	6337
Force component F_2 (kN)	1982
Force component F_3 (kN)	18.2
Force component F_4 (kN)	−548
Force component F_5 (kN)	3436
Force component F_6 (kN)	13

The hydraulic thrust under the rated condition is $F_{\text{th}} = F_1 + F_2 + F_3 - F_4 - F_5 - F_6 = 5436$ kN, and the total axial load is $F_T = G + F_{\text{th}} = 14158$ kN, about 1387 tons. It can be seen that the hydraulic thrust is up to 62% of the gravity of rotating parts and up to 39% of the total axial load. So, the hydraulic thrust, as one important component of the axial load, should not be ignored. The thrust force varies with water head, speed, and flow, whose value on other work conditions could be evaluated according to (1)–(6).

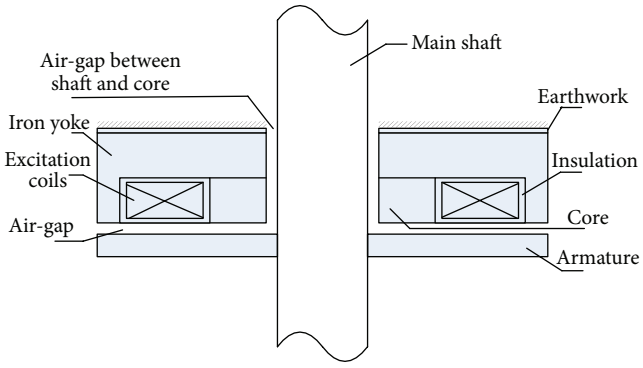


Figure 2: Structure of hybrid magnetic electromagnetic device.

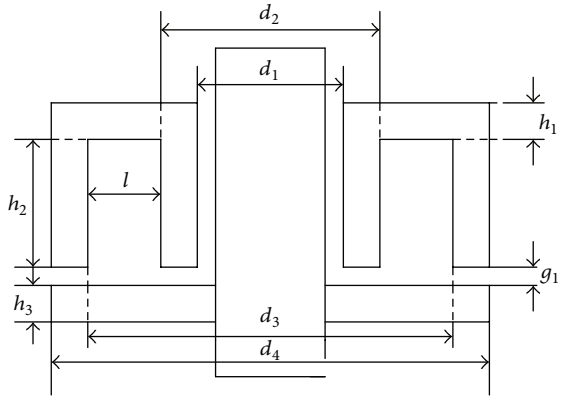


Figure 3: The profile diagram of electromagnetic device.

3. Modeling of Electromagnetic Levitation Device

3.1. Principle of Electromagnetic Levitation Device. The load shedding device (i.e., DC sucker electromagnetic levitation device) has the character of compact conformation [3]. The device could be installed outer flank of main shaft, which consists of the rotating armature iron (connected with the main shaft) and fixed iron core (fixed by the earthwork), as shown in Figure 2. The excitation coils are placed in soft magnetic materials (fixed iron core) and contained in epoxy, which can keep good insulation and is suitable for high humidity environment.

When the device is electrified, armature would be attracted, and there exists an upward pull. Due to the connection of armature and main shaft, the upward attractive force is passed to main shaft, the rotating parts of hydrogenerator set could be subject to the pulling force, and the force can be adjusted according to the actual operation. The profile diagram for the electromagnetic levitation device is shown in Figure 3.

Assumed to neglect the effect of saturation and flux leakage, the electromagnetic force generated by electromagnetic device can be expressed as follows [8, 9]:

$$F_{\text{ele}} = \frac{1}{2\mu_0\pi}\left(\frac{NI}{R_m}\right)^2\left(\frac{4}{\left(d_2^2 - d_1^2\right)} + \frac{4}{2\mu_0\pi\left(d_4^2 - d_3^2\right)}\right). \quad (7)$$

R_m is magnetic reluctance and can be expressed as

$$R_m = \frac{g}{\mu_0}\left(\frac{1}{S_1} + \frac{1}{S_2}\right) + \frac{h_2 + h_1/2 + h_3/2}{\mu}\left(\frac{1}{S_1} + \frac{1}{S_2}\right) + \frac{1}{2\pi\mu}\left(\frac{1}{h_1} + \frac{1}{h_3}\right)\ln\frac{d_{34}}{d_{12}}, \quad (8)$$

where $d_{12} = (d_1 + d_2)/4$ and $d_{34} = (d_3 + d_4)/4$; μ is the magnetic permeability of iron.

3.2. Mathematical Model. Electromagnetic levitation device uses ring configuration; in order to improve the utilization ratio of iron cores, the areas of inner and outer rings should be identical. Magnetic pressure and system loss are determined by coil turns and exciting current; therefore, assume that the design variable is $X = [d_2, l, h_2, N, I]$. In this paper, take excitation loss and total weight of electromagnet as the objective functions:

$$P_{\text{loss}} = \pi I^2 N \rho d_{23},$$
$$M = \rho_{\text{Fe}}\frac{\pi}{4}\left[\left(d_4^2 h_2 - d_3^2 h_2\right) + \left(d_2^2 h_2 - d_1^2 h_2\right) + \left(d_4^2 - d_1^2\right)\left(h_3 + h_1\right)\right], \quad (9)$$

where ρ_{Fe} is the density of ferrite. According to the design requirement, electromagnetic force should not be less than $F_{\min} = 0.7F_T$, while electromagnetic force should not be larger than $F_{\max} = 0.8F_T$; otherwise, it would cause the instability of the hydrogenerator unit, so

$$0.7F_T < F_{\text{ele}} < 0.8F_T. \quad (10)$$

Structure design should consider restraining factors, such as the actual installation space, cabling requirement, and heat dissipation. To guarantee that the outer and inner magnetic flux intensities are more or less the same, outer and inner ring areas might be equal. So, there exist the following constraints:

$$S_1 - S_2 = 0,$$
$$0.08 < h_2 < 0.25,$$
$$3000 < N < 8000,$$
$$h_2 l \geq \frac{\pi d_w^2 K_y N}{(4K_t)}, \quad (11)$$

where d_w is the wire diameter; k_y is the margin coefficient, generally set as 1.2; k_t is filling factor of the coil and is about 0.5. The wire of electromagnetic device is concerned with the allowed maximum current. In this paper, choose the nominal wire diameter of 1.784 mm. According to the above, the limit condition is given as

$$0 < I < 8.0. \quad (12)$$

4. Optimization Technology and Experiment

The essence of PSO is that letting information about good solutions spread around the population, each particle would

Table 2: Comparison of results before and after optimization.

Parameters	Before optimization	After optimization
d_2 (m)	3.464	3.667
d_3 (m)	4.664	4.563
h_2 (m)	0.12	0.100
N	5000	4163
I (A)	3	3.2614
F_{ele} (kN)	1132.64	1094.723
M (ton)	109.254	108.685
P_{loss} (kW)	4.307	4.291

tend to move to good areas [10, 11]. At each iteration time t, particle i is moved to a new position by adding a velocity term $\mathbf{V}_i(t+1)$ to its current position $\mathbf{X}_i(t+1)$ according to

$$\begin{aligned}\mathbf{V}_i(t+1) &= \omega \mathbf{V}_i(t) + c_1 r_1 \left[X_{Pbest,i} - X_i(t)\right] \\ &\quad + c_2 r_2 \left[X_{Gbest} - X_i(t)\right], \\ \mathbf{X}_i(t+1) &= \mathbf{X}_i(t) + \mathbf{V}_i(t+1),\end{aligned} \tag{13}$$

where $i = 1, 2, \ldots,$ pop and pop is the population size; $X_{Pbest,i}$ is the best position of particle i; X_{Gbest} is the global best position; ω is called inertia weight; c_1 and c_2 are acceleration factors; r_1 and r_2 are random selection of constants among $[0, 1]$. Optimization design ((7) to (12)) is a constrained nonlinear problem; by the constructed penalty functions, the problem can be simplified as follows:

$$\begin{aligned} f_{itness} &= \lambda_1 P_{loss} + \lambda_2 V + L_1 + L_2 \\ \text{s.t.} \quad & d_1 < d_2 \\ & 0 < l \le 0.6 \\ & 0 < I < 8.0 \\ & 3000 < N < 8000 \\ & 0.08 < h_2 < 0.25, \end{aligned} \tag{14}$$

where λ_1, λ_2 are weight coefficients and $\lambda_1 + \lambda_2 = 1$; L_1, L_2 are the penalty functions, whose specific descriptions are shown as

$$\begin{aligned} L_1 &= \begin{cases} 0, & \text{if } \operatorname{sign}\left(h_2 l - \dfrac{\pi d_w^2 K_y N}{4K_t}\right) \ge 0 \\ 10^5 \left| h_2 l - \dfrac{\pi d_w^2 K_y N}{4K_t} \right| & \text{else,} \end{cases} \\ L_2 &= 10^5 \left| S_1 - S_2 \right|. \end{aligned} \tag{15}$$

In this paper, assume that $g = 0.005$ m, $h_1 = h_3 = 0.3$ m, $d_1 = 2$ m, and $d_4 = 5.5$ m. The algorithm parameters are set

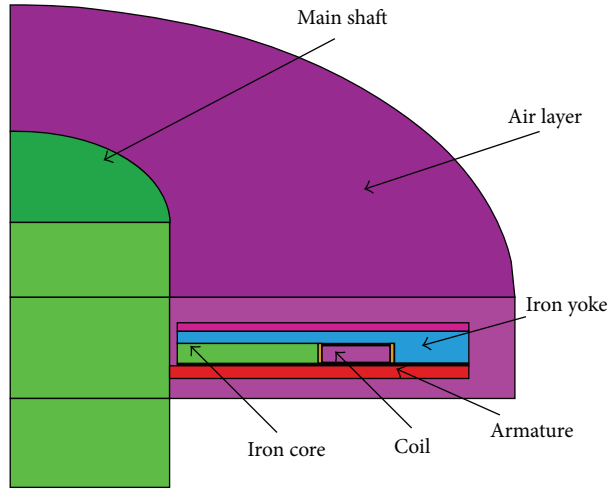


Figure 4: 3D finite element model of electromagnetic device.

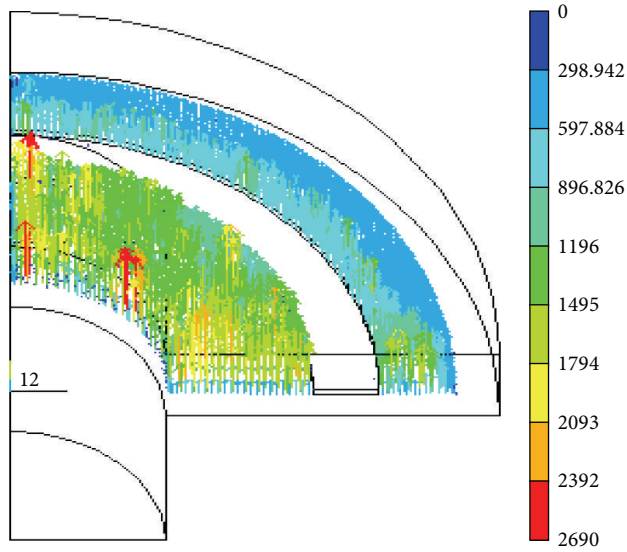


Figure 5: Vector distribution of electromagnetic force.

as follows: $\lambda_1 = \lambda_2 = 0.5$, $t_{max} = 200$, pop = 36, $\omega = 0.8$, and $c_1 = c_2 = 0.8$. Test for 50 times; select the best result, shown in Table 2.

To verify the result, use ANSYS software [12] to simulate 1/4 of this device (shown in Figure 4). The simulation result is $F'_{ele} = 1053.46$ kN, the error between FEM and numerical analysis is less than 5%, and the electromagnetic force distribution on the armature is shown in Figure 5.

From Table 2, the electromagnetic forces between pre- and postoptimization have little difference, not exceeding 4%, but the required magnetic potential is decreased about 1423 ampere turns, saving about 10.5%. When guaranteeing the load reduction requirement, the system loss and total gravity are decreased by PSO. The reason is that, after optimization technology, the effective pole areas grow from 12.949 m^2 to 14.818 m^2. When the air-gap keeps in step, there are two ways to increase electromagnetic force, by raising

the magnetic pressure or effective pole areas. The magnetic flux density is limited by material properties, and raising magnetic pressure would be susceptible to magnetic saturation problem. Meanwhile, there might be increase in the system loss. Therefore, on the condition of actual allowed space, increasing effective pole areas by optimization technology would meet the design task and simultaneously reduce the excitation loss and manufacturing cost.

5. Conclusion

The analytic process about hydraulic thrust has been described in this paper, whose calculation is a critical factor to the axial load and reduction design of the hydroturbine generator set. According to the calculation of turbine model HL-220-LT-550, the design requirement, and structure restriction of electromagnetic levitation load reduction device, the mathematical model is built. Then, use PSO to search for the optimum solution, and, by comparison, the optimization solution makes the electromagnetic device more efficient and of more performance. The analysis of hydraulic thrust can be applied to other similar turbines; the structural optimization method also could be extended to similar electromagnetic devices.

Conflict of Interests

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 51177039), Doctoral Fund of Ministry of Education of China (Grant no. 20090094110011), the Nantong Science and Technology Project, Nantong, China (Grant no. BK2014024), and the Open Project of Jiangsu Province Key Laboratory of Environmental Engineering, Nanjing, China (Grant no. KF2014001). Meanwhile, the authors would like to thank Professor Ping Ju for his valuable discussions in electromagnetic device modeling.

References

[1] J. Ferguson, "GE hydro thrust bearings-applying development work to upgrading existing bearings," in *Proceedings of the IEEE International Electric Machines and Drives Conference (IEMDC '99)*, pp. 809–811, May 1999.

[2] H. Z. Ma, Q. Y. Wang, and P. Ju, "Study of the load reduction for hydro-generator bearing by hybrid magnetic levitation," *International Journal of Applied Electromagnetics and Mechanics*, vol. 43, no. 3, pp. 215–226, 2013.

[3] H. Z. Ma, B. Wang, and P. Ju, "Feasibility of mixed magnetic-levitation weight support system of hydroelectric generating set," *Journal of Hohai University: Natural Sciences*, vol. 38, no. 4, pp. 467–471, 2010.

[4] H. Ma, Z. Geng, Q. Wang, and Z. Zhang, "Analysis and calculation for heating and temperature rise of magnetic levitation device," *Proceedings of the Chinese Society of Electrical Engineering*, vol. 32, no. 30, pp. 126–132, 2012.

[5] Z. P. Xia, Z. Q. Zhu, and D. Howe, "Analytical magnetic field analysis of halbach magnetized permanent-magnet machines," *IEEE Transactions on Magnetics*, vol. 40, no. 4 I, pp. 1864–1872, 2004.

[6] D. Y. Liu, G. H. You, F. Wang, and J. Zhang, "Calculation and analysis of axial thrust acting on turning wheel of flow-mixing reversible hydraulic turbines," *Journal of Hohai University: Natural Sciences*, vol. 5, 2004.

[7] R. F. Li, X. P. Li, X. Qian, T. Xu, Y. Wang, and Y. Zeng, "Calculation of axial water thrust of the inverted-installation runner of mixed-flow turbine," *Journal of Waterpower*, vol. 5, pp. 50–52, 2001.

[8] N. R. Nannapaneni, *Fundamentals of Engineering Electromagnetics*, Mechanical Industry Press, Beijing, China, 2006.

[9] E. Atienza, M. Perrault, F. Wurtz, V. Mazauric, and J. Bigeon, "Methodology for the sizing and the optimization of an electromagnetic release," *IEEE Transactions on Magnetics*, vol. 36, no. 4 I, pp. 1659–1663, 2000.

[10] M. Kaucic, "A multi-start opposition-based particle swarm optimization algorithm with adaptive velocity for bound constrained global optimization," *Journal of Global Optimization*, vol. 55, no. 1, pp. 165–188, 2013.

[11] P. Umapathy, C. Venkataseshaiah, and M. S. Arumugam, "Particle swarm optimization with various inertia weight variants for optimal power flow solution," *Discrete Dynamics in Nature and Society*, vol. 2010, Article ID 462145, 15 pages, 2010.

[12] J.-A. Duan, H.-B. Zhou, and N.-P. Guo, "Electromagnetic design of a novel linear maglev transportation platform with finite-element analysis," *IEEE Transactions on Magnetics*, vol. 47, no. 1, pp. 260–263, 2011.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 639787, 10 pages
http://dx.doi.org/10.1155/2015/639787

Research Article

A New Hybrid Algorithm to Solve Winner Determination Problem in Multiunit Double Internet Auction

Mourad Ykhlef[1] and Reem Alqifari[2]

[1]*Department of Information Systems, College of Computer and Information Sciences, King Saud University, P.O. Box 51178, Riyadh 11543, Saudi Arabia*
[2]*Information Technology Department, College of Computer and Information Sciences, P.O. Box 51178, Riyadh 11543, Saudi Arabia*

Correspondence should be addressed to Mourad Ykhlef; ykhlef@ksu.edu.sa

Received 3 November 2014; Revised 1 March 2015; Accepted 1 March 2015

Academic Editor: Pandian Vasant

Solving winner determination problem in multiunit double auction has become an important E-business task. The main issue in double auction is to improve the reward in order to match the ideal prices and quantity and make the best profit for sellers and buyers according to their bids and predefined quantities. There are many algorithms introduced for solving winner in multiunit double auction. Conventional algorithms can find the optimal solution but they take a long time, particularly when they are applied to large dataset. Nowadays, some evolutionary algorithms, such as particle swarm optimization and genetic algorithm, were proposed and have been applied. In order to improve the speed of evolutionary algorithms convergence, we will propose a new kind of hybrid evolutionary algorithm that combines genetic algorithm (GA) with particle swarm optimization (PSO) to solve winner determination problem in multiunit double auction; we will refer to this algorithm as AUC-GAPSO.

1. Introduction

Internet auctions appeared on the scene in the mid-1990s and quickly became one of the most successful applications of electronic commerce [1]. Auction is defined as a market mechanism for accepting bids or offers from buyers or sellers and then used a set of rules to allocate goods. Double auction includes multiple sellers and multiple buyers in the same market where they are competing against each other for a variety of different items [2, 3]. In a multiunit double auction, each participant should determine how many units of an item and in which price he/she asks or bids. When a buyer's bid exceeds or matches a seller's ask, a trade will occur [4]. The most important issue in a double auction is how to allocate or match buyer with seller to maximize profit in the market. This problem is known as winner determination problem [5].

There are two approaches to solve this problem: exact approach and evolutionary approach; the exact approach, such as linear programming or simplex method, gives the optimal solution but in exponential time while evolutionary approach, such as genetic algorithm (GA) [2, 6] and particle swarm optimization (PSO) [3], solves the problem in a short time without the guarantee of obtaining always the optimal solution. In [7] a parallel genetic algorithm hybridization with local search was proposed. Other metaheuristics have been also used in double auction like memetic algorithm, Tabu search; in [8] "a computational experience regarding four well-known metaheuristics (stochastic local search, Tabu search, genetic algorithms, and memetic algorithms) for solving the winner determination problem" has been conducted.

Genetic algorithm is a general purpose search algorithm which uses principles inspired by natural genetic populations [9]. Particle swarm optimization (PSO) can be considered as an alternative to the standard GAs. The PSO was inspired by insect swarms and has been shown to be a competitor to the GA for optimization problems. Since then, several improved PSO general purpose algorithms have been developed [10, 11]. Both GA and PSO algorithms have shown good performance for some particular applications but not for other ones. For example, sometimes GAs outperformed PSO, but occasionally the opposite happened showing the typical application

driven characteristic of any single technique. This is due to the different search method adopted by the two algorithms, the typical selection-crossover-mutation approach versus the velocity-position-global-local best communication [12–15].

In order to improve the speed of convergence of existing multievolutionary algorithms, we will propose a new kind of hybrid evolutionary algorithm AUC-GAPSO that consists of combination of GA and PSO to solve winner determination problem in a multiunit double auction. AUC-GAPSO focuses on decreasing the time with good solution.

The rest of the paper is organized as follows. Double auction is defined in Section 2; however genetic algorithm and particle swarm optimization are defined in Section 3. Solving winner determination problem in double auction using our AUC-GAPSO algorithm is detailed in Section 4. Experimental results are reported in Section 5. We conclude this work in Section 6.

2. Double Auction

Double auction is referring to multiple buyers and sellers competing among each other. The double auction is divided into single-unit double auction and multiunit double auction [16]. In multiunit double auctions there are many sellers' bids prices and many buyers' bids for a variety of items with multiunit. Trade occurs when the buyers bid is greater than the sellers bids [2, 3, 17, 18]. Both the seller and the buyer have to determine the price and quantity of each specific item as follows:

$$\begin{aligned} &\text{Buyer} \longrightarrow \text{item, price, and quantity} \\ &\text{Seller} \longrightarrow \text{item, price, and quantity.} \end{aligned} \tag{1}$$

The auctioneer assigns buyers to sellers based on their preferences; there are many sellers asks that may satisfy some buyers' bids and vice versa. There are two kinds of double auctions: synchronous double auction (SDA) and asynchronous double auction (ASDA). Discrete or continuous time of the trading process is the main difference between SDA and ASDA. In the continuous double auction when the auctioneer finds compatible bids, it will match buyers and sellers and clear the market. On the contrary, the auctioneer in the synchronous auction gets all bids in a preset period of time and then makes the match and clears the market [2]. The factors that play roles in winners' determination can be listed as follows [19].

(i) Aggregation: the market maker role in auction is to dissemble or reassemble boundless of items. The options include buy-side, sell-side aggregation or it may include both sides' aggregation. However, if aggregation is not permitted, each bid and ask needs to be matched separately.

(ii) Divisibility: a divisible bid means that bidders can accept a trade of part of the quantity that they have specified at bidding while the indivisible bid means that bidders will reject the bid if the quantity is less than what they bid for [20].

(iii) There are homogeneous/heterogeneous goods.

This paper focuses on two features of the auction which are price and quantity; we will not include other performance factors such as quality, warranty, shipping time, and cost. We are particularly interested in solving winner determination problem in a multiunit synchronous double auction using a hybridization of genetic algorithm and particle swarm optimization.

3. Genetic Algorithm and Particle Swarm Optimization

Genetic algorithm was inspired by Darwin's theory about evolution [9]. Genetic algorithm can be defined as a particular class of evolutionary algorithms that uses techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover. Genetic algorithm starts with a set of random solutions represented as chromosomes which are called population. The term chromosome usually refers to a candidate solution for a problem that is often encoded as a bit string. The "gene" is a part of chromosome. Crossover typically consists of exchanging genes between two single chromosomes (parents). Mutation is usually flipping the bit at randomly chosen genes. Forming a new population is driven by the previous populations using crossover and mutation. Hoping that, the new population will be better than the old one. Solutions which are selected to form new solutions (offspring) are selected according to their fitness; chances of reproducing are linked by their suitability. This is repeated until stop condition (e.g., number of populations or improvement of the best solution) is reached. Mostly, the genetic algorithm requires a fitness function which assigns a score to each chromosome in the current population [9].

Particle swarm optimization (PSO) was originally proposed by Kennedy and Eberhart in 1995 [21]. Social behavior and movement dynamics of insects, birds, and fish were the main inspiration. The PSO algorithm is derived from a simplified social model that is closely linked to the swarming theory [22]. In PSO, individual particles of a swarm represent potential solutions. Each particle moves through the problem search space looking for an optimal or acceptable solution. The particles current positions are broadcasted to their neighboring particles. Velocity and the difference between its current positions (the best position found by its neighbors and the best position it has found so far) are the controllers of updating the position of each particle [23].

A good comparison between PSO and GA can be found in [14] where the authors proved that the difference in computational cost between PSO and the GA is problem dependent. The authors of [14] also conclude that PSO outperforms the GA with a greater degree of difference in computational cost when used to solve unconstrained nonlinear problems with continuous variables and smaller when applied to constrained nonlinear problems with continuous or discrete variables. GA has been widely applied to solve complex optimization problems because it can control both discrete and continuous variables, in addition to nonlinear objective and constraint functions [24]. In contrast, PSO may face a problem with a constrained optimization problem [11].

However, both GA and PSO have strengths and weaknesses; hybridization of GA and PSO could lead to further advances that have been proved by [11–13, 15, 25]. In addition, the hybridization could be achieved by different ways or structure.

In this paper, a hybridization of GA with PSO, called AUC-GAPSO, is proposed to overcome the limitations of GA and PSO and to combine their advantages. The convergence in GA has been controlled through crossover and mutation rates. In fact the decrease of inertia weight increases the swarm's convergence. The main problem with PSO is that it prematurely converges [10] to stable point which can be avoided through the genetic operator (crossover and mutation).

4. Solving Winner Determination Problem in Double Auction Using AUC-GAPSO Algorithm

In this section, we describe the AUC-GAPSO algorithm for solving winner determination problem in multiunit double auction. We first explain how we represented the auction dataset in our formalism, and then we present our chromosome structure, encoding scheme, and particles. After that, we describe genetic operators and define the utility or fitness assignment and selection criteria. Finally, we give the algorithmic structure and the pseudocode of AUC-GAPSO algorithm.

4.1. Dataset. Dataset contains data which AUC-GAPSO algorithm is applied to, in order to match bidders. The sellers/buyers information contains bidder ID, item, max quantity, maximum price that the buyers are willing to pay or the minimum price that the sellers are willing to accept, and sensitivity to price and quantity. Table 1 gives a sample of all possible matches of buyers and sellers; however Table 2 gives possible matches' matrix that can be derived from Table 1.

4.2. Chromosome. The chromosome in AUC-GAPSO represents all the winners with price and quantity of their winning bids. Also, in the chromosome we should match sellers with buyers. The size of chromosome is determined based on number of sellers, buyers, and items. Consequently, number of possible matches should be calculated to be equal to number of buyers multiplied by number of sellers for each item they bid/ask for.

There are many types of encoding that we can choose based on the problem such as binary encodings [26], which are the most common type; many characters; real-valued encodings [20]; and integer encoding [9]. Chromosomes representation became an issue for the different auction problems based on the auction type. There are many alternatives to represent a chromosome based on other problem domains. To decide which one is the best representation to solve double auction problem, Goldberg in his 1989 textbook recommends that "*the user should select a coding so that short, low-order schemata are relevant to the underlying problem and relatively unrelated to schemata over other fixed positions*" and "*the user should select the smallest alphabet that permits a natural expression of the problem*" [27]. The main meaning behind these recommendations is that the appropriate choice of genetic representation depends on the problem. According to that, in our problem, we will choose real-valued representation because it is the suitable one for constrained optimization problem. The sample of Tables 1 and 2 is represented by the chromosome of Table 3 where each couple of genes (price P and quantity Q) belongs to one match of the possible matches' matrix (Table 2); for example, buyer 1 and seller 1 won item 1 for price 13 and quantity 6.

Moreover, it is important to obtain sellers asks and buyers bids. In order to have less space, that information in addition to item vector will be kept as datasets.

4.3. Particles. After representing the chromosomes, each chromosome has to be linked with a particle that includes PSO variables related to position and velocity. Chromosomes can update their position and velocity using

$$V_{i+1} = \left[(w * V_i) + (c_1 * r_1) \otimes (p_i - x_i) + (c_2 * r_2) \otimes \left(p_g - x_i\right)\right], \quad (2)$$

$$x_{i+1} = x_i + V_{i+1}. \quad (3)$$

Table 1: Bidders information.

Bidder	Item	Quantity	Price	Price sensitivity	Quantity sensitivity
b1	1	10	14	2	1
b2	2	3	15	2	2
b3	1	5	12	1	1
s1	1	9	10	2	1
s2	2	3	14	2	1
s3	1	6	11	2	1

Table 2: Possible matches matrix.

Buyer	Seller	Item
1	1	1
1	3	1
2	2	2
3	1	1
3	3	1

Table 3: Chromosome.

P	Q	P	Q	P	Q	P	Q	P	Q
13	6	12	4	15	3	10	3	11	2

Particles local and global best position can be updated by using

$$p_{i+1} = \begin{cases} p_i & \text{if } p_i \geq x_{i+1} \\ x_{i+1} & \text{if } p_i < x_{i+1}, \end{cases} \quad (4)$$

$$p_g = \begin{cases} p_g & \text{if } p_g \geq p_i \\ p_i & \text{if } p_g < p_i, \end{cases} \quad (5)$$

where $i = [1 \cdots \text{MaxPopulation}]$ and w is called inertia weight which controls the influence of previous velocity v_i on the new velocity v_{i+1}. Variable v_i denotes the velocity of the ith particle in the swarm, x_i denotes particle position, p_i denotes the personal best position, p_g denotes the best position found by particles in its neighborhood, c_1 and c_2 are acceleration coefficients, and r_1; r_2 are random numbers, uniformly distributed in $[0 \cdots 1]$ which are used to maintain the diversity of the population. The symbol $\otimes$ denotes pointwise vector multiplication. The population size refers to number of particles in the iterations. As in the genetic algorithm, the initial populations are generated randomly. Eberhart and Shi proved that the performance of the PSO is not sensitive to the population size [12]. The maximum generations refer to the maximum number of generations allowed for the utility value to converge with the optimal solution [28]. Acceleration coefficients c_1 and c_2 [23, 29, 30] are two positive numbers such that $c_1 + c_2 = 4$; c_1 is a self-confidence factor to determine the relative influence of the cognitive component and c_2 is a swarm confidence factor used to determine the relative influence of the social component.

The PSO algorithm used a constant value for the inertia weight w; it could be equal to 0.9 as proposed by [30]. This inertia weight could also linearly decrease with respect to time [23, 31, 32]. Generally for initial stages of the search process, large inertia weight to enhance the global exploration is recommended while, for last stages, the inertia weight is reduced for local exploration [25, 28, 33]. The mathematical expression could be as follows:

$$\text{Inertia weight } w = w_{\max} - \left(\frac{w_{\max} - w_{\min}}{\text{MaxIter}} \right) + \text{Iter}, \quad (6)$$

where $w_{\max}$ is an initial value of the inertia weight, $w_{\min}$ is a final value of the inertia weight, Iter is a current iteration, and MaxIter is the maximum number of allowable iterations. The values of $w_{\max}$ and $w_{\min}$ that we have been using in our algorithm are 0.9 and 0.4, respectively [23]. We will define the inertia weight w to linearly decrease from 0.9 to 0.4. During the first 5,000 iterations of our algorithm w will be decreased by 0.0001 and then stay constant at 0.4 for the next iterations.

4.4. Genetic Operators. Genetic algorithm uses genetic operators to generate the offspring of the existing population. In this section, we describe three operators of genetic algorithms that we used in our algorithm: selection, crossover, and mutation.

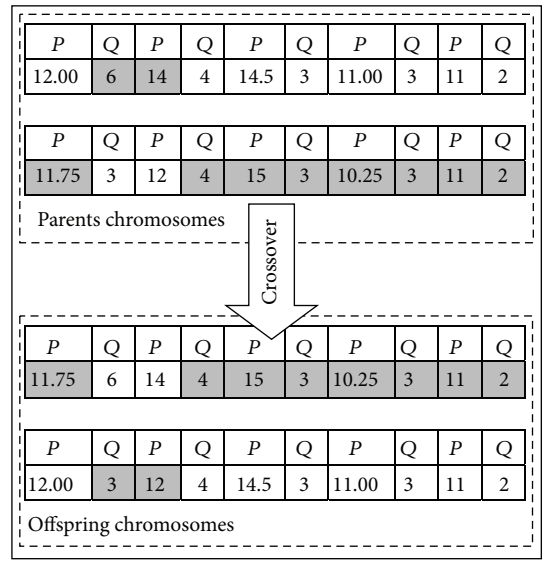


Figure 1: Crossover operator.

4.4.1. Selection. Variety of techniques can be used by a genetic algorithm to select the chromosomes to be copied into the new population. In AUC-GAPSO algorithm, we used "Elitism," where the fittest members of each generation are copied into the next generation.

4.4.2. Crossover. Crossover is working to combine two or more solutions to create new one (i.e., offspring). It is used in a way to come up with possibly better solutions. In most crossover operators, selecting chromosome for crossing over is based on a probability known as crossover probability. In AUC-GAPSO algorithm, we have used uniform crossover operation; crossover can be applied as shown in Figure 1.

4.4.3. Mutation. Mutation is the simplest operator but it is rarely happened. In mutation one or more chromosome genes values are altered randomly. The mutation is used to increase the variability of the population. Also, it ensures that every point in the search space will be reached. There are many mutation operators depending on the problem and the representation type. One of the most common mutations for the binary string is the bit-flip mutation. In contrast, in real-value encoding, the mutation is done by adding or subtracting a small number to the selected genes. The AUC-GAPSO mutation can be applied as illustrated in Figure 2.

4.5. Utility Function. The utility (or fitness according to GA literature) function is used to measure the performance of the chromosomes. In double auction the trade price alone is not ideal or makes the most profit for both sellers and buyers. Here we have the problem where we have to improve the reward of the double auction to match the ideal price and quantity and make the best profit for sellers and buyers according to their bids and predefined quantities. So how can we improve the reward for the auction bidders? It can be done through utility mechanism by improving how much

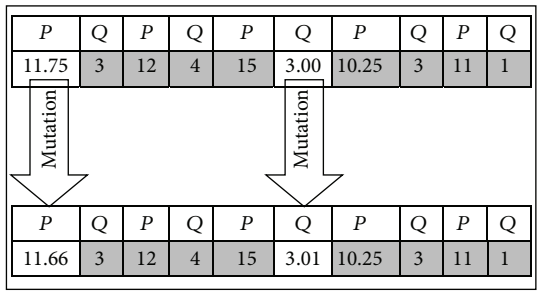


Figure 2: Mutation operator.

bidders can gain in the auction process. The highest utility of the seller when it matches with the highest bid price of buyers, however the highest utility for buyers is reached when he get matched with lowest bid price of sellers. So, to design an efficient auction the utility should be maximized [2]. The most suitable utility function is to maximize sellers and buyers utility function.

The utility function that can be solved as a linear programming problem to find the optimal solution was proposed by [34] where the authors represented the utility as the following linear equation:

$$\begin{aligned}\text{Max} &\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{a=1}^{k}\left(P_{i\cdot j,a}-P_{i,a}^{\text{WILL}}\right)\cdot Q_{i\cdot j,a}\\ &+\sum_{j=1}^{n}\sum_{i=1}^{m}\sum_{a=1}^{k}\left(P_{j,a}^{\text{WILL}}-P_{i\cdot j,a}\right)\cdot Q_{i\cdot j,a}.\end{aligned}\tag{7}$$

Consider the following constraints:

$$\begin{aligned}&\sum_{i=1}^{m}Q_{i,a}-\sum_{j=1}^{n}Q_{j,a}=0\quad \forall i\in S,\ \forall j\in B,\ \forall a\in\{1,\ldots,k\},\\ &P_{i\cdot j,a}\geq P_{i,a}^{\text{WILL}}\quad \forall i\in S,\ \forall j\in B,\ \forall a\in\{1,\ldots,k\},\\ &P_{i\cdot j,a}\leq P_{j,a}^{\text{WILL}}\quad \forall i\in S,\ \forall j\in B,\ \forall a\in\{1,\ldots,k\},\\ &0\leq Q_{i,a}\leq Q_{i,a}^{\text{MAX}}\quad \forall i\in(S\cup B),\ \forall a\in\{1,\ldots,k\},\end{aligned}\tag{8}$$

where m is the total number of sellers, n is the total number of buyers, k represents total number of items, $P_{i\cdot j,a}$ is the price of item a traded between seller i and buyer j, $Q_{i\cdot j,a}$ is the quantity of item a traded between seller i and buyer j, $P_{i,a}^{\text{WILL}}$ is seller i's willingness to receive (the minimum acceptable selling price) for item a, $P_{j,a}^{\text{WILL}}$ is buyer i's willingness to pay (the maximum acceptable buying price) for item a, S is set of sellers, in which there are $|S|=m$ sellers, and B is the set of buyers, in which there are $|B|=n$ buyers.

We will use utility function of [2] where each bidder has his own utility function; therefore, he can specify his sensitivity or the range of acceptance of each feature (price and quantity). This utility function is shown as in

$$\begin{aligned}\text{Max} &\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{a=1}^{k}\left(f_1^i\left(P_{i\cdot j,a}\right)-P_{i,a}^{\text{WILL}}\right)\cdot f_2^i\left(Q_{i\cdot j,a}\right)\\ &+\sum_{j=1}^{n}\sum_{i=1}^{m}\sum_{a=1}^{k}\left(P_{j,a}^{\text{WILL}}-f_1^i\left(P_{i\cdot j,a}\right)\right)\cdot f_2^i\left(Q_{i\cdot j,a}\right)\\ &\forall i\in S,\quad \forall j\in B,\quad \forall a\in\{1,\ldots,k\}.\end{aligned}\tag{9}$$

We can see that the price and the quantity have been replaced with functions. The authors of [2] suggest two types of function the early-growth style and the late-growth style based on buyer/seller choice; early-growth style is concave downward. In contrast, the late-growth style is concave upward. In early-growth style the bidder has most influence in changing price or quantity. In the following $Q_{i,a}^{\text{MAX}}$ means trader i's maximum acceptable quantity for item a; t means the level of steepness ($t=1,2,3,4,5$).

Early-growth equations are as follows:

price:

$$\begin{aligned}&f_1^i\left(P_{i\cdot j,a}\right)\\ &=\left\{-\left(\frac{P_{i\cdot j,a}-P_{i,a}^{\text{WILL}}}{P_{j,a}^{\text{WILL}}-P_{i,a}^{\text{WILL}}}-1\right)^{2t}+1\right\}\\ &\cdot\left(P_{j,a}^{\text{WILL}}-P_{i,a}^{\text{WILL}}\right)+P_{i,a}^{\text{WILL}}\quad \forall j\in B,\ \forall a\in\{1,\ldots,k\};\end{aligned}\tag{10}$$

seller quantity:

$$f_2^i\left(Q_{i\cdot j,a}\right)=\left\{-\left(\frac{Q_{i\cdot j,a}}{Q_{i,a}^{\text{MAX}}}-1\right)^{2t}+1\right\}\cdot Q_{i,a}^{\text{MAX}}\tag{11}$$
$$\forall i\in S,\quad \forall j\in B,\quad \forall a\in\{1,\ldots,k\};$$

buyer quantity:

$$f_2^i\left(Q_{i\cdot j,a}\right)=\left\{-\left(\frac{Q_{i\cdot j,a}}{Q_{j,a}^{\text{MAX}}}-1\right)^{2t}+1\right\}\cdot Q_{j,a}^{\text{MAX}}\tag{12}$$
$$\forall i\in S,\quad \forall j\in B,\quad \forall a\in\{1,\ldots,k\}.$$

Late-growth equations are as follows:

price:

$$f_1^i\left(P_{i\cdot j,a}\right)\left(\frac{P_{i\cdot j,a}-P_{i,a}^{\text{WILL}}}{P_{j,a}^{\text{WILL}}-P_{i,a}^{\text{WILL}}}-1\right)^{2t}\cdot\left(P_{j,a}^{\text{WILL}}-P_{i,a}^{\text{WILL}}\right)+P_{i,a}^{\text{WILL}}$$
$$\forall j\in B,\quad \forall a\in\{1,\ldots,k\};\tag{13}$$

seller quantity:

$$f_2^i\left(Q_{i\cdot j,a}\right) = \left(\frac{Q_{i\cdot j,a}}{Q_{i,a}^{\mathrm{MAX}}} - 1\right)^{2t} \cdot Q_{i,a}^{\mathrm{MAX}} \quad \forall i \in S, \quad \forall j \in B, \quad \forall a \in \{1, \ldots, k\}; \tag{14}$$

buyer quantity:

$$f_2^i\left(Q_{i\cdot j,a}\right) = \left(\frac{Q_{i\cdot j,a}}{Q_{j,a}^{\mathrm{MAX}}} - 1\right)^{2t} \cdot Q_{j,a}^{\mathrm{MAX}} \quad \forall i \in S, \quad \forall j \in B, \quad \forall a \in \{1, \ldots, k\}. \tag{15}$$

The advantage of [2] utility is that the participant has the flexibility to change his preferences in each round while in [34] utility the participant has identical utility. The authors of [3] suggested a fuzzy utility function.

4.6. Termination. AUC-GAPSO operations applied for all parents in all iterations. At the end of an operation, new chromosomes (offspring) generated as a result of mutation and crossover operators. These chromosomes should be evaluated by using utility or fitness function. Therefore, AUC-GAPSO algorithm obtained the best N chromosome, where N is the size of population in each generation. This process is repeated until a termination condition is reached. There are many termination conditions that can be used. In AUC-GAPSO the condition is either the maximum generation number (G) is reached or the global best (Pg) is not improved for a specific number of times. At the end, the best chromosome of the last iteration is considered as the fittest matching.

4.7. AUC-GAPSO Algorithm to Solve Winner Determination Problem in Multiunit Double Auction. In this section we present the algorithmic structure of AUC-GAPSO that we produced in addition to the description of its pseudocode. The flowchart of AUC-GAPSO algorithm is shown in Figure 3. After finding the possible matches, the algorithm starts by initializing the population. Then, its fitness or utility value is determined for each chromosome. Subsequently, the following processes are repeated until the prespecified maximum number of generations is achieved or the global best is not improved for a predetermined period of times.

The pseudocode of AUC-GAPSO is shown in Algorithm 1. First, population size, maximum number of generations, and buyers and sellers bidding are taken as input. Then AUC-GAPSO algorithm works as follows.

(1) Loop counter and counter of global best improvement are set to zero.

(2) Find the possible matches from the bidding.

(3) Initial population is generated.

(4) Calculate the fitness (utility) for each chromosome using formula (9) and particle variables (velocity and position are set to zero while local and global best positions are set using formulas (4) and (5), resp.).

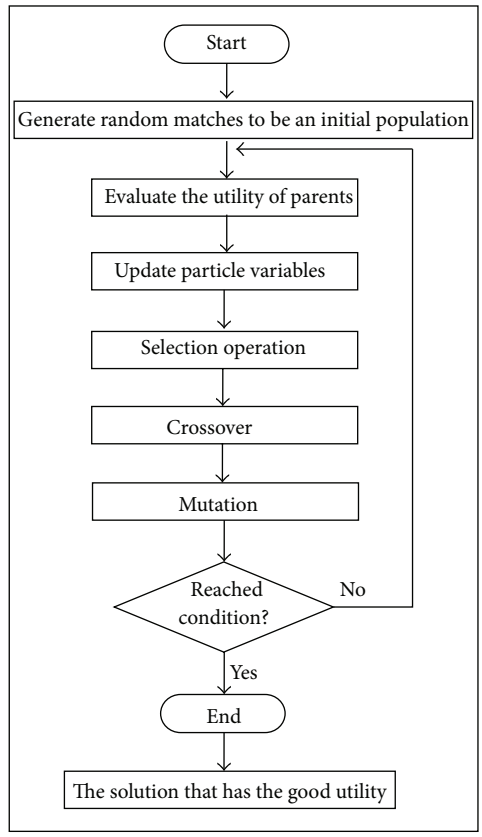


Figure 3: AUC-GAPSO flowchart.

(5) Preserve global best, to be compared to the new one, in order to decide whether it is improved or not.

(6) Selection operation is conducted to find best chromosome in population P and copy the best chromosome in the new population Pn.

(7) Population P is mutated and crossed-over based on the probability of crossover and mutation and then put into Pn.

(8) Calculate Pn fitness and particle variables value using formulas (9) and (2), (3), (4), and (5), respectively.

(9) Update CountPg according to the value of Pg. If the global best did not improve after the selection, between original P and Pn, then the counter will be increased. This step takes advantage of PSO in order to reduce the time.

(10) Increase the loop counter.

(11) Check the termination condition. If the maximum generation number is reached or when the global best is not improved for a specific number of times (i.e., CountPg variable equals quarter maximum generation number), then stop the algorithm and put P into W, which have the fittest matches conducted from the algorithm. Otherwise, go to step (4).

Input:
population size: N
Maximum number of generations: G
Buyers input: B
Sellers input: S
Output:
Interesting Winners: W
Algorithm:
(1) Initialize $t = 0$ and CountPg = 0
(2) Find possible matches from B and S
(3) Generate population P of size N
(4) For each chromosome $i \in P$
  (4.1) Generate winners i from chromosome i
  (4.2) Calculate fitness of winners i
  (4.3) Initialize $X0$, $V0$, Pi, Pg
(5) OldPg = Pg
(6) For each chromosome i and $i + 1 \in P$
  (6.1) Select the best chromosomes to be copied into Pn
  (6.2) Mutate and crossover chromosome i with $i + 1$
  (6.3) Add reproduced chromosome to Pn
  (6.4) Generate winners i from reproduced chromosome i
  (6.5) Calculate fitness of winners i
  (6.6) Update Vi, Xi, Pi, Pg
(7) If OldPg = Pg then
    CountPg ++
  Else
    CountPg = 0
(8) t++
(9) If (($t > G$) or (CountPg > (G/4))) then
    get the fittest chromosome C from P
    $W = C$
    Stop algorithm
  Else
    Go to Step (4)

Algorithm 1: AUC-GAPSO algorithm.

Table 4: AUC-GAPSO output.

Buyer	Seller	Item	Price	Quantity	Fitness
1	1	1	11.08	9	
1	3	1	11.76	1	
2	2	2	15	1	69.4508
3	1	1	10.18	1	
3	3	1	11.02	4	

5. Experiments Results

The programming of AUC-GAPSO algorithm is done by using java Borland JBuilder environment on Intel Core 2 Duo 3 GHz computer with 4 GB RAM, running with Microsoft Windows 7. For controlling the parameters of GA part, the crossover and mutation rates are set at 70% and 30%, respectively. During the experiments, all the bidding, items, population size, and generation were given by the user. The output of the experiments is a file that includes the most suitable solution or matches (fittest chromosome), along with its fitness, for example, for the scenario where there are three buyers, three sellers, and two items (see Tables 1 and 2). The output fittest solution obtained by AUC-GAPSO is given in Table 4.

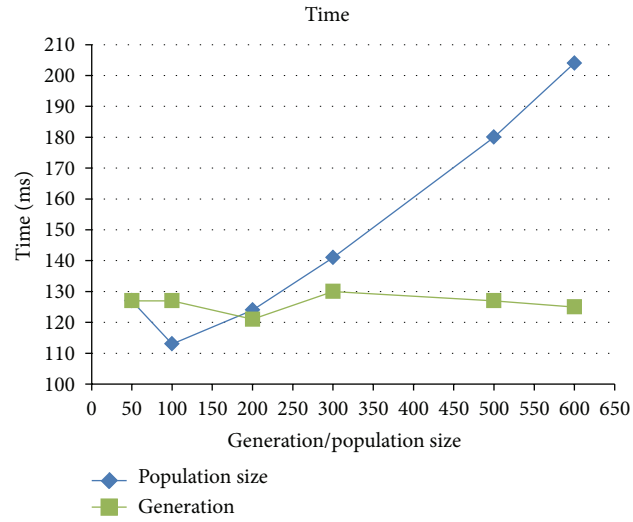


Figure 4: The effect on time by altering population/generation size.

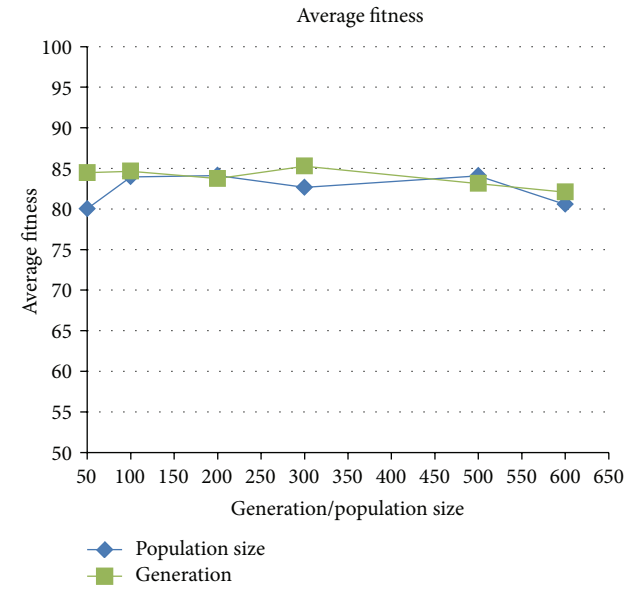


Figure 5: The effect on average fitness by altering population/generation size.

We conducted two experiments for AUC-GAPSO in order to test the speed and fitness or utility. First experiment set the population to 200 and the generation of [50$\cdots$600] while the second experiment set the generation to 200 and the population of [50$\cdots$600]. Referring to the result of AUC-GAPSO of Figure 4, it can be observed that AUC-GAPSO spends less time to be completed. In addition, in first experiment of AUC-GAPSO, increasing of the generation will not necessarily affect the increase of the spent time because its termination condition does not only depend on the generation (iteration), but also depends on the improvement

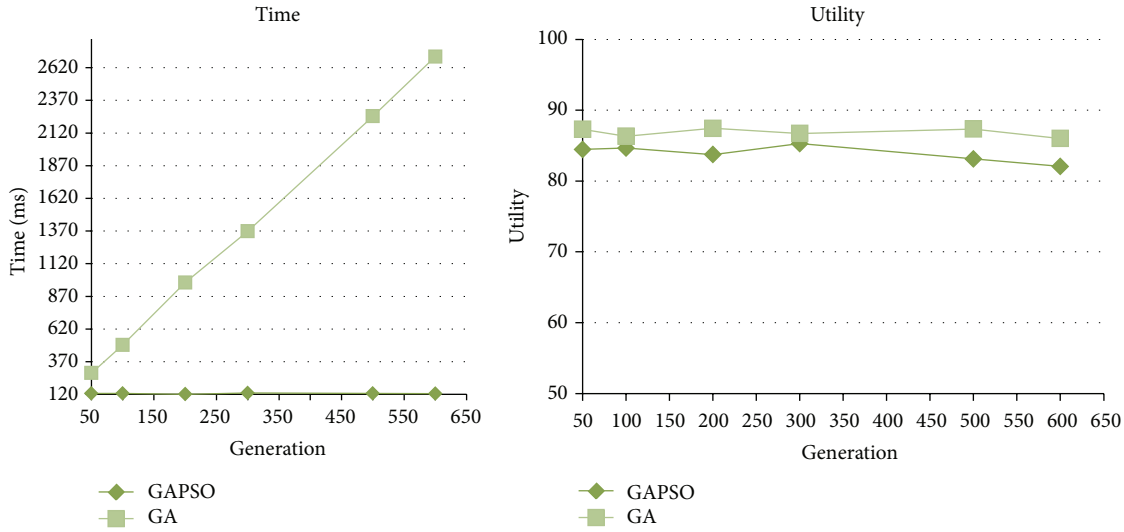


Figure 6: Comparison between GA and GAPSO on effect of generation.

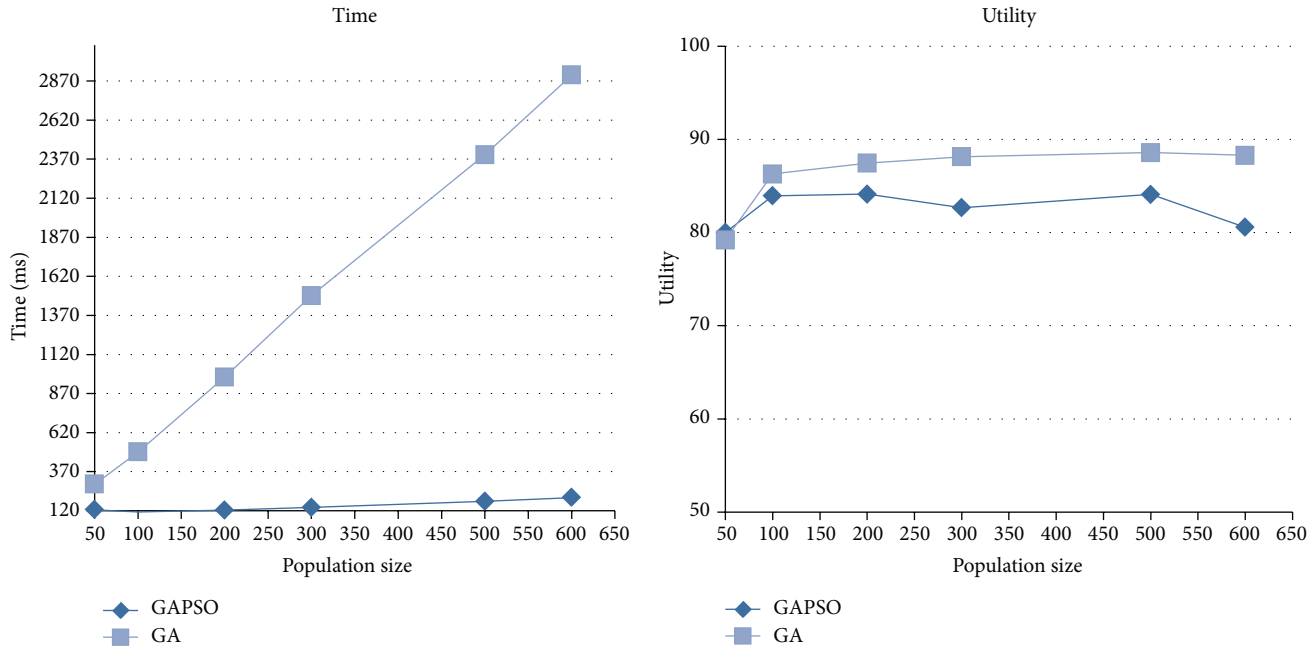


Figure 7: Comparison between GA and GAPSO on effect of population size.

cycle of the global best. On the other hand, in the second experiment (see Figure 4), increasing the population size will most likely increase the execution time; AUC-GAPSO took more time to apply the GA operators and calculate the PSO variables of each chromosome in the population. Figure 5 shows that increasing in the population or generation will not guarantee an improvement of the average fitness, and it could even make it worse.

5.1. AUC-GA and AUC-GAPSO Comparison. For comparing AUC-GAPSO to only genetic based solution AUC-GA, we conducted two experiments for AUC-GA and two experiments for AUC-GAPSO. First experiment set the population to 200 and the generation of [50$\cdots$600] while the second experiment set the generation to 200 and the population of [50$\cdots$600]. Figure 6 shows the result of the first experiment, in both AUC-GA and AUC-GAPSO, related to increasing the generations. Figure 7 shows the result of the second experiment in both AUC-GAPSO and AUC-GA, related to increasing the population size. By comparing the graphs in Figures 6 and 7, we can observe that AUC-GAPSO has better performance than AUC-GA, especially in the time spent to extract the fittest matches. Therefore, we can conclude that it is better to use AUC-GAPSO rather than AUC-GA.

We can also see from Figures 6 and 7 that increasing the population and generation will not guarantee an improvement of the fitness; sometimes it even makes it worse. Moreover, even if AUC-GAPSO reduces the time, it is important to choose the right generation and population size, which means that we have to be careful in choosing the generation and population size. We think that it is better to use AUC-GAPSO with high population size and moderate generation to extract winners. This trade-off decision will prospectively guarantee good fitness in short time.

6. Conclusion

In this paper, we proposed a hybrid evolutionary algorithm AUC-GAPSO, which combined genetic and particle swarm optimization, to solve the winner determination problem in multiunit double auction. The AUC-GAPSO algorithm used the property of evolutionary algorithms that matches bidders in a short time. The proposed solution is focusing on maximizing all traders' utility or fitness through optimization based on hybridization of GA operations and PSO parameters. The algorithmic components, including real-coded representation, selection, crossover, mutation operators, and particles variables, all contribute to achieving a runtime improvement. Experimental results demonstrated that AUC-GAPSO in double auction incorporates high performance concerning the time of execution. In addition, AUC-GAPSO reduced the time when compared to AUC-GA. Moreover, we exposed how it is important to be careful in choosing the right generation and population size. In the future, one can solve multi-item (combinatorial) multiunit double auction which can be achieved by introducing multiobjective evolutionary algorithm.

Conflict of Interests

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

Acknowledgments

This work was supported by Deanship of Scientific Research and Research Center of College of Computer and Information Sciences, King Saud University. The authors are grateful for this support.

References

[1] P. R. Wurman, "Online auction site management," in *The Internet Encyclopedia*, B. Hossein, Ed., John Wiley & Sons, New York, NY, USA, 2004.

[2] J. H. Choi, H. Ahn, and I. Han, "Utility-based double auction mechanism using genetic algorithms," *Expert Systems with Applications*, vol. 34, no. 1, pp. 150–158, 2008.

[3] M. A. Rigi, S. Mohammadi, and M. Delgir, "Designing fuzzy utility-based double auctions using particle swarm optimization algorithm," in *Proceedings of the International Conference on Innovations in Information Technology (IIT '09)*, pp. 110–114, Al Ain, United Arab Emirates, December 2009.

[4] S. Gjerstad and J. Dickhaut, "Price formation in double auctions," *Games and Economic Behavior*, vol. 22, no. 1, pp. 1–29, 1998.

[5] J. E. Teich, H. Wallenius, J. Wallenius, and O. Koppius, "Emerging multiple issue E-auctions," Erasmus Research Institute of Management Report, 2003.

[6] M. Ykhlef and R. Alqifari, "Genetic algorithm based optimization for E-auction," in *Proceedings of the IADIS International Conference on E-Commerce*, Rome, Italy, 2011.

[7] D. Boughaci, L. Slaouti, and K. Achour, "A hybrid parallel genetic algorithm for the winner determination problem in combinatorial auctions," in *Proceedings of the 31st SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence (AI '11)*, pp. 65–78, Springer, December 2011.

[8] D. Boughaci, "Metaheuristic approaches for the winner determination problem in combinatorial auction," in *Artificial Intelligence, Evolutionary Computing and Metaheuristics: In the Footsteps of Alan Turing* , vol. 427, pp. 775–791, Springer, Berlin, Germany, 2013.

[9] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Boston, Mass, USA, 1998.

[10] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to participle swam optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.

[11] A. Banks, J. Vincent, and C. Anyakoha, "A review of particle swarm optimization. Part II. Hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications," *Natural Computing*, vol. 7, no. 1, pp. 109–124, 2008.

[12] R. C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in *Proceedings of the 7th International Conference on Evolutionary Programming (EP '98)*, pp. 611–616, 1998.

[13] M. Settles and T. Soule, "A hybrid GA/PSO to evolve artificial recurrent neural networks," in *Intelligent Engineering Systems Through Aritificial Neural Networks*, vol. 3, pp. 51–56, 2003.

[14] R. Hassan, B. Cohanim, O. de Weck, and G. Venter, "A comparison of particle swarm optimization and the genetic algorithm," in *Proceedings of the Structural Dynamics and Materials Conference*, Austin, Tex, USA, 2005.

[15] M. Settles and T. Soule, "Breeding swarms: a GA/PSO hybrid," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '05)*, pp. 161–168, June 2005.

[16] X. Wang, M. Huang, Y. Ma, and Z. Tan, "A double auction method for resource management and bidding strategy on grid resources," in *Proceedings of the 4th International Conference on Genetic and Evolutionary Computing (ICGEC '10)*, pp. 422–425, December 2010.

[17] M. Bichler, J. Kalagnanam, H. Soo Lee, and J. Lee, "Winner determination algorithms for electronic auctions: a framework design," in *E-Commerce and Web Technologies: Third International Conference, EC-Web 2002 Aix-en-Provence, France, September 2–6, 2002 Proceedings*, vol. 2455, pp. 37–46, Springer, Berlin, Germany, 2002.

[18] P. Steve, S. Elizabeth, and P. Simon, "Using genetic programming to optimise pricing rules for a double auction market," in *Proceedings of the Workshop on Agents for Electronic Commerce*, Pittsburgh, Pa, USA, 2003.

[19] K. Jayant and D. C. Parkes, "Auctions, bidding and exchange design," in *Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era*, vol. 74 of *International Series*

in Operations Research & Management Science, pp. 143–212, Springer, Berlin, Germany, 2004.

[20] A. Movaghar and S. Khanpour, "Design and implementation of optimal winner determination problem in combinatorial e-auction," *World Academy of Science, Engineering and Technology*, vol. 20, pp. 57–60, 2006.

[21] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, Perth, Australia, December 1995.

[22] J. E. Onwunalu and L. J. Durlofsky, "Application of a particle swarm optimization algorithm for determining optimum well location and type," *Computational Geosciences*, vol. 14, no. 1, pp. 183–198, 2010.

[23] C. Blum and X. Li, *Swarm Intelligence in Optimization*, Natural Computing Series, Springer, Berlin, Germany, 2008.

[24] T. Zhang and B. W. Brorsen, "Particle swarm optimization algorithm for agent-based artificial markets," *Computational Economics*, vol. 34, no. 4, pp. 399–417, 2009.

[25] M. Sudhakaran, P. Ajay-D-Vimalraj, and T. G. Palanivelu, "GA and PSO culled hybrid technique for economic dispatch problem with prohibited operating zones," *Journal of Zhejiang University: Science A*, vol. 8, no. 6, pp. 896–903, 2007.

[26] T. Buer and G. Pankratz, "Solving a bi-objective winner determination problem in a transportation procurement auction," *Logistics Research*, vol. 2, no. 2, pp. 65–78, 2010.

[27] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, Reading, Mass, USA, 1989.

[28] P. Umapathy, C. Venkataseshaiah, and M. Arumugam, "Particle swarm optimization with various inertia weight variants for optimal power flow solution," *Discrete Dynamics in Nature and Society*, vol. 2010, Article ID 462145, 15 pages, 2010.

[29] K. Premalatha and A. M. Natarajan, "Hybrid PSO and GA for global maximization," *International Journal of Open Problems in Computer Science and Mathematics*, vol. 2, no. 4, pp. 597–608, 2009.

[30] F. Kui and N. Guihua, "Application of particle swarm optimization algorithm in e-commerce negotiation," in *Proceedings of the 6th Wuhan International Conference on E-Business*, pp. 225–231, 2007.

[31] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proceedings of the IEEE International Conference Evolutionary Computation (CEC '00)*, pp. 84–88, La Jolla, Calif, USA, July 2000.

[32] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Evolutionary Programming VII*, V. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, Eds., vol. 1447 of *Lecture Notes in Computer Science*, pp. 591–600, Springer, Berlin, Germany, 1998.

[33] K. Premalatha and A. M. Natarajan, "Hybrid PSO and GA models for document clustering," *International Journal of Advances in Soft Computing and its Applications*, vol. 2, no. 3, pp. 302–320, 2010.

[34] P. Huang, A. Scheller-Wolf, and K. Sycara, "Design of a multi-unit double auction e-market," *Computational Intelligence*, vol. 18, no. 4, pp. 596–617, 2002.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 587925, 11 pages
http://dx.doi.org/10.1155/2015/587925

Research Article

Intuitionistic Fuzzy Kernel Matching Pursuit Based on Particle Swarm Optimization for Target Recognition

Xiaodong Yu, Yingjie Lei, Shaohua Yue, and Feixiang Meng

Air Defense and Antimissile Institute, Air Force Engineering University, Xi'an 710051, China

Correspondence should be addressed to Yingjie Lei; leiyjie@163.com

Received 18 November 2014; Accepted 28 February 2015

Academic Editor: Gerhard-Wilhelm Weber

In order to overcome the long training time caused by searching optimal basic functions based on greedy strategy from a redundant basis function dictionary for the intuitionistic fuzzy kernel matching pursuit (IFKMP), the particle swarm optimization algorithm with powerful ability of global search and quick convergence rate is applied to speed up searching optimal basic function data in function dictionary. The approach of intuitionistic fuzzy kernel matching pursuit based on particle swarm optimization algorithm, namely, PS-IFKMP, is proposed. This algorithm is applied to the aerospace target recognition, which requires real-time ability. Simulation results show that, compared with the conventional approaches, the proposed algorithm can decrease training time and improve calculation efficiency obviously with almost unchanged classification accuracy, while the model has better sparsity and generalization. It is also demonstrated that this approach is suitable for the application requiring both accuracy and efficiency.

1. Introduction

The kernel matching pursuit (KMP) Classifier is a new classification technique proposed by Vincent and Bengio in 2002 [1]. The basic idea of KMP originates from matching pursuit (MP) method of sign processing theory, a greedy constructive algorithm that approximates a given function by a linear combination of basis functions chosen from a redundant basis function dictionary, and can be seen as a form of boosting. When training a KMP Classifier, there are mainly three steps. First, it maps training data into a redundant basis dictionary by a given kernel function and then uses greedy algorithm to find a linear expansion of basis functions that are selected from a redundant dictionary of functions to minimize the loss function and, at last, adjusts the corresponding coefficients by backfitting algorithm. The linear expansion of basis functions is the target function that we seek. The performance of the KMP Classifier is comparable to that of support vector machine (SVM) for classification, while typically requiring far fewer support points [2, 3]. Compared with the other Kernel methods, KMP is very flexible and has no constraint on the form of kernels. It can be allowed to mix several shapes of kernels in one instance. Up to now, the theory of KMP has been successfully applied to a variety of fields, such as pattern classification [4, 5], image recognition [6], and intrusion detection [7].

However, there is a special case in practical application; the importance degree (or threat degree) of one type is higher than the other targets, so we need to classify the appointed important target much more precisely and reduce the recognition accuracy of the other unimportant targets. For example, in missile defense battle, the requirement of recognition accuracy of the true warhead is higher than that of bait, debris, and so forth. The traditional kernel matching pursuit algorithm deals with all samples equally and the decision function gives a full scale consideration for all samples in order to minimize total recognition error.

Unfortunately, such algorithm cannot perform well in special cases. To circumvent the disadvantage, Li et al. [8] proposed a fuzzy kernel matching pursuit (FKMP) machine, which can classify the appointed important samples much more precisely according to the predefined importance of the data. However, this method sets the fuzzy factor on the basis of artificial experience, which may bring risks to the training process. Lei et al. [9] proposed intuitionistic fuzzy kernel matching pursuit (IFKMP) machine and expanded KMP algorithm into intuitionistic fuzzy field to solve the problem that some important data cannot be classified precisely by

assigning the intuitionistic fuzzy parameters to different samples efficiently. Nevertheless, the IFKMP machine essentially uses MP greedy algorithm to find a linear combination of basis functions that are selected from a redundant dictionary of functions, so the problem of greatly computer burden has not been solved, and the computing time can be increased greatly with searching spaces increasing, which may be more serious when the dictionary has two or few functions [10]. To overcome the above drawback, the optimization methods have been introduced into MP algorithm. An improved genetic algorithm (GA) is proposed to simplify the calculation of MP algorithm in [11]; however, due to GA's prematurity, the stable capability of learning machine is not good. In [12], a quantum genetic algorithm is introduced for modifying MP process; however, the computing time cannot reduce effectively due to the slow search speed of quantum genetic algorithm. In [13], a novel technique for KMP based on intuitionistic fuzzy c-means (IFCM) was proposed, but IFCM is still a local optimization algorithm and reduces the recognition accuracy of KMP. Particle swarm optimization (PSO) proposed by Kennedy and Eberhart [14] is a global optimization algorithm. Since its appearance, for its powerful ability of global search and quick convergence rate, PSO has received more and more attention [15, 16]. Based on analysis above, this paper put forward a novel technique for intuitionistic fuzzy kernel matching pursuit based on particle swarm optimization (PS-IFKMP), where the greedy algorithm is replaced by PSO algorithm. In order to test the effect and validity of PS-IFKMP algorithm, three different datasets are used for simulation. The results are compared with those based on KMP and IFKMP. Experiments results validate the preference and efficiency of PS-IFKMP.

2. Intuitionistic Fuzzy Kernel Matching Pursuit

2.1. Matching Pursuit. Matching pursuit was introduced in the signal-processing community as an algorithm that decomposes any signal into a linear expansion of waveforms that are selected from a redundant dictionary of functions.

Given l noisy observations $\{y_1, y_2, \ldots, y_l\}$ at $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_l\}$ and a finite dictionary $\mathbf{D}$ of functions in Hilbert space H, we aim to find sparse approximations of $\{y_1, y_2, \ldots, y_l\}$ that are expansion of the form $\mathbf{f}_N = \sum_{n=1}^{N} \alpha_n \mathbf{g}_n(\mathbf{x})$, which minimize the squared norm of the residue $\|\mathbf{R}_N\|^2 = \|\mathbf{y} - \mathbf{f}_N\|^2$, where $\{\alpha_1, \alpha_2, \ldots, \alpha_l\} \in \mathbf{R}^N$ and $\{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_l\} \in \mathbf{D}$.

The process of MP can be described as follows: it starts at stage 0 with $\mathbf{R}_0 = \mathbf{y}$ and $\mathbf{f}_0 = 0$ and recursively appends functions to an initially empty basis. Given f_n, we build $\mathbf{f}_{n+1} = \mathbf{f}_n + \alpha_{n+1}\mathbf{g}_{n+1}$, by searching for $\mathbf{g}_{n+1} \in \mathbf{D}$ and $\alpha_{n+1} \in \mathbf{R}$ that minimize the squared norm of the residue $\|\mathbf{R}_{N+1}\|^2 = \|\mathbf{R}_N - \alpha_{n+1}\mathbf{g}_{n+1}\|^2$,

$$\left(\mathbf{g}_{n+1}, \alpha_{n+1}\right) = \underset{(\mathbf{g}\in\mathbf{D}, \alpha\in\mathbf{R})}{\arg\min} \left\|\mathbf{R}_{N+1}\right\| = \left\| \left(\sum_{k=1}^{n} \alpha_k \mathbf{g}_k \right) + \alpha\mathbf{g} - \mathbf{y} \right\|^2. \tag{1}$$

Moreover,

$$\mathbf{g}_{n+1} = \underset{\mathbf{g}\in\mathbf{D}}{\arg\max} \left| \frac{\langle \mathbf{g}, \mathbf{R}_n \rangle}{\|\mathbf{g}\|} \right|, \qquad \alpha_{n+1} = \left| \frac{\langle \mathbf{g}_{n+1}, \mathbf{R}_n \rangle}{\|\mathbf{g}_{n+1}\|^2} \right|. \tag{2}$$

From above, we can see that MP actually adopts greedy algorithm. In each iterative process, MP finds the most correlative basis function in $\mathbf{D}$ with the current residue $\mathbf{R}_n$. As the basis is appended continuously, $\mathbf{f}_n$ can approach observations $\{y_1, y_2, \ldots, y_l\}$ more and more accurately. However, algorithm will generally stop when the norm of the residue $\|\mathbf{R}_n\|^2$ goes below a predefined threshold or the number of the bases reaches the maximum.

2.2. Kernel Matching Pursuit. Kernel matching pursuit is simply the idea of applying the matching pursuit family of algorithms to problem in machine learning, using a kernel-based dictionary [5]. Given a kernel function $K : \mathbf{R}^d \times \mathbf{R}^d \to \mathbf{R}$, we construct the basis dictionary of KMP on the training data: $\mathbf{D} = \{\mathbf{g}_i = K(\cdot, \mathbf{x}_i) \mid i = 1, \ldots, l\}$.

Kernel method is enlightened in great part to the success of the SVM, but there are some differences between them. The kernel used in SVM must satisfy the Mercer condition, while kernel in KMP has no such restriction. Simultaneously, one can use more than one kernel when constructing the function dictionary of KMP. There are some kernels in common use:

(1) polynomial kernel: $K(x, x_i) = [(x - x_i) + 1]^d$,

(2) radial basis function (RBF) kernel:

$$K\left(x, x_i\right) = \exp\left(-\frac{\left\|x - x_i\right\|^2}{2\rho} \right). \tag{3}$$

(3) Sigmoid kernel: $K(x, x_i) = S(v(x, x_i) + c)$.

2.3. Intuitionistic Fuzzy Kernel Matching Pursuit. By assigning the intuitionistic fuzzy parameters to different samples efficiently, IFKMP algorithm can classify the target samples of diverse importance much more precisely according to the predefined importance of samples.

Definition 1 ($\odot$ operation). Let $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$ and $\mathbf{y} = \{y_1, y_2, \ldots, y_m\}$ be two vectors; then we define the $\odot$ operation as

$$\mathbf{x} \odot \mathbf{y} = \left(x_1 \cdot y_1, \ldots, x_m \cdot y_m\right). \tag{4}$$

Moreover,

$$\left\|\mathbf{x} \odot \mathbf{y}\right\|^2 = \sum_{i=1}^{m} \left(x_i \cdot y_i\right)^2. \tag{5}$$

Given l noisy samples $\{(\mathbf{x}_1, y_1, \omega(y_1)), \ldots, (\mathbf{x}_l, y_l, \omega(y_l))\}$, where $\mathbf{x}_i \in \mathbf{R}^N$ represents the sample feature, $y_i \in \mathbf{R}$ represents the sample observation, and $\omega(y_i)$ is intuitionistic fuzzy

parameter and given a kernel function $K : \mathbf{R}^d \times \mathbf{R}^d \rightarrow \mathbf{R}$, we construct the basis dictionary on the training data: $\mathbf{D} = \{\mathbf{g}_i = K(\cdot, \mathbf{x}_i) \mid i = 1, \ldots, l\}$.

Then the residue is

$$\mathbf{r}_N = \boldsymbol{\omega} \odot (\mathbf{y} - \mathbf{f}_N) = \begin{bmatrix} \omega(y_1)(y_1 - f_N(\mathbf{x}_1)) \\ \cdots \\ \omega(y_l)(y_l - f_N(\mathbf{x}_l)) \end{bmatrix}, \quad (6)$$

where $f_N(\mathbf{x}_i) = \sum_{j=1}^{N} \alpha_j g_j(\mathbf{x}_i)$ represents the estimated value $\widehat{y}_i$ of the ith sample; then the squared norm of the residue can be written as

$$\|\mathbf{r}_N\|^2 = \|\boldsymbol{\omega} \odot (\mathbf{y} - \mathbf{f}_N)\|^2 = \sum_{i=1}^{l} (\omega(y_i)(y_i - f_N(\mathbf{x}_i)))^2. \quad (7)$$

According to MP algorithm, we can get

$$\begin{aligned} \|\mathbf{r}_{N+1}\|^2 &= \|\boldsymbol{\omega} \odot (\mathbf{y} - (\mathbf{f}_N + \alpha_{N+1}\mathbf{g}_{N+1}))\|^2 \\ &= \|\boldsymbol{\omega} \odot (\mathbf{y} - \mathbf{f}_N) - \boldsymbol{\omega} \odot (\alpha_{N+1}\mathbf{g}_{N+1})\|^2 \\ &= \|\mathbf{r}_N - \boldsymbol{\omega} \odot (\alpha_{N+1}\mathbf{g}_{N+1})\|^2 \\ &\stackrel{\text{def}}{=} \|\mathbf{r}_N - \boldsymbol{\omega} \odot (\alpha \mathbf{g})\|^2. \end{aligned} \quad (8)$$

Then,

$$\|\mathbf{r}_{N+1}\|^2 = \|\mathbf{r}_N\|^2 - 2\alpha \langle \mathbf{r}_N, \boldsymbol{\omega} \odot \mathbf{g} \rangle + a^2 \|\boldsymbol{\omega} \odot \mathbf{g}\|^2. \quad (9)$$

For any $\mathbf{g} \in \mathbf{D}, \alpha \in \mathbf{R}$ that minimizes the squared norm of the residue $\|\mathbf{r}_{N+1}\|^2$ is given by

$$\begin{aligned} &\frac{\partial \|\mathbf{r}_N\|^2}{\partial \alpha} = 0 \\ &\Longrightarrow -2 \langle \mathbf{r}_N, \boldsymbol{\omega} \odot \mathbf{g} \rangle + 2a \|\boldsymbol{\omega} \odot \mathbf{g}\|^2 = 0 \\ &\Longrightarrow \alpha = \frac{\langle \mathbf{r}_N, \boldsymbol{\omega} \odot \mathbf{g} \rangle}{\|\boldsymbol{\omega} \odot \mathbf{g}\|^2}. \end{aligned} \quad (10)$$

For this optimal value of α, we have

$$\begin{aligned} \|\mathbf{r}_{N+1}\|^2 &= \|\mathbf{r}_N\|^2 - 2 \cdot \frac{\langle \mathbf{r}_N, \boldsymbol{\omega} \odot \mathbf{g} \rangle}{\|\boldsymbol{\omega} \odot \mathbf{g}\|^2} \cdot \langle \mathbf{r}_N, \boldsymbol{\omega} \odot \mathbf{g} \rangle \\ &\quad + \left[\frac{\langle \mathbf{r}_N, \boldsymbol{\omega} \odot \mathbf{g} \rangle}{\|\boldsymbol{\omega} \odot \mathbf{g}\|^2} \right]^2 \cdot \|\boldsymbol{\omega} \odot \mathbf{g}\|^2 \\ &= \|\mathbf{r}_N\|^2 - \left(\frac{\langle \mathbf{r}_N, \boldsymbol{\omega} \odot \mathbf{g} \rangle}{\|\boldsymbol{\omega} \odot \mathbf{g}\|} \right)^2. \end{aligned} \quad (11)$$

So $\mathbf{g} \in \mathbf{D}$ that minimizes expression (9) is the one that minimizes (11); we can get

$$\begin{aligned} \mathbf{g}_{N+1} &= \arg\min_{\mathbf{g} \in \mathbf{D}} \left(\|\mathbf{r}_N\|^2 - \left(\frac{\langle \mathbf{r}_N, \boldsymbol{\omega} \odot \mathbf{g} \rangle}{\|\boldsymbol{\omega} \odot \mathbf{g}\|} \right)^2 \right) \\ \Longrightarrow \mathbf{g}_{N+1} &= \arg\max_{\mathbf{g} \in \mathbf{D}} \left| \left(\frac{\langle \mathbf{r}_N, \boldsymbol{\omega} \odot \mathbf{g} \rangle}{\|\boldsymbol{\omega} \odot \mathbf{g}\|} \right) \right|, \end{aligned} \quad (12)$$

and the corresponding coefficient α_{N+1} is

$$\alpha_{N+1} = \frac{\langle \mathbf{r}_N, \boldsymbol{\omega} \odot \mathbf{g}_{N+1} \rangle}{\|\boldsymbol{\omega} \odot \mathbf{g}_{N+1}\|^2}. \quad (13)$$

At every step in this algorithm, not only the set of basis functions $\mathbf{g}_i$ but also their coefficients α_i are obtained. So, the approximate form of observations $\{y_1, y_2, \ldots, y_l\}$ in ith iteration is

$$\mathbf{f}_i = \sum_{k=1}^{i-1} \alpha_k \mathbf{g}_k + \alpha_i \mathbf{g}_i. \quad (14)$$

However, when appending $\alpha_i \mathbf{g}_i$, the linear combination may not be optimal. So a backfitting method is also done by reoptimizing all coefficients $\alpha_1, \ldots, \alpha_i$ to minimize the target cost:

$$\begin{aligned} \alpha_1, \ldots, \alpha_i &= \arg\min_{\alpha_k \in \mathbf{R}(i=1,\ldots,i)} \|\omega \odot (\mathbf{f}_i - \mathbf{y})\|^2 \\ &= \arg\min_{\alpha_k \in \mathbf{R}(i=1,\ldots,i)} \left\| \omega \odot \left(\sum_{k=1}^{i} \alpha_k \mathbf{g}_k - \mathbf{y} \right) \right\|^2. \end{aligned} \quad (15)$$

Finally, we can get the approximation functions in regression:

$$\mathbf{f}_N(\mathbf{x}) = \sum_{k=1}^{N} \alpha_i \mathbf{g}_i(\mathbf{x}) = \sum_{i \in \{sp\}} \alpha_i k(\mathbf{x}, \mathbf{x}_i) \quad (16)$$

or decision function in classification:

$$\mathbf{f}_N(\mathbf{x}) = \operatorname{sgn}\left(\sum_{k=1}^{N} \alpha_i \mathbf{g}_i(\mathbf{x}) \right) = \operatorname{sgn}\left(\sum_{i \in \{sv\}} \alpha_i k(\mathbf{x}, \mathbf{x}_i) \right), \quad (17)$$

where $\{sv\}$ is the support vector set obtained by IFKMP algorithm.

2.4. The Selection of Intuitionistic Fuzzy Parameter. In practice, the recognition accuracy of different targets is different according to the threat degree. Generally, higher threat degree of the targets needs higher recognition accuracy. Aiming at this fact, the selection algorithm of intuitionistic fuzzy parameter can be listed as follows.

Algorithm 1 (the selection algorithm of intuitionistic fuzzy parameter).

Input. Input is the type of sample y_i.

Output. Output is the intuitionistic fuzzy parameter $\omega(y_i)$.

Step 1. Determine y_i as appointed important target or nonappointed target.

Step 2. Calculate the membership degree $\mu(y_i)$ and nonmembership degree $\gamma(y_i)$; this paper uses Gaussian function as membership function, so

$$\mu(y_i) = \exp\left(-\frac{(y_i - c)^2}{2\rho}\right),$$
$$\gamma(y_i) = \delta(y_i) - \exp\left(-\frac{(y_i - c)^2}{2\rho}\right), \tag{18}$$

where ρ and c denote width and center point, respectively, $\delta(y_i) = 1 - \pi(y_i)$ is denoted as nonhesitancy degree, and $\pi(y_i)$ denote the uncertainty degree which must be predefined on the basis of practice use.

Step 3. Calculate the intuitionistic fuzzy parameter $\omega(y_i)$ by the following formula:

$$\omega(y_i) = \begin{cases} \mu(y_i) + \sigma(y_i) & y_i \text{ as appointed important target} \\ \dfrac{\sigma(y_i) - \gamma(y_i)}{\sigma(y_i) + \gamma(y_i)} & y_i \text{ as nonappointed target.} \end{cases} \tag{19}$$

The intuitionistic fuzzy parameter can distinguish the appointed important target and nonappointed important target specifically and classify the appointed important samples much more precisely. Moreover, the total misclassification rate will still be lower even with a higher misclassification rate ε of the nonappointed important target.

3. IFKMP Based on Particle Swarm Optimization

Theoretically IFKMP algorithm solves the problem that some important data cannot be classified precisely according to the predefined importance of the data efficiently, but its implement is a greedy algorithm. Greedy algorithm requires that every step of searching process be global optimal searching in the redundant dictionary of function in order to select best matching signal structure, from which the large amounts of computing time have often not suffered. Particle swarm optimization algorithm with powerful ability of global search and quick convergence rate solves the problem, by replacing greedy algorithm to find a linear expansion of basis functions.

3.1. Theoretical Analysis. We notice that a bottleneck of the IFKMP algorithm is represented by the search of the current best element from the redundant dictionary to be added in the function expansion. Usually, this requires a full search over the whole dictionary and may necessitate a large number of floating point operations. An alternative is provided by the so-called Weak Greedy Algorithm (WGA) [17] which provides an approximation of the MP and related greedy algorithms. WGA and its different formulations have been analyzed in [17, 18] and proofs of convergence of the algorithm exist under various conditions. Unlike classical greedy algorithm, WGA generates an approximate sequence as follows:

$$\begin{aligned} \widetilde{f}_{N+1} &= \widetilde{f}_N + t_{N+1}\alpha_{N+1}\mathbf{g}_{N+1} \\ &= \widetilde{f}_N + \widetilde{\alpha}_{N+1}\mathbf{g}_{N+1}, \quad t_{N+1} \in [0, 1], \end{aligned} \tag{20}$$

where $\widetilde{\alpha}_{N+1} = t_{N+1}\alpha_{N+1}$, and, clearly, for $t_{N+1} = 1$, WGA retrieves the original algorithm. The sequence $\tau = \{t_1, t_2, \ldots, t_N\}$ is named weakness sequence and it must obey some constraints for the algorithm to converge. While different conditions on τ result in different guaranteed convergence rates, we will simply require that $\exists \widetilde{t} > 0$ such that $\forall N \geq 1$, $t_N > \widetilde{t}$, which ensures the convergence [18]. These modifications imply that we are no longer forced to produce the global maximum at each iteration of the IFKMP algorithm but just an approximate optimal value. So we can use PSO algorithm to search the next element from the redundant dictionary instead of greedy algorithm. Compared with greedy algorithm, the PSO algorithm may only find an approximate optimal value; we still can ensure the convergence of PS-IFKMP algorithm according to the theory of WGA. As a consequence, the performance of the decision function we get from PS-IFKMP algorithm is similar to the decision function from IFKMP algorithm under the same termination condition. Based on this point, we believe that PS-IFKMP algorithm can decrease computational complexity while classification accuracy remains statistically the same by using PSO algorithm to find a linear expansion of basis functions.

3.2. Particle Swarm Optimization. Particle swarm optimization is an evolutionary optimization technique based on metaphors for social interaction and communication such as flocks of birds and schools of fish. This stochastic, population-based approach has been proven effective for solving both continuous and discrete optimization problems. Each particle in a swarm, which is analogous to a bird in a flock or a fish in a school, moves around in d dimensional search space. Based on its own experience and that of the swarm, it moves toward the best position in the search space [19].

The position and velocity of particle i at iteration t are represented by $\mathbf{X}_i^t$ and $\mathbf{V}_i^t$, which can be defined as $\mathbf{X}_i^t = (x_{i1}^t, x_{i2}^t, \ldots, x_{id}^t)$ and $\mathbf{V}_i^t = (v_{i1}^t, v_{i2}^t, \ldots, v_{id}^t)$, respectively, and the personal best ($p$best) of particle i is represented by $\mathbf{P}_i^t$, which denotes the position of particle i with the best fitness value found so far and is defined as $\mathbf{P}_i^t = (p_{i1}^t, p_{i2}^t, \ldots, p_{id}^t)$. The global best ($g$best) of all particles at iteration t is represented by $\mathbf{P}_g^t$, which denotes the best position of the particle with the best fitness value in the swarm found so far and is defined as $\mathbf{P}_g^t = (p_{g1}^t, p_{g2}^t, \ldots, p_{gd}^t)$. The new velocity and position of particle i can be obtained by (20) and (21), respectively:

$$v_{id}^{t+1} = \omega \cdot v_{id}^t(t) + c_1 \cdot r_1\left(p_{id}^t - x_{id}^t\right) + c_2 \cdot r_2\left(p_{gd}^t - x_{id}^t\right), \tag{21}$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}, \tag{22}$$

where $v_{id}^t(t)$ represents velocity of particle i at iteration t with respect to the dth dimension and v_{id}^{t+1} is new velocity at iteration $t+1$, x_{id}^t is position of particle i at iteration t with respect to the dth dimension, and x_{id}^{t+1} is new position at iteration $t+1$. p_{id}^t is position value of the ith $p$best at iteration t with respect to the dth dimension and p_{gd}^t is position value of the $g$best at iteration t. t is current iteration. c_1, c_2 are acceleration coefficients. r_1, r_2 are uniform random numbers between 0 and 1. ω is the inertia weight. Moreover, in order to make the speed of the particle not too high, a speed threshold $v_{\max}$ should be set. If $v_{id}^t > v_{\max}$, then $v_{id}^t = v_{\max}$, and if $v_{id}^t < -v_{\max}$, then $v_{id}^t = -v_{\max}$.

3.3. Target Recognition Algorithm of IFKMP Based on PSO. IFKMP algorithm uses greedy algorithm to find a linear combination of basis functions, that is, the decision function $\mathbf{f}_N$ in classification from a redundant dictionary of functions to minimize the objective function f. The PS-IFKMP algorithm first maps training data into a redundant basis dictionary $\mathbf{D}$ by a given kernel function and then sets the population size according to the scale of the training dataset. At last, the linear expansion of basis functions is found by using PSO algorithm. Because the larger the value of linear expansion of basis functions, the more effective the Classifier, formula (16) is selected as the fitness function. The detailed steps of PS-IFKMP algorithm are listed as follows.

Algorithm 2 (intuitionistic fuzzy kernel matching pursuit based on particle swarm optimization).

Input. Input is the training dataset $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$, the intuitionistic fuzzy parameters $\omega(y_i)$, and the kernel parameter ρ.

Output. Output is the decision function $\mathbf{f}_N(x)$.

Step 1 (initialization). Set the maximum iterations L and iterative threshold ε of IFKMP, set the population size N and the maximum iterations $\text{iter}_{\max}$ of PSO, and suppose generation $k = 1$, $t = 1$.

Step 2. Construct the basis dictionary $\mathbf{D}$ on the training data: $\mathbf{D} = \{\mathbf{g}_i = K(\cdot, \mathbf{x}_i) \mid i = 1, \ldots, l\}$.

Step 3. Calculate coefficients $\alpha_i = \mathbf{g}_i^T(\mathbf{x}) \cdot \mathbf{y} / \|\mathbf{g}_i(\mathbf{x})\|^2$ based on the rule $\min_{\alpha_i} \|\boldsymbol{\omega} \odot (\mathbf{y} - \alpha_i \mathbf{g}_i(\mathbf{x}))\|$; then $\alpha_1^t, \alpha_2^t, \ldots, \alpha_N^t$ are selected as the initial population of particles randomly.

Step 4. Measure the fitness of each particle in the population by $\mathbf{f}_t(\alpha_i, \mathbf{g}_i) = \sum_{t=1}^{L} \alpha_i^t \mathbf{g}_i^t$.

Step 5. Update particles according to formulas (21) and (22); then update the $p$best $\mathbf{P}_i^t$ and $g$best $\mathbf{P}_g^t$.

Step 6. If $k \geq \text{iter}_{\max}$, stop the iteration; otherwise $k = k + 1$, and go to Step 4.

Step 7. Let $\mathbf{y} = \mathbf{y} - \mathbf{f}_t$, and if $\mathbf{y} \geq \varepsilon$ and $t < L$, go to Step 3 by using the updated $\mathbf{y}$ and solve the next basic function $\mathbf{g}_i$ and weight coefficient α_i.

Step 8. Recompute the optimal set of coefficients $\alpha_1, \ldots, \alpha_i$ by backfitting algorithm; then output the linear combination of the best weight coefficients and the corresponding basic functions, which is also the decision function $\mathbf{f}_N(x)$.

The flowchart of Algorithm 2 is shown in Figure 1.

3.4. Parameters Setting of PS-IFKMP. The inertia weight ω can be a positive constant or even a positive linear or nonlinear function of time. When $\omega > 1.2$, the velocity item becomes the main item in the search direction of the particle. It extends the search area and finds the global optimum. When ω is between 0.8 and 1.2, three factors, velocity, $p$best, and $g$best, affect the velocity calculation for both local search and global search. When $\omega < 0.8$, only the $p$best and $g$best affect the new velocity calculation, which converges to the local optimum. Therefore, the value of inertia weight ω is a tradeoff between the global search and the local search. In this paper, the inertia weight starts with a high value $\omega_{\max}$ and nonlinearly decreases to $\omega_{\min}$ at the maximal number of iterations [20]:

$$\omega = \omega_{\min} + \left(\frac{\text{iter}_{\max} - \text{iter}}{\text{iter}_{\max}} \right)^{\beta} \times (\omega_{\max} - \omega_{\min}), \tag{23}$$

where $\omega_{\max}$ and $\omega_{\min}$ are the initial and final values of the inertia weight, respectively, iter denotes the current number of iterations, and $\text{iter}_{\max}$ denotes the maximum number of iterations. The value β is a constant coefficient.

The acceleration factors c_1 and c_2 are positive constants controlling the relative impact of the personal (local) and common (global) knowledge on the movement of each particle, and we improve the selection of acceleration factors by introducing inheritance mechanism:

$$c_1 \cdot r_1 = \omega_1 = \frac{\mathbf{p}_{id}^t}{\mathbf{f}_{id}^t} + \omega, \qquad c_2 \cdot r_2 = \omega_2 = \frac{\mathbf{p}_{gd}^t}{\mathbf{p}_{id}^t}, \tag{24}$$

where $\mathbf{f}_{id}^t$ is the fitness value of particle i at iteration t and the convergence rate of PS-IFKMP speeds up efficiently by introducing inheritance mechanism.

In order to keep the diversity of the population, the Gaussian distribution is introduced. If the $\mathbf{P}_g^t$ has not been improved in a few steps, a certain proportion of particles should be randomly selected for mutation:

$$x_{id}^{t\prime} = x_{id}^t + r_0 \cdot x_{id}^t, \tag{25}$$

where r_0 is a Gaussian random number with mean of 0 and standard deviation of 1.

3.5. Complexity Analysis. The performance of the proposed PS-IFKMP algorithm is analyzed in theory in this section. The IFKMP is a novel algorithm which was proposed in 2011, and no relevant improved algorithms have been proposed at present. Therefore we mainly compared theoretical complexity of the treatment of the proposed PS-IFKMP algorithm with IFKMP algorithm. IFKMP algorithm maps training data into a high-dimensional Hilbert space by a given

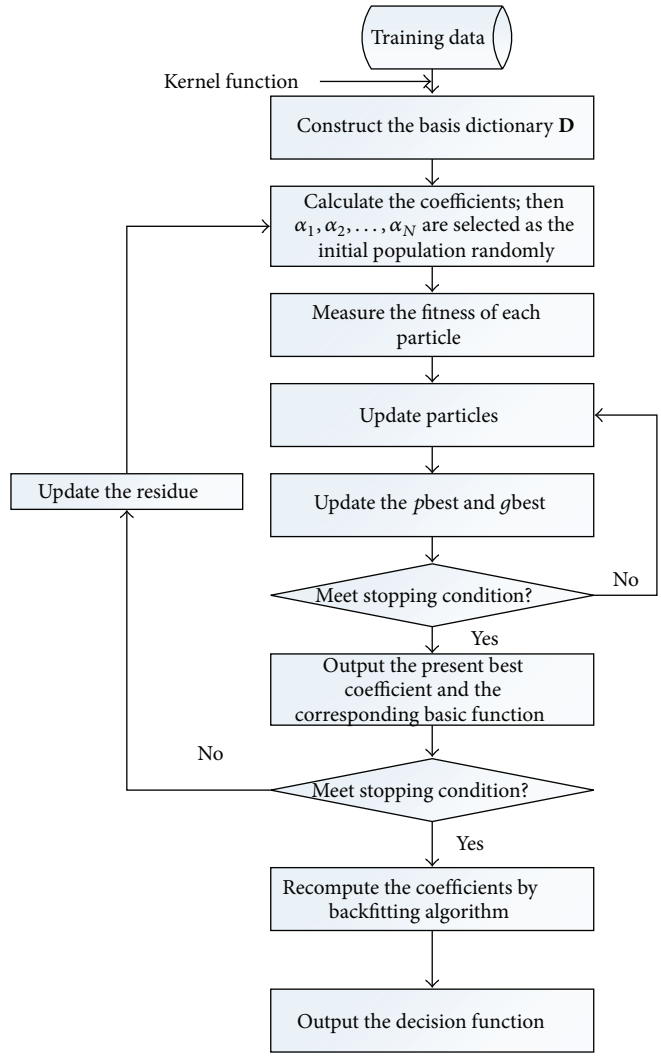


Figure 1: Flowchart of PS-IFKMP algorithm.

kernel function to construct the basis function dictionary and then searches the best basis functions and the corresponding weight in each iteration. So the computation complexity can be described by the number of multiplications in one matching process in this paper.

Assuming that the size of dictionary is n and the iteration number is L, IFKMP algorithm requires $n \cdot L$ single-precision multiplications in one matching process, and the complexity can be described as $O(n \cdot L)$ based on the computational analyses of Montgomery reduction algorithm. In PSO algorithm, position, velocity, and fitness of each particle need to be calculated in every iteration, so the algorithm requires $3 \cdot N \cdot L$ single-precision multiplications in one matching process, where N is the population size and the complexity of PS-IFKMP can be described as $O(3 \cdot N \cdot L)$. As the population size N is far less than the dictionary size n, the computational cost of PS-IFKMP is far smaller than that of IFKMP when n is quite big, while the costs are familiar when n is small.

4. Experiment Results and Analysis

This section evaluates the performance of the proposed PS-IFKMP algorithm by comparing its classification results with those obtained for the same cases by the KMP in [1] and IFKMP in [9] where PSO is encoded by real number and RBF kernel is used for both algorithms. To avoid the weak problem, each experiment has been performed at 50 independent runs. The average results and standard deviations (Std. in short) are given. All experiments were carried on a Pentium (R) Dual-Core CPU E5500 @2.8 GHz with 2 GB RAM using MATLAB 7.6 compiler.

4.1. Test on UCI Data. In this experiment, five UCI datasets of Musk, Waveform, German, Diabetes, and Breast Cancer Wisconsin (Breast in short) are used to test our methods, where Waveform contains three classes, and two of them (class 0 and class 2) are selected as experimental dataset.

Table 1: UCI dataset characteristics.

Title	Instances	Attributes	Classes
Musk	6598	166	2
Waveform	5000	21	3
German	1000	20	2
Diabetes	768	8	2
Breast	699	9	2

The characteristics of the UCI datasets are shown in Table 1, which shows a wide range of attributes sizes and class sizes.

The selection of different kernel function parameters has great influence on algorithm performance, so this paper validates the kernel parameter by a special training dataset. The intuitionistic fuzzy parameters are set as $\omega(y_1) = \omega(y_2) = 1$, $L = 200$, $\varepsilon = 0.05$, $N = 30$, and $\text{iter}_{\max} = 50$ and the kernel function parameter ρ is taken in $[1, 500]$ by equal interval sampling. The influence of ρ on the accuracy is shown in Figure 2.

From Figure 2, we can notice that the optimal kernel function parameters based on different datasets are different. Validated by experiments, $\rho = 360$, 3, and 6 for Musk, Waveform, and German, respectively. There is no special requirement on the importance of samples; we can set $\omega(y_1) = \omega(y_2) = 1$; then IFKMP algorithm is equivalent to KMP algorithm, so we only compare PS-IFKMP with the KMP in this section; the results are listed in Tables 2–4.

From the above experiments, the average accuracy of KMP and PS-IFKMP is 85.37% and 86.34%, respectively, so the performance of the PS-IFKMP is competitive to that of KMP for classification. However, the average training time in one matching process of KMP and PS-IFKMP is 0.16 and 0.048, respectively; compared with KMP, the average training time of PS-IFKMP is decreased by 70%. So the proposed PS-IFKMP algorithm can decrease obviously training time with almost unchanged classification accuracy, especially for the large size datasets.

Diabetes contains 500 negative samples and 268 positive samples, and Breast contains 241 negative samples and 458 positive samples. As we know, the positive samples describe the pathological features of diseases, so the positive samples should be classified much more precisely in terms of medical diagnosis. Validated by experiments, $\rho = 15$ and 3 for Diabetes and Breast, respectively. The intuitionistic fuzzy parameters are selected by Algorithm 1, $\omega(y_1)$ of appointed positive samples is 1.6, and $\omega(y_2)$ of nonappointed negative samples is 0.3. The results are listed in Table 5.

From Table 5, facing imbalanced training set, standard KMP cannot classify the appointed important samples efficiently. IFKMP solves the aforementioned problem by assigning the intuitionistic fuzzy parameters to different samples and studying the weak samples sufficiently. Compared with IFKMP, the proposed PS-IFKMP algorithm can decrease training time obviously, and the classification accuracy is almost unchanged simultaneously.

4.2. Test on Artificial Data. Learning to tell two spirals apart is important both for purely academic reasons and for industrial application. In the research of the pattern recognition, it is a well-known problem for its difficulty. The parametric equation of the two spirals can be presented as follows:

$$\begin{aligned} &\text{Spiral-1:} \\ &x_1 = (k_1\theta + e_1)\cos\theta \\ &y_1 = (k_1\theta + e_1)\sin\theta \\ &(\theta \in U \sim [0, 2\pi]), \\ &\text{Spiral-2:} \\ &x_2 = (k_2\theta + e_2)\cos\theta \\ &y_2 = (k_2\theta + e_2)\sin\theta \\ &(\theta \in U \sim [0, 2\pi]), \end{aligned} \tag{26}$$

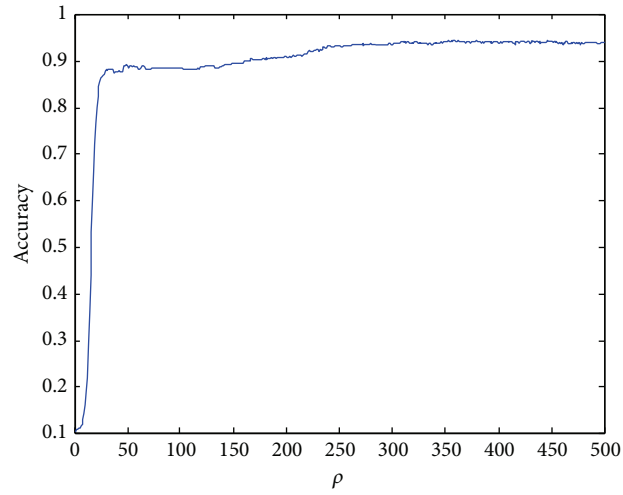
where k_1, k_2, e_1, and e_2 are parameters. In our experiment, we choose $k_1 = k_2 = 4$, $e_1 = 1$, and $e_2 = 3$. We generate 12000 samples randomly, and the samples distribution is shown in Figure 3(a). Moreover, a set of concentric circles samples also is selected to test our methods, and the parametric equation of the concentric circles data can be shown as

$$\begin{aligned} x &= \sigma \cdot \cos\theta \\ y &= \sigma \cdot \sin\theta \\ (\theta &\in U \sim [0, 2\pi]). \end{aligned} \tag{27}$$

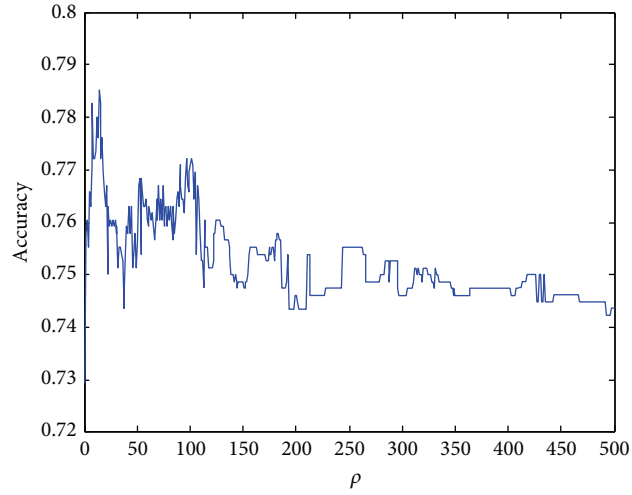
Parameter σ of the first class is of continuous uniform distribution $U[0, 55]$ and the second class $\sigma \sim U[45, 100]$. 16000 samples are generated randomly, and the samples distribution is shown in Figure 3(b). Setting $L = 150$, $\varepsilon = 0.02$, $N = 50$, $\text{iter}_{\max} = 50$, and $\rho = 8$, the intuitionistic fuzzy parameters are selected by Algorithm 1; $\omega(y_1)$ of appointed important samples (shown by "•" in Figure 3) is 1.5 and $\omega(y_2)$ of nonappointed samples (shown by "×" in Figure 3) is 0.3. Before the training, noise is added to training data, randomly choosing 10% samples and changing its class attributes. The results are listed in Table 6.

In Table 6, it is easily found that KMP cannot classify the appointed important samples efficiently when training Classifier with a subset of the data. IFKMP can classify the appointed important samples much more precisely by assigning the intuitionistic fuzzy parameters to different samples and studying the weak samples sufficiently. The proposed PS-IFKMP with advantage of both PSO and IFKMP reduces the training time enormously with the classification accuracy almost unchanged.

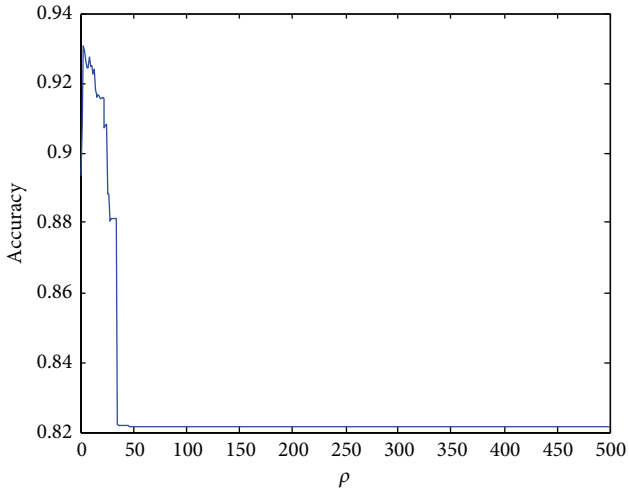
4.3. Test on Aerospace Target Data. In order to test the performance of the proposed PS-IFKMP algorithm in practical application, the aerospace targets data are selected for experiment. Generally, the threat degree of tactical ballistic missile (TBM), cruise missile (CM), and stealth aircraft (SA) is higher than the other targets, so we set these three targets as the appointed important targets. The radar cross



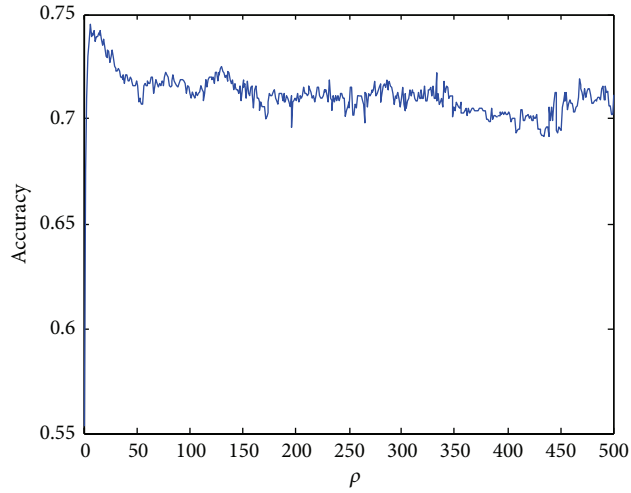
(a) Accuracy varying with ρ based on Musk data



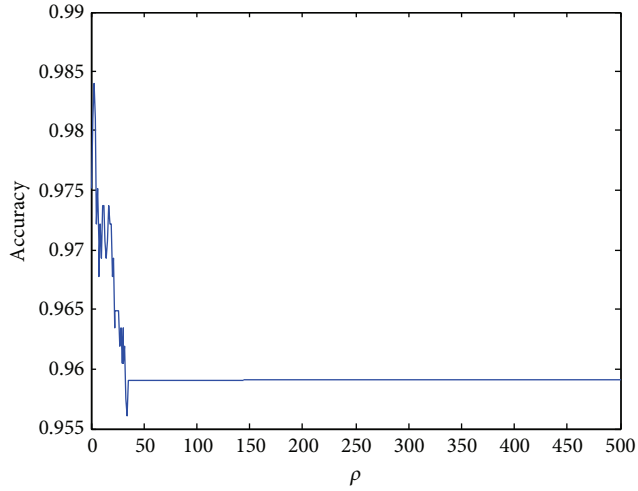
(b) Accuracy varying with ρ based on Diabetes data



(c) Accuracy varying with ρ based on Waveform



(d) Accuracy varying with ρ based on German data



(e) Accuracy varying with ρ based on Breast data

Figure 2: Accuracy varying with ρ.

Table 2: Testing of Musk dataset.

Title	Training set	Tested data	Method	# s. v.[a]	Training time (s)	Accuracy (%)	Std. of error (%)
Musk	1000	2000	KMP	156	0.389	92.12	**2.17**
			PS-IFKMP	132	**0.097**	**92.43**	2.92
	300	2000	KMP	100	0.084	**90.44**	4.76
			PS-IFKMP	97	**0.036**	90.26	**4.69**

[a]# s. v.: number of support vectors.

Table 3: Testing of Waveform dataset.

Title	Training set	Tested data	Method	# s. v.	Training time (s)	Accuracy (%)	Std. of error (%)
Waveform	1000	1000	KMP	120	0.297	**92.70**	3.16
			PS-IFKMP	113	**0.094**	92.43	**3.08**
	300	1000	KMP	123	0.072	**91.56**	1.59
			PS-IFKMP	115	**0.022**	91.47	**1.30**

Table 4: Testing of German dataset.

Title	Training set	Tested data	Method	# s. v.	Training time (s)	Accuracy (%)	Std. of error (%)
German	600	500	KMP	28	0.096	73.62	1.35
			PS-IFKMP	23	**0.029**	**73.86**	**1.33**
	300	500	KMP	32	0.022	**71.80**	2.82
			PS-IFKMP	27	**0.008**	71.60	**2.19**

Table 5: Testing of appointed important samples.

Title	Training set	Tested data	Method	# s. v.	Training time (s)	Accuracy (%)	Std. of error (%)
Diabetes	Positive: 54 Negative: 100	Positive: 67	KMP	32	0.014	66.67	3.14
			IFKMP	33	0.016	98.45	**2.76**
			PS-IFKMP	29	**0.008**	**98.82**	2.97
Breast	Positive: 60 Negative: 115	Positive: 72	KMP	16	0.012	78.26	3.37
			IFKMP	13	0.014	**99.12**	3.14
			PS-IFKMP	15	**0.009**	98.92	**3.07**

Table 6: Testing of artificial dataset.

Title	Training set	Tested data	Method	# s. v.[a]	Training time (s)	Accuracy (%)	Std. of error (%)
Two spirals	Positive: 500 Negative: 500	Positive: 1000	KMP	24	0.362	90.26	2.43
			IFKMP	18	0.396	97.16	2.17
			PS-IFKMP	15	**0.047**	**98.12**	**1.85**
Concentric circles	Positive: 500 Negative: 500	Positive: 1000	KMP	200	0.453	91.13	1.96
			IFKMP	165	0.470	**99.03**	**1.22**
			PS-IFKMP	165	**0.121**	98.46	1.52

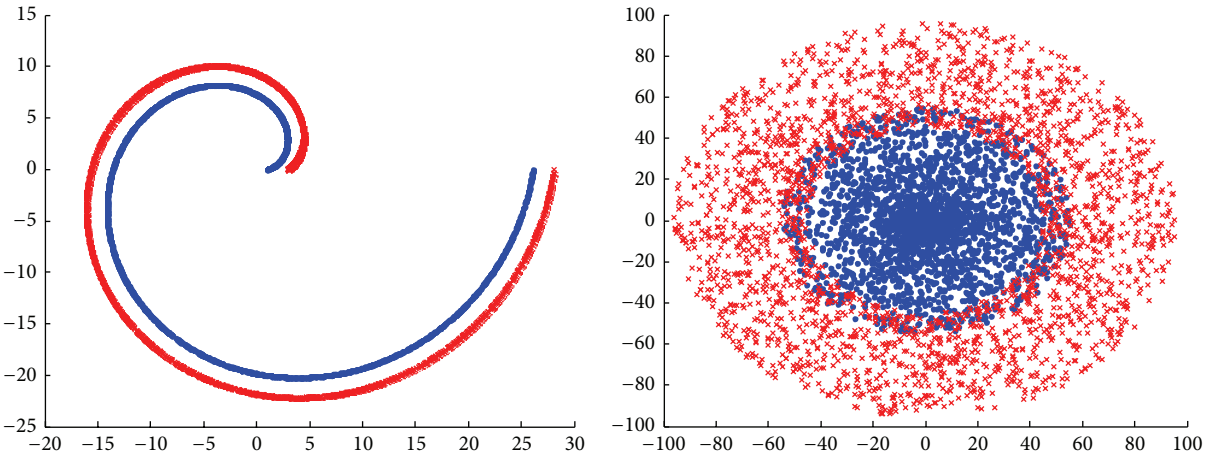
section (RCS) values of the target are simulated by FEKO software, and the maximum, minimum, mean, and variance of RCS sequence are selected as sample features. And the RCS sequence of TBM within 30 seconds is showed in Figure 4.

Beside the appointed important targets, a large number of debris, baits, and other aircrafts, and so forth, are selected as negative samples. Setting $L = 200$, $\varepsilon = 0.05$, $N = 30$, $\text{iter}_{\max} = 50$, and $\rho = 3.5$, the $\omega(y_1)$ of TBM equals 1.5, the $\omega(y_2)$ of CM equals 1.3, and the $\omega(y_3)$ of CM equals 1.6 by Algorithm 1. The results are listed in Table 7.

From Table 7, by providing equal treatment on all training samples, the classical KMP algorithm cannot classify the weak important data efficiently. The IFKMP circumvents this problem by studying the important samples sufficiently and the less important samples roughly; however, the complexity problem still exists by adopting the greedy algorithm.

Table 7: Testing of artificial dataset.

Title	Training set	Tested data	Method	Training time (s)	Accuracy (%)	Std. of error (%)
TBM	Positive: 122 Negative: 478	Positive: 83	KMP	0.285	47.76	6.49
			IFKMP	0.289	80.89	**4.26**
			PS-IFKMP	**0.116**	**81.65**	4.75
CM	Positive: 223 Negative: 387	Positive: 148	KMP	0.247	67.35	5.62
			IFKMP	0.254	84.66	4.88
			PS-IFKMP	**0.095**	**85.99**	**4.37**
SA	Positive: 188 Negative: 412	Positive: 125	KMP	0.234	54.26	6.36
			IFKMP	0.255	**82.75**	**4.65**
			PS-IFKMP	**0.130**	82.51	5.02



(a) The distribution map of two-spirals data

(b) The distribution map of concentric circles

Figure 3: The distribution map of artificial data.

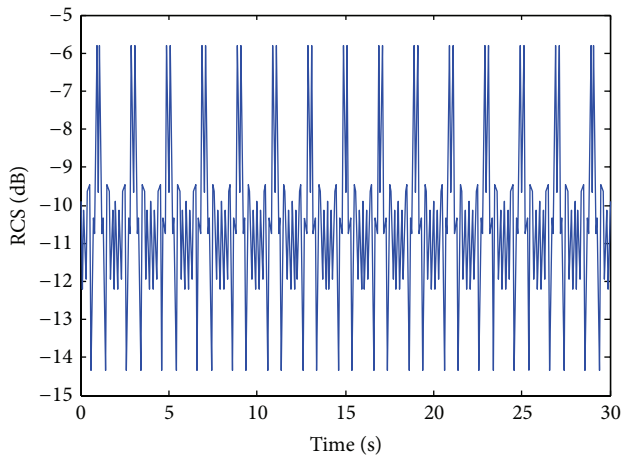


Figure 4: RCS sequence of TBM within 30 s.

By introducing particle swarm optimization algorithm, PS-IFKMP decreases the time complexity and reduces the training time with almost unchanged classification accuracy. It is also demonstrated that this approach is suitable to the application requiring both accuracy and timeliness.

5. Conclusion

Theoretically IFKMP algorithm is an excellent method but its implement is based on a greedy algorithm. Therefore, the defect of great computer time has not been solved when dealing with huge volume of data. In order to overcome the limitation, we propose the PS-IFKMP algorithm which optimizes searching process based on particle swarm optimization and reduces computational complexity of IFKMP algorithm. Simulation results show that, compared with the conventional approaches, the proposed algorithm can decrease training time and improve calculation efficiency obviously with the classification accuracy almost unchanged, especially for large size datasets. However, further improvements are needed by this algorithm, and the future research should focus on problems such as how to provide better multiclass recognition performance and how to select the parameters of the kernel function.

Conflict of Interests

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

Acknowledgment

The work was supported by the National Natural Science Foundation of China (nos. 61272011 and 61309022).

References

[1] P. Vincent and Y. Bengio, "Kernel matching pursuit," *Machine Learning*, vol. 48, no. 1–3, pp. 165–187, 2002.

[2] V. Cevher and A. Krause, "Greedy dictionary selection for sparse representation," *IEEE Journal on Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 979–988, 2011.

[3] P. Sun and X. Yao, "Sparse approximation through boosting for learning large scale kernel machines," *IEEE Transactions on Neural Networks*, vol. 21, no. 6, pp. 883–894, 2010.

[4] V. Popovici, S. Bengio, and J.-P. Thiran, "Kernel matching pursuit for large datasets," *Pattern Recognition*, vol. 38, no. 12, pp. 2385–2390, 2005.

[5] L. Jiao and Q. Li, "Kernel matching pursuit classifier ensemble," *Pattern Recognition*, vol. 39, no. 4, pp. 587–594, 2006.

[6] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification via kernel sparse representation," in *Proceedings of the 18th IEEE International Conference on Image Processing (ICIP '11)*, pp. 1233–1236, September 2011.

[7] W. Wang, X. Yang, and S. H. Chen, "A Kernel matching pursuit approach to man-made objects detection in aerial images," in *Pattern Recognition and Image Analysis*, vol. 4478 of *Lecture Notes in Computer Science*, pp. 507–514, Springer, Berlin, Germany, 2007.

[8] Q. Li, L.-C. Jiao, and W.-D. Zhou, "Pattern recognition based on the fuzzy kernel matching pursuit," *Chinese Journal of Computers*, vol. 32, no. 8, pp. 1687–1694, 2009.

[9] Y. Lei, Y.-J. Lei, C.-M. Zhou, and W.-W. Kong, "Techniques for target recognition based on intuitionistic fuzzy kernel matching pursuit," *Acta Electronica Sinica*, vol. 39, no. 6, pp. 1441–1446, 2011.

[10] J. W. Li and Y. Lu, "Refining kernel matching pursuit," in *Advances in Neural Networks—ISNN 2010*, vol. 6064 of *Lecture Notes in Computer Science*, pp. 25–32, Springer, 2010.

[11] Y. Zanjireh, A. H. Rezaie, and H. Amindavar, "Multi component signal decomposition based on chirplet pursuit and genetic algorithms," *Applied Acoustics*, vol. 74, no. 12, pp. 1333–1342, 2013.

[12] T. Ray, M. Asafuddoula, and A. Isaacs, "A steady state decomposition based quantum genetic algorithm for many objective optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 2817–2824, 2013.

[13] Y. Lei, W. W. Kong, and Y. J. Lei, "Technique for target recognition based on intuitionistic fuzzy c-means clustering and kernel matching pursuit," *Journal on Communications*, vol. 33, no. 11, pp. 136–143, 2012.

[14] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, Australia, December 1995.

[15] S. Rana, S. Jasola, and R. Kumar, "A review on particle swarm optimization algorithms and their applications to data clustering," *Artificial Intelligence Review*, vol. 35, no. 3, pp. 211–222, 2011.

[16] M. Salahi, A. Jamalian, and A. Taati, "Global minimization of multi-funnel functions using particle swarm optimization," *Neural Computing and Applications*, vol. 23, no. 7-8, pp. 2101–2106, 2013.

[17] V. N. Temlyakov, "Weak greedy algorithms," *Advances in Computational Mathematics*, vol. 12, no. 2-3, pp. 213–227, 2000.

[18] A. V. Sil'nichenko, "On the convergence of order-preserving weak greedy algorithms," *Mathematical Notes*, vol. 84, no. 5-6, pp. 741–747, 2008.

[19] J.-T. Tsai, C.-I. Yang, and J.-H. Chou, "Hybrid sliding level Taguchi-based particle swarm optimization for flowshop scheduling problems," *Applied Soft Computing Journal*, vol. 15, pp. 177–192, 2014.

[20] W. Zhang, D. Ma, J.-J. Wei, and H.-F. Liang, "A parameter selection strategy for particle swarm optimization based on particle positions," *Expert Systems with Applications*, vol. 41, no. 7, pp. 3576–3584, 2014.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 329753, 8 pages
http://dx.doi.org/10.1155/2015/329753

Research Article

Multiview Sample Classification Algorithm Based on L1-Graph Domain Adaptation Learning

Huibin Lu, Zhengping Hu, and Hongxiao Gao

School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China

Correspondence should be addressed to Zhengping Hu; hzp@ysu.edu.cn

Received 14 November 2014; Revised 4 February 2015; Accepted 7 February 2015

Academic Editor: Gerhard-Wilhelm Weber

In the case of multiview sample classification with different distribution, training and testing samples are from different domains. In order to improve the classification performance, a multiview sample classification algorithm based on L1-Graph domain adaptation learning is presented. First of all, a framework of nonnegative matrix trifactorization based on domain adaptation learning is formed, in which the unchanged information is regarded as the bridge of knowledge transformation from the source domain to the target domain; the second step is to construct L1-Graph on the basis of sparse representation, so as to search for the nearest neighbor data with self-adaptation and preserve the samples and the geometric structure; lastly, we integrate two complementary objective functions into the unified optimization issue and use the iterative algorithm to cope with it, and then the estimation of the testing sample classification is completed. Comparative experiments are conducted in USPS-Binary digital database, Three-Domain Object Benchmark database, and ALOI database; the experimental results verify the effectiveness of the proposed algorithm, which improves the recognition accuracy and ensures the robustness of algorithm.

1. Introduction

Traditional machine learning algorithms are usually applicable to the data, during which the training and testing samples are from the same characteristic space with the same distribution. With the change of characteristic and distribution, most statistical models need to be reconstructed by the new training sample collection. But, in practice, the training and testing samples are often collected in different periods and different environment; therefore, the distribution may be different. In order to solve this problem, the transfer learning theory is introduced, which aims to transform knowledge from the source domain marked to the target domain unmarked, for example, natural language learning processing [1], sentiment analysis [2], and image classification [3]. Using transfer learning to improve the recognition accuracy and robustness of different distribution is widely concerned by the domestic and foreign scholars.

In recent years, in view of the fact that the marked training samples and unmarked testing samples are from different domains, more and more researchers begin to attach importance to transfer learning [4, 5]. The basic idea of transfer learning is that although the different data distribution in the source domain and the target domain exists, some associated domains can share some of the same knowledge structure, which can be used as a bridge of knowledge transformation from the source domain to the target domain. The existing methods usually look for these common structures through the optimization of predefined objective functions, including the maximum of empirical likelihood and the preservation of geometric structure. For example, from the perspective of empirical likelihood, Dai et al. proposed the coclustering model based on classification [6], which, through the source domain data, adds constraints to set of words to provide classification structure and part of classification information, and coclustering was regarded as a bridge, through which the classification structure and information are transformed from the source domain to the target domain. The shortcoming of the method is considering the same concept in the document only, so Zhuang et al. studied the relationship between set of words and document classification; some invariant factors were regarded as the bridge of information transformation from the source domain to the target domain by trifactorization [7], but the method only considers

the similar concepts in the text. With the similar idea, Wang et al. carried it out in the network data classification of different domains [8]. In consideration of the shortcoming of the above methods, a joint model of similar concepts and the same concepts was built [9], with synchronized learning of boundary condition distribution. Then, taking the unique concept in each domain document into consideration, Zhuang et al. studied with synchronization the sharing concept and the unique concept in all domains by matrix trifactorization [10], which was more flexible in the process of data-fitting so as to get a better recognition rate. From the geometric point of view, if the two sample data in the domain are similar in terms of the essence of data distribution and the geometric structure, their markers should also be similar [11]. To retain the essential structure, Ling et al. explored the consistency between the supervision of source domain and the essential structure of target domain through the spectrum learning [12]. And Pan et al. put forward the conversion component analysis [13], which aims to find a set of common conversion components in the two domains. The samples are projected into the subspace, which makes the reduction of different degrees of data distribution in different domains. With the same idea, Wang and Mahadevan presented the projection of different domains to the new potential space, synchronously matching the corresponding samples and preserving the geometric structure of each domain [14]. Accounting for the two views, a graph canonical transformation learning (GTL) [15] was put forward, which retained some geometric structures based on the maximum of empirical likelihood.

Based on study of joint canonical transformation learning and considering the fact that the L1-Graph [16, 17] has better adaptive ability and better stability compared with k neighbor graph, we put forward a multiview classification algorithm based on L1-Graph domain adaptation learning, which is based on Long et al. [15]. The idea is to construct a framework of nonnegative matrix trifactorization based on transfer learning, in which the unchanged information is regarded as the bridge of knowledge transformation from the source domain to the target domain, and the next step is to construct L1-Graph on the basis of sparse representation, so as to search for the nearest neighbor data with self-adaptation and preserve the samples and the geometric structure; finally, the paper uses the iterative algorithm to cope with the optimization issue of unified objective functions, and then the classification of the testing samples is completed.

2. The Multidomain and Multiview Sample Classification Based on L1-Graph Domain Adaptation Learning

2.1. Description of L1-Graph Domain Adaptation Learning. L1-Graph domain adaptation learning can be applied to all different domains, but, for purposes of explanation, two domains are presented here: the source domain D_s and the target domain D_t, and the domain index is expressed as $\ell = \{s, t\}$. Each domain D_π, $\pi \in \ell$, has a characteristic matrix X_π. In order to find the common structure, X_π will be broken down into three nonnegative matrices; that is, $X_\pi = U_\pi H V_\pi^T$, among which $U_\pi = [u_{*1}^\pi, \ldots, u_{*k}^\pi]$, $V_\pi = [v_{*1}^\pi, \ldots, v_{*c}^\pi]$, and a_{*i} is the i column of the matrix A. In the domain D_π, the characteristic samples of matrix trifactorization are classified through the maximum of empirical likelihood; H represents the relationship between the characteristic set U_π and the sample classification V_π, and it can be regarded as a bridge of knowledge transformation because of the cross domain stability. In addition, the use of sparse representation is carried out to construct graphs and diagrams G_π^u and G_π^v to represent the geometry information of the characteristic space and the sample space in domain D_π, respectively. The basic idea of L1-Graph domain adaptation learning algorithm is presented in [15]. In general, similar characteristics represent the same meaning; likewise, similar samples have the same marker. These two graphs are regarded as joint regularization function so that the trifactorization model of learning successfully forecasts the sample marker in the case of retaining the essence of geometric structure, and the geometric information will be effectively integrated into the clustering process, thus ensuring that the common structure information H can effectively promote transfer learning.

2.2. Model of Matrix Trifactorization. Assuming that the source domain is D_s and the target domain is D_t, the domain index is expressed as $\ell = \{s, t\}$. D_s and D_t share the same characteristic space and marker space, including m characteristics and c categories in each characteristic space. $X_\pi = [x_{*1}^\pi, \ldots, x_{*n_\pi}^\pi] \in R^{m \times n_\pi}$, $\pi \in \ell$, represents the characteristic sample matrix in domain D_π, in which x_{*i}^π indicates the i column of the domain D_π. $Y_\pi \in R^{n_\pi \times c}$ represents the sample marker of the source domain, as x_{*i}^π belongs to category j, $y_{ij}^\pi = 1$; otherwise, $y_{ij}^\pi = 0$.

A total of structural information exists in every associated domain D_π; for the same structure information, the nonnegative matrix trifactorization is conducted as for the characteristic sample matrix X_π, $\pi \in \ell$, and the optimization is planned as follows:

$$\min_{U_\pi, H_\pi, V_\pi \geq 0} L_\pi = \left\| X_\pi - U_\pi H_\pi V_\pi^T \right\|^2 \tag{1}$$

in which $\|A\|$ is the Frobenius norm of matrix A; $U_\pi = [u_{*1}^\pi, \ldots, u_{*k}^\pi] \in R^{m \times k}$, and each u_{*i}^π represents semantic concepts, namely, characteristic clustering; $V_\pi = [v_{*i}^\pi, \ldots, v_{*c}^\pi] \in R^{n_\pi \times c}$, and each v_{*i}^π represents a sample type. U_π and V_π are the clustering results as to characteristic and sample separately. $H_\pi \in R^{k \times c}$ represents the relationship between the characteristic set U_π and the sample classification V_π, which can keep better stability when compared with U_π and V_π in different domains. Therefore, assuming that $H_\pi = H$ adapts to each domain, then the collective matrix trifactorization is planned as follows:

$$\min_{U_\pi, H, V_\pi \geq 0} L = \sum_{\pi \in \ell} \left\| X_\pi - U_\pi H V_\pi^T \right\|^2. \tag{2}$$

The common structural information H, as the stable bridge of knowledge transformation, can be executed as the monitor information in the source domain, namely, executing $V_s \equiv Y_s$. Through the bridge, the knowledge marker

Y_s in source domain can be converted to the sample V_t in target domain, and the process is in correspondence to the maximum of multidomain empirical likelihood.

2.3. Sparse Representation of L1-Graph Structure. From the geometric point of view, it can be thought that the data points are sampled in the distribution, which is formed by a low dimensional manifold, and then embedded into a high dimensional data space. Therefore, in order to avoid the change of the essence of the data distribution and to hope that the geometric structure is preserved in the conversion process, the assumption is presented here; if the inherent geometric structure of data distribution of two samples x_{*i}^{π} and x_{*j}^{π} in domain D_π is close to each other, the markers of v_{*i}^{π} and v_{*j}^{π} should also be close to each other, so a model can be formed on the geometric structure of sample space. The traditional graph construction methods are often heavily dependent on the choice of parameters, and it is difficult to effectively reflect the complexity of the data distribution. According to the theory of sparse representation, any sample can be linearly reconstructed by other samples (allowing for some reconstruction error), and the sparse reconstruction coefficient of the sample can be obtained by coping with a L1 norm optimization issue. The reconstructed coefficients, as weights between two samples, can adjust the relationship between samples with self-adaption, so that the sparse graph, which represents the local relationship between samples, contains more useful structural information. From the perspective of duality between characteristic and sample, characteristic is also sampled from the distribution, which is supported by a low dimensional manifold, and then embedded into a high dimensional space. If the geometric structure of data distribution of two characteristics x_{i*}^{π} and x_{j*}^{π} in domain D_π is close to each other, their characteristic sets u_{i*}^{π} and u_{j*}^{π} should also be close to each other. Therefore, as for the characteristic space, a sparse graph based on the principle of sparse representation can be constructed to retain the characteristic geometric structure in each domain, just the same as retaining the geometric structure of sample space. The sparse graph construction of sample here is expressed as G_π^v, the corresponding sparse graph construction of characteristic is expressed as G_π^u, and the sparse graph construction of sample space is presented in the following steps.

(1) Input. Sample $X_\pi = [x_{*1}^{\pi}, \ldots, x_{*n_\pi}^{\pi}] \in R^{m \times n_\pi}$, $\pi \in \ell$; each sample $x_{*i}^{\pi} \in R^m$ is normalized, so that $\|x_{*i}^{\pi}\|_2 = 1$.

(2) To Solve the Reconstruction Coefficient. The sparse reconstruction coefficient for each sample x_{*i}^{π} in each domain can be obtained by coping with the following optimization issue of minimization of L1 paradigm:

$$\begin{aligned} &\min_{\alpha_i^{\pi}} \ \left\|\alpha_{*i}^{\pi}\right\|_1, \\ &\text{s.t.} \quad x_{*i}^{\pi} = B_i^{\pi} \alpha_{*i}^{\pi} \end{aligned} \tag{3}$$

in which $B_i^{\pi} = [x_{*,1}^{\pi}, \ldots, x_{*,i-1}^{\pi}, x_{*,i+1}^{\pi}, \ldots, x_{*,n_\pi}^{\pi}, I] \in R^{m \times (n_\pi - 1 + m)}$ is an overcomplete dictionary and $\alpha_{*i}^{\pi} \in R^{n_\pi - 1 + m}$ is a column vector which constitutes the reconstruction coefficient, indicating a relationship between sample x_{*i}^{π} and other samples.

(3) To Set the Edge Weights of L1-Graph. L1-Graph is expressed as $G_\pi^v = \{X_\pi, W_\pi^v\}$, in which X_π represents the set of all nodes in domain π and W_π^v represents the weight matrix of L1-Graph in the domain, that is, the similarity matrix. When $i > j$, $(W_\pi^v)_{ij} = |\alpha_{ij}^{\pi}|$; when $i < j$, $(W_\pi^v)_{ij} = |\alpha_{i(j-1)}^{\pi}|$; when $i = j$, $(W_\pi^v)_{ij} = 0$. The number of nearest neighbors of each sample is identified by optimizing the L1 paradigm issue instead of parameters, which are manually set up.

(4) The Similarity of Symmetrization Matrix. Consider $W_\pi^v = (W_\pi^v + (W_\pi^v)^T)/2$.

Similarly, the weights matrix of characteristic space W_π^u can also be obtained.

To preserve the L1-Graph regularization function of the geometric structure minimization sample in domain D_π,

$$\begin{aligned} R_\pi^v &= \frac{1}{2} \sum_{ij} \left\| v_{i*}^{\pi} - v_{j*}^{\pi} \right\|^2 (W_\pi^v)_{ij} \\ &= \sum_i v_{i*}^{\pi} (v_{i*}^{\pi})^T (W_\pi^v)_{ii} - \sum_{ij} v_{i*}^{\pi} (v_{j*}^{\pi})^T (W_\pi^v)_{ij} \\ &= \operatorname{tr}\left(V_\pi^T (D_\pi^v - W_\pi^v) V_\pi\right). \end{aligned} \tag{4}$$

To further preserve the L1-Graph regularization function of the geometric structure synchronization minimization characteristic in domain D_π,

$$R_\pi^u = \frac{1}{2} \sum_{ij} \left\| u_{i*}^{\pi} - u_{j*}^{\pi} \right\|^2 (W_\pi^u)_{ij} = \operatorname{tr}\left(U_\pi^T (D_\pi^u - W_\pi^u) U_\pi\right). \tag{5}$$

2.4. Joint Optimization. Evidently, (4) and (5) define that the geometric structure of sample and characteristic can be retained through the L1-Graph regularization function. Therefore, the two equations, as a combined L1-Graph regularization function, can be integrated into (2), which defines the optimization issue of L1-Graph coregularized collective matrix trifactorization (L1-GCMF) as follows:

$$\begin{aligned} \min_{U_\pi, H_\pi, V_\pi \geq 0} \quad & O = L + \sum_{\pi \in \ell} \left(\lambda R_\pi^u + \gamma R_\pi^v\right) \\ \text{s.t.} \quad & U_\pi^T 1_m = 1_k, \\ & V_\pi^T 1_{n_\pi} = 1_c, \\ & \forall \pi \in \ell \end{aligned} \tag{6}$$

in which, λ and γ are regularization parameters; the optimization issue can be better defined by constraining the norm ℓ_1 of each column of U_π and V_π. According to the optimization

results, the marking of a random sample x_{*i}^{π} in target domain D_{π} can be easily inferred by the following formula:

$$f\left(x_{*i}^{\pi}\right) = \arg\max_{j} \left(V_{\pi}\right)_{ij}. \tag{7}$$

In the process of optimization of formula (6), the common structure information H, which is found through the synchronous maximum of empirical likelihood and preserving the geometric structure, becomes more smooth in the process of conversion learning. L1-GCMF can be extended to handle multidomain issues and to research the commonality in the collection structure. The approaches to the optimization issue in (6) can be derived based on the constrained optimization theory. Specifically, the updating rules are deduced through fixing the rest of variables and optimizing one variable, and the process is repeated till the convergence. In consideration of the nonnegative and the constraint of norm ℓ_1, a Lagrange function is constructed as follows:

$$\begin{aligned}\Phi = O &+ \sum_{\pi\in\ell} \operatorname{tr}\left(\Lambda_{\pi}\left(U_{\pi}^{T}1_{m} - 1_{k}\right)\left(U_{\pi}^{T}1_{m} - 1_{k}\right)^{T}\right) \\ &+ \sum_{\pi\in\ell} \operatorname{tr}\left(\Gamma_{\pi}\left(V_{\pi}^{T}1_{n_{\pi}} - 1_{c}\right)\left(V_{\pi}^{T}1_{n_{\pi}} - 1_{c}\right)^{T}\right)\end{aligned} \tag{8}$$

in which $\Lambda_{\pi} \in R^{k\times k}$ and $\Gamma_{\pi} \in R^{c\times c}$ are constrained Lagrange multipliers and 1_m is 1 vector. By using the complementary conditions of Karush-Kuhn-Tucker (KKT), the constraint conditions of U_{π} are deduced as follows:

$$\begin{aligned}\frac{1}{2}\nabla_{U_{\pi}}\Phi \circ U_{\pi} &= \left(V_{\pi}H^{T}U_{\pi}^{T}U_{\pi}H - X_{\pi}V_{\pi}H^{T}\right)\circ U_{\pi} \\ &+ \left(\lambda D_{\pi}^{u}U_{\pi} - \lambda W_{\pi}^{u}U_{\pi} + 1_{m}1_{m}^{T}D_{\pi}^{v}\Lambda_{\pi}\right. \\ &\left. - 1_{m}1_{k}^{T}\Lambda_{\pi}\right)\circ U_{\pi} = 0.\end{aligned} \tag{9}$$

According to the KKT conditions, the updating plan is obtained as follows:

$$U_{\pi} \longleftarrow U_{\pi} \circ \sqrt{\frac{\left[X_{\pi}V_{\pi}H^{T} + \lambda W_{\pi}^{u}U_{\pi} + 1_{m}1_{k}^{T}\Lambda_{\pi}\right]}{\left[U_{\pi}HV_{\pi}^{T}V_{\pi}H^{T} + \lambda D_{\pi}^{u}U_{\pi} + 1_{m}1_{m}^{T}D_{\pi}^{v}\Lambda_{\pi}\right]}}. \tag{10}$$

The computation of Λ_{π} can be avoided through the use of an iterative standardization technology. In the process of each iteration, each column of U_{π} can be normalized, so that $U_{\pi}^{T}1_{m} = 1_{k}$. Then, two equal items, depending on Λ_{π}, can be obtained; namely, $1_{m}1_{m}^{T}D_{\pi}^{v}\Lambda_{\pi} = 1_{m}1_{k}^{T}\Lambda_{\pi}$, which can be omitted in the updating plan with no influence on the convergence; therefore, the updating rules are presented as follows:

$$U_{\pi} \longleftarrow U_{\pi} \circ \sqrt{\frac{\left[X_{\pi}V_{\pi}H^{T} + \lambda W_{\pi}^{u}U_{\pi}\right]}{\left[U_{\pi}HV_{\pi}^{T}V_{\pi}H^{T} + \lambda D_{\pi}^{u}U_{\pi}\right]}}. \tag{11}$$

Similarly, the updating rules can also be obtained as follows:

$$\begin{aligned}V_{\pi} &\longleftarrow V_{\pi} \circ \sqrt{\frac{\left[X_{\pi}^{T}U_{\pi}H + \gamma W_{\pi}^{v}V_{\pi}\right]}{\left[V_{\pi}H^{T}U_{\pi}^{T}U_{\pi}H + \gamma D_{\pi}^{v}V_{\pi}\right]}}, \\ H &\longleftarrow H \circ \sqrt{\frac{\left[\sum_{\pi\in\ell}U_{\pi}^{T}X_{\pi}V_{\pi}\right]}{\left[\sum_{\pi\in\ell}U_{\pi}^{T}U_{\pi}HV_{\pi}^{T}V_{\pi}\right]}}\end{aligned} \tag{12}$$

in which $\circ$ represents the element product, $[\cdot]/[\cdot]$ the element division, and $\sqrt{\cdot}$ the square root of elements.

2.5. Description of a Multiview Sample Classification Algorithm Based on L1-Graph Domain Adaptation Learning. See Algorithm 1.

3. Experiment Results and Analysis

3.1. Comparative Experiment Results Based on USPS-Binary Digital Database. The experimental samples are selected from the two digital databases: USPS and Binary. The USPS database contains 10 groups of handwritten digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9, as shown in Figure 1(a), and each group contains 1100 gray scale samples. Some gray scale samples of one number are shown in Figure 1(b). The Binary database contains a total of 36 kinds of data, including the numbers 0–9 and the letters A–Z, and each kind of data includes 39 samples; only 10 groups of handwritten digits 0–9 are selected in the experiment. Figure 2(a) presents 10 categories of digits, and Figure 2(b) presents a part of two binary samples of one category.

All images are cut into the same dimension, SIFT characteristic extraction is made for each image, and then all the pictures are transferred into the 300-dimension characteristic histogram. The 10 categories from two databases were considered as the known ones; USPS database was treated as the source domain and Binary database the target domain; the data with the number $m = \{50, 100, 200\}$ were randomly selected from each category of data in USPS database to constitute the training samples, and 39 samples from each category of data in Binary database were treated as the observation samples. In order to obtain the optimal parameter values, we conducted the experiment 10 times to calculate the average recognition rate. The GCMF method obtains the optimal value when $\lambda = 0.1$, $\gamma = 1$, and the L1-GCMF method obtains the optimal value when $\lambda = 0.3$ and $\gamma = 1$. In all experiments, the number of k neighbor unified values is 10, and the number of iterations is 100. In view of the different values of m, 10 training samples were randomly selected from each group of data in USPS database to carry out the experiment, and Figure 3 showed the average recognition rate in correspondence with different training samples and different methods.

The experimental results from Figure 3 show that, for different number of training samples, the recognition rate of L1-GCMF algorithm is higher than that of GCMF [15] and ITML [18] algorithms, and, with the changing number

Input: data set $\{X_\pi\}_{\pi\in\ell}$, Y_s, parameters λ, γ, max $Iter$.
Output: the target domain classification results V_t
Start:
to construct the graph G_π^v, G_π^u through the principle of sparse representation; to normalize the data sets through $X_\pi \leftarrow X_\pi/\|X_\pi\|, \forall\pi \in \ell$.
to initialize $\{U_\pi\}_{\pi\in\ell}$, to generate H randomly, to obtain V_s through Y_s, to obtain V_t through the training of logistic regression in $\{X_s, Y_s\}$.
for $iter \leftarrow 1$ to max $Iter$ do
foreach $\pi \in \ell$ do
to update U_π, V_π and H through (11)~(12)
to fix $V_s \equiv Y_s$
to make the norm ℓ_1 for each column of U_π and V_π.
to calculate the objective function O^{iter} through (6)
End

Algorithm 1: L1-GCMF multiview sample classification algorithm.

(a) Ten categories of USPS database (b) Gray scale samples within one category

Figure 1: Samples of USPS database.

(a) Ten categories of Binary database (b) Binary samples within one category

Figure 2: Samples of Binary database.

of training samples, L1-GCMF algorithm is more stable compared with the other two algorithms. Due to the use of L1-Graph, the method can automatically search for the nearest neighbor parameters, which is more conducive to the connection to the same type samples, so as to improve the classification accuracy of multiview samples and make it have stronger stability.

3.2. Comparative Experiment Results Based on Three-Domain Object Benchmark Database. The Three-Domain Object Benchmark database [19] contains three different domains of amazon, dslr, and webcam, and each domain contains a total of 4652 images from 31 different object categories. There are about 90 images for each category in the domain of amazon, some of which from one category are displayed in Figure 4(a); there are 30 images for each category in domains of dslr and webcam, some of which are separately shown in Figures 4(b) and 4(c).

With amazon as the source domain and webcam as the target domain, 20 categories from the two databases were selected for experiments, all of the selected category of images will be converted to SIFT characteristics. Then 10 samples were randomly selected as training samples from the source domain, and 5 samples as testing samples from the target domain. The GCMF method obtains the optimal value when $\lambda = 0.2$, $\gamma = 1$, and the L1-GCMF method obtains the optimal value when $\lambda = 0.1$ and $\gamma = 2$. A random selection of 10 training samples and testing samples was used for experiments, and the average recognition rate and standard deviation are shown in Table 1.

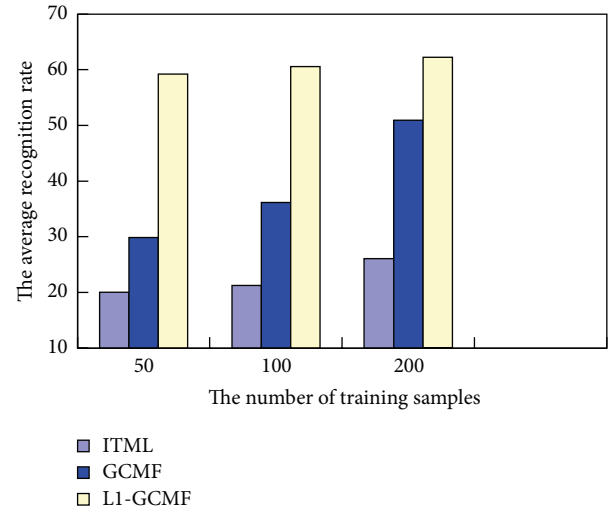


Figure 3: Average correctness recognition rate in USPS-VS-Binary.

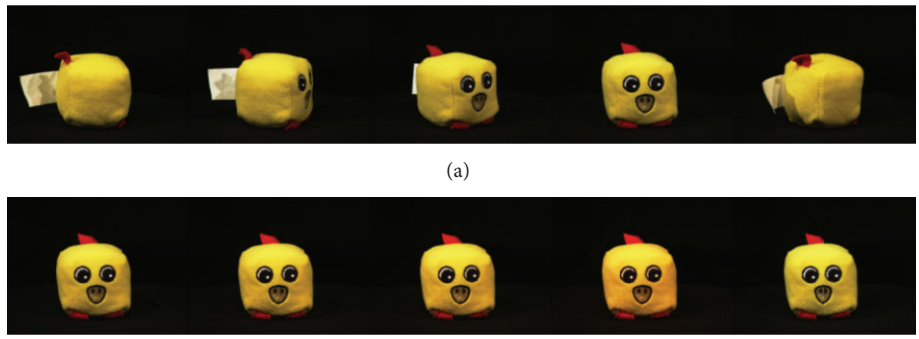
The experimental results from Table 1 show that the average recognition rate of L1-GCMF algorithm is relatively

(a)

(b)

(c)

Figure 4: (a) Samples of amazon domain; (b) samples of dslr domain; (c) samples of webcam domain.



(a)

(b)

Figure 5: (a) Samples of different angles; (b) samples of different light.

Table 1: Average correctness recognition rate in Three-Domain Object Benchmark database.

Algorithm	ITML [19]	GCMF [15]	L1-GCMF
Recognition rate (%)	10.2	24.9	50.55
Standard deviation	4.37	5.97	6.76

high, compared with information-theoretic metric learning (ITML) and GCMF algorithms. Different from the fact that the selection of the neighboring number of ITML and GCMF classification algorithms is dependent on the parameters, the algorithm presented in this paper can get higher recognition rate by using sparse principle of composition, which does not need to manually set up the parameters.

3.3. Comparative Experiment Results Based on ALOI Database. This set of experimental data is from the ALOI database. The database contains 1000 objects, and each object contains images with different light and different angles. 50 object images were selected for experiments, and all of the selected object images were transferred into the 800-dimension SIFT characteristics. Images under different angles were treated as the source domain, as shown in Figure 5(a), and images under different light as the target domain, as shown in Figure 5(b).

30 data were randomly selected from each group in the source domain to be treated as the training samples, and 10 images were randomly selected from each group in the target domain as the observation samples; the data of each group were used to carry out 10 random experiments, so data in the results are the mean of 500 random tests of the 50 categories; the parameter values are the same as in experiment of Section 3.1, and the average recognition rate and the standard deviation are shown in Table 2.

Experimental results from Table 2 show that the average recognition rate of this method is higher than that of ITML and GCMF algorithms, and the robustness of the method is

Table 2: Average correctness recognition rate in ALOI database.

Algorithm	ITML [18]	GCMF [15]	L1-GCMF
Recognition rate (%)	14.95	37.3	55.88
Standard deviation	8.66	5.66	9.41

presented because of the smaller standard deviation than the other two methods. Compared with the other algorithms, the algorithm has strong adaptability, avoids the artificial setting of the parameters, and achieves better multiview sample classification, which explains the rationality of the algorithm using sparse composition.

4. Conclusion

Based on the research of marked and unmarked data from different distribution, the paper proposed the multiview sample classification algorithm based on L1-Graph domain adaptation learning. The method first builds a nonnegative matrix trifactorization framework, in which the common structural information was regarded as the bridge of knowledge transformation from the source domain to the target domain, and then constructs L1-Graph by using the principle of sparse representation, so that the neighboring numbers of each sample are determined by the optimization of L1 paradigm issue, so as to search for the nearest neighbor data with self-adaptation. The next is to jointly optimize the objective function by using the iterative algorithm, and then the estimation of classification of the testing samples is completed. Comparative experimental results based on USPS-Binary digital database, Three-Domain Object Benchmark database, and ALOI database show the rationality of the method. However, compared with the traditional k nearest neighbor composition, the composition method based on sparse representation has relatively high complexity and relatively large computational cost, due to the need to calculate the reconstruction coefficients of all samples.

Conflict of Interests

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

References

[1] H. Daumé III, "Frustratingly easy domain adaptation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL '07)*, pp. 256–263, June 2007.

[2] T. Li, V. Sindhwani, C. Ding, and Y. Zhang, "Knowledge transformation for cross-domain sentiment classification," in *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)*, pp. 716–717, July 2009.

[3] Y. Luo, D. Tao, B. Geng, C. Xu, and S. J. Maybank, "Manifold regularized multitask learning for semi-supervised multilabel image classification," *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 523–536, 2013.

[4] Y. Luo, T. Liu, D. Tao, and C. Xu, "Decomposition-based transfer distance metric learning for image classification," *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 3789–3801, 2014.

[5] S. Si, D. Tao, and B. Geng, "Bregman divergence-based regularization for transfer subspace learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 7, pp. 929–942, 2010.

[6] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Co-clustering based classification for out-of-domain documents," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 210–219, 2007.

[7] F. Zhuang, P. Luo, H. Xiong, Q. He, Y. Xiong, and Z. Shi, "Exploiting associations between word clusters and document classes for cross-domain text categorization," *Statistical Analysis and Data Mining*, vol. 4, no. 1, pp. 100–114, 2011.

[8] H. Wang, H. Huang, F. Nie, and C. Ding, "Cross-language web page classification via dual knowledge transfer using nonnegative matrix tri-factorization," in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '11)*, pp. 933–942, July 2011.

[9] M. Long, J. Wang, G. Ding, W. Cheng, X. Zhang, and W. Wang, "Dual transfer learning," in *Proceedings of the 12th SIAM International Conference on Data Mining*, pp. 540–551, 2012.

[10] F. Zhuang, P. Luo, C. Du, Q. He, and Z. Shi, "Triplex transfer learning: exploiting both shared and distinct concepts for text classification," in *Proceedings of the 6th ACM International Conference on Web Search and Data Mining (WSDM '13)*, pp. 425–434, it, February 2013.

[11] D. Cai, X. He, X. Wang, H. Bao, and J. Han, "Locality preserving nonnegative matrix factorization," in *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pp. 1010–1015, July 2009.

[12] X. Ling, W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Spectral domain-transfer learning," in *Proceedings of the 14th ACMKDD International Conference on Knowledge Discovery and Data Mining*, pp. 488–496, 2008.

[13] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[14] C. Wang and S. Mahadevan, "Heterogeneous domain adaptation using manifold alignment," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 1541–1546, July 2011.

[15] M. Long, J. Wang, G. Ding, D. Shen, and Q. Yang, "Transfer learning with graph regularization," in *Proceedings of the 26th AAAI Conference on Artificial Intelligence and the 24th Innovative Applications of Artificial Intelligence Conference*, pp. 1033–1039, July 2012.

[16] B. Cheng, J. Yang, B. Yan, and Y. Fu, "Learning with l^1-graph for image analysis," *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 858–866, 2010.

[17] Y. Fang, R. Wang, B. Dai, and X. Wu, "Graph-based learning via auto-grouped sparse regularization and kernelized extension," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 1, pp. 142–154, 2015.

[18] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*, pp. 209–216, June 2007.

[19] S. Kate, K. Brian, F. Mario, and D. Trevor, "Adapting visual category models to new domains," in *Proceedings of the 11th European Conference on Computer Vision, Part IV, Heraklion, Greece, September 2010*, vol. 6314, pp. 213–226, Springer, 2010.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 754582, 8 pages
http://dx.doi.org/10.1155/2015/754582

Research Article

Opposition-Based Improved PSO for Optimal Reactive Power Dispatch and Voltage Control

Shengrang Cao,[1,2] Xiaoqun Ding,[1] Qingyan Wang,[1] and Bingyan Chen[1,3,4]

[1] *College of Energy and Electrical Engineering, Hohai University, Nanjing 211100, China*
[2] *Jiangsu Union Technical Institute, Nanjing Branch, Nanjing 210019, China*
[3] *Hohai University, Nantong Institute of Marine and Offshore Engineering, Nantong 226000, China*
[4] *Jiangsu Province Key Laboratory of Environmental Engineering, Nanjing 210000, China*

Correspondence should be addressed to Shengrang Cao; csrhhu@foxmail.com

Received 11 September 2014; Revised 31 December 2014; Accepted 3 March 2015

Academic Editor: Pandian Vasant

An opposition-based improved particle swarm optimization algorithm (OIPSO) is presented for solving multiobjective reactive power optimization problem. OIPSO uses the opposition learning to improve search efficiency, adopts inertia weight factors to balance global and local exploration, and takes crossover and mutation and neighborhood model strategy to enhance population diversity. Then, a new multiobjective model is built, which includes system network loss, voltage dissatisfaction, and switching operation. Based on the market cost prices, objective functions are converted to least-cost model. In modeling process, switching operation cost is described according to the life cycle cost of transformer, and voltage dissatisfaction penalty is developed considering different voltage quality requirements of customers. The experiment is done on the new mathematical model. Through the simulation of IEEE 30-, 118-bus power systems, the results prove that OIPSO is more efficient to solve reactive power optimization problems and the model is more accurate to reflect the real power system operation.

1. Introduction

Particle swarm optimization (PSO) proposed by Dr. Eberhart and Dr. Kennedy in 1995 is a population-based evolutionary algorithm that emulates the social behavior of bird flocks in an attempt to optimally explore some given problem space. In the past decade, PSO has been studied and applied in many research and application areas. However, many experiments have shown that PSO converges too fast and easily falls into local optima especially when solving complex multimodal problems [1]. Many researchers have put forward improved programs. In general, there are three kinds of improved methods, that is, parameter factor [2, 3], neighborhood topology [4], and hybridization with other algorithms [5, 6]. Each of them has its own advantages but also has defects. The hybridization of PSO with other algorithms has been proved to be a promising technique. There are two ways to combine. One is with evolutionary algorithms, for instance, [5] is PSO combined with genetic algorithm, in which genetic algorithm was used to increase exploration ability of particles, [6] is the combination hybridized PSO with ant algorithm, and uses the characteristic of positive feedback to enhance the search ability of particles. The other is with nonevolutionary algorithms, such as hybridizing PSO with Filter [7], Cauchy mutation [8], and opposition-based learning [9], in which opposition-based learning has a good character; it can accelerate the convergence speed of the algorithm, and this concept was first proposed by Dr. Tizhoosh. When we seek the solution in a direction, it is beneficial to consider the opposite direction [10]. By comparison of fitness values of the two directions, the optimal candidate solution is chosen from them. For this reason, many algorithms were enhanced by opposition-based learning [11, 12].

In this paper, a new improved particle swarm optimization algorithm OIPSO is proposed, which makes some expansion and correction by inertia weight factors, crossover and mutation, neighborhood model, and opposition-based learning. Through these strategies, the new OIPSO algorithm can enhance population diversity and avoid premature convergence and stagnation.

The improved algorithm would be developed to multiobjective reactive power optimization. It is known that the purpose of power system reactive optimization is to find the best compensation methods to make the power system safe and economic under the demand of reactive load power system [13]. For that, we establish a mathematical model, including system network loss, voltage dissatisfaction, and switching operator. In the modeling process, introduce life cycle cost of the transformer, define switching operator cost, and also give descriptions of the customers' loss, voltage quality requirement, and dissatisfaction degree and then reactive power optimization tests through IEEE 30- and 118-bus systems.

The remaining of this paper is organized as follows: Section 2 provides an overview of PSO algorithm, opposition-based learning, and other improvement strategies. In Section 3, the multiobjective model of reactive power system is established. Combined with reactive power optimization, main steps of the improved OIPSO algorithm are given in Section 4. Then, the algorithm is applied to IEEE 30- and 118-bus systems in Section 5. Finally, Section 6 concludes the paper with a summary.

2. PSO Algorithm and Its Improvement

The essence of PSO algorithm is that by letting the information about good solutions spread out through the swarm, the particles would tend to move to good areas [14, 15]. At each iteration time t, particle i is moved to a new position by adding a velocity term to its current position according to formulas (1) and (2) as follows:

$$X_i(t+1) = X_i(t) + V_i(t+1), \tag{1}$$

$$\begin{aligned} V_i(t+1) = {} & \omega V_i(t) + c_1 \mathrm{rand}_1 \left(P_{\mathrm{best},i} - X_i(t)\right) \\ & + c_2 \mathrm{rand}_2 \left(G_{\mathrm{best}} - X_i(t)\right), \end{aligned} \tag{2}$$

where $i = 1, 2, \ldots, \mathrm{pop}$, pop is the size of swarm particles, $P_{\mathrm{best},i}$ is the best position of particle i, G_{best} is the global best position, ω is called inertia weight, and c_1 and c_2 are acceleration factors.

PSO algorithm converges fast, but it is also easy to fall into the local optimal. The reason is that, in the optimal process, all particles consider the global best position G_{best} as the goal and search directions run towards it. This situation worsens in the later process and leads to reduction of the ability to explore unknown area. Therefore, it is very important to enhance the diversity of particles. This paper makes some expansion and modification to the basic particle swarm algorithm. The main improvement measures are as follows.

2.1. Opposition-Based Learning. In PSO algorithm, starting points are given randomly. If the starting points are close to the optimal point, convergence speed would be faster. The opposite operation used in the selection of the starting points has proven that utilizing opposition in learning yields more efficient algorithms than using only pure randomness [10, 16]. Besides the opposition point used in the initial population, this paper would intersectionally use the opposition point and the following crossover and mutation to increase the diversity of particles. The definition of opposition is given as follows.

Definition 1 (opposite point). Let $X = (x_1, x_2, \ldots, x_n)$ be a point in n-dimensional space and let $x_i \in [a_i, b_i]$ be a real number, where $i = 1, 2, \ldots, n$. The opposite point $X' = (x_1', x_2', \ldots, x_n')$ is defined by its components $x_i' = a_i + b_i - x_i$.

2.2. Inertia Weight Factor. Generally, for initial stage of search process, large inertia weight can enhance the global exploration and, for last stage, the small inertia weight is good for local exploration, so time-varying inertia weight, which typically decreases linearly from about 0.9 to 0.4 [17], can balance the global and local search; specific measures are as follows:

$$\omega = \omega_{\max} - \frac{t}{t_{\max}} \left(\omega_{\max} - \omega_{\min}\right), \tag{3}$$

where $\omega_{\max}$ and $\omega_{\min}$ are the maximum and minimum of ω, $t_{\max}$ is the maximum iterating times, and t is the current iteration.

2.3. Neighborhood Exchange. In particle swarm algorithm, each particle generally learns from its own best position and global best position. In social cognitive system, an individual, besides its own experience and excellent information obtained from the whole society, should exchange with other better individuals to improve itself [13, 18]. Based on this idea, PSO algorithm is modified as follows:

$$\begin{aligned} V_i(t+1) = {} & \omega V_i(t) + c_1 \mathrm{rand}_1 \left(P_{\mathrm{best},i} - X_i(t)\right) \\ & + c_2 \mathrm{rand}_2 \left(G_{\mathrm{best}} - X_i(t)\right) \\ & + c_3 \mathrm{rand}_3 \left(P_{\mathrm{best},n} - X_i(t)\right), \end{aligned} \tag{4}$$

where $P_{\mathrm{best},n}$ is position vector of the better individual in domain and c_3 is an accelerating constant. Larger c_3 can increase the ability of the individual to explore unknown better area, especially in the later process. In contrast, the set value of c_2 is related to convergence rate; based on their own characteristics, parameters c_2, c_3 are improved as follows:

$$\begin{aligned} c_3 &= c_{3\min} + \frac{t}{t_{\max}} \left(c_{3\max} - c_{3\min}\right), \\ c_2 &= c_{2\max} - \frac{t}{t_{\max}} \left(c_{2\max} - c_{2\min}\right), \end{aligned} \tag{5}$$

where $c_{2\max}$ and $c_{2\min}$ are the maximum and minimum of c_2 and $c_{3\max}$ and $c_{3\min}$ are the maximum and minimum of c_3.

2.4. Crossover and Mutation. Crossover and mutation operators are key technologies in genetic algorithm, which are used to enhance the diversity of the species. The particles of PSO

algorithm can be crossed and varied [13, 18]. For particle i, the crossover operation is described as follows:

$$V_i(t+1) = \frac{V_i(t) + V_n(t)}{\left\|V_i(t) + V_n(t)\right\|} \left\|V_i(t)\right\|,$$
$$X_i(t+1) = rX_i(t) + (1-r)X_n(t), \tag{6}$$

where r is a random variable in the range $[0, 1]$.

The method of mutation operator is

$$V_i(t+1) = \frac{V_i(t) + V_i(t+1)}{\left\|V_i(t) + V_i(t+1)\right\|} \left\|V_i(t)\right\|,$$
$$X_i(t+1) = \gamma X_i(t) + (1-r)\left(X_{i\max} - X_{i\min}\right), \tag{7}$$

where γ is a random coefficient in the range $[0, 1]$. $X_{i\max}$, $X_{i\min}$ are the maximum and minimum of X_i. A variable parameter P_m selected from $[0, 1]$ determines the mutation operator.

3. Mathematical Model of Power System Reactive Power Optimization

Objective function of reactive power optimization model is commonly the minimization of active power loss and ignores many security and economic factors. On security grounds, its optimal results often approach the upper limits of voltage level on load buses. Although the results satisfy equality and inequality constraints, it is not conducive to the safe operation of power system. In this paper, we introduce voltage dissatisfaction and use penalty to punish voltage dissatisfaction. And, in economic terms, transformer tap has mechanical life and the total number of switching times has a limit. Therefore, the switching operation cost should not be neglected. In this section, we present transformer operation cost by life cycle cost and use it as an objective function.

3.1. Active Power Loss. Reactive power is usually used as the objective function. Utilize the following equal to convert power loss into expense:

$$C_{P\text{loss}} = \lambda_1 \times P_{\text{loss}} = \lambda_1 \times \sum_{\substack{i\in N \\ j\in N_i}} G_{ij}\left(V_i^2 + V_j^2 - 2V_iV_j\cos\theta_{ij}\right), \tag{8}$$

where λ_1 is the unit price for system network loss and its value depends on market electricity prices, P_{loss} is system network loss, N is set of branches numbers, and N_i is the collection of nodes number associated with the node i (including i itself).

3.2. Voltage Dissatisfaction. The cost of voltage dissatisfaction describes the users' losses caused by overlimit voltages on the load buses. When the voltage amplitudes are far from the ideal range, the losses would become more heavy. Voltage dissatisfaction can be expressed by the following mathematical description [19]:

$$C_{\text{VVIO}} = \lambda_2 \times S_V = \lambda_2 \times \sum_{i=1}^{N_{\text{load}}} S_{Vi},$$
$$S_{Vi} = \begin{cases} a_i \times P_{Di} \times \left(V_i^{\min} - V_i\right)^{\theta} & V_i < V_i^{\min} \\ 0 & V_i^{\min} < V_i < V_i^{\max} \\ b_i \times P_{Di} \times \left(V_i - V_i^{\max}\right) & V_i^{\max} < V_i, \end{cases} \tag{9}$$

where λ_2 is the penalty for voltage dissatisfaction; S_V is the sum of nodal voltage dissatisfaction degree; P_{Di} is the active power load of node i; a_i, b_i are the voltage dissatisfaction degree when the voltage values overstep the lower or upper limit, which are declared by the relevant users; θ is the index parameter about the relationship of the losses and voltage deviation; we set $\theta = 1$ in this paper. In general, voltage range is $[0.94, 1.06]$, but some power consumption equipment needs high security and stability. It is obvious that the voltages are better to control in a satisfactory interval, instead of approaching upper limits, so we change voltage range into satisfactory interval. The satisfactory interval is set by users at different load buses. In this paper, we set satisfactory interval to be $[0.95, 1.05]$ if voltages satisfy equality and inequality constraints but overstep the satisfactory interval, and dissatisfactory penalty can be relatively small and can be set based on users' requirement of electricity quality.

3.3. Switching Operations. Transformer operation cost is mainly composed of construction, installation, equipment acquisition, maintenance, and so on. Define S_{Ti} as the change of the ratio at transformer i before and after optimization and it is described as

$$S_{Ti} = \frac{\left|T_i - T_{i0}\right|}{\Delta T_i}, \tag{10}$$

where T_i is the ratio of transformer i after optimization; T_{i0} is the ratio of transformer i before optimization; ΔT_i is the step of transformer i. After optimization, tap operation costs are

$$C_{\text{TC}} = \lambda_3 \times S_T = \lambda_3 \times \sum_{i=1}^{N_T} S_{Ti}, \tag{11}$$

where N_T is the total of transformers and λ_3 is the unit price of each action, which depends on the annual value of investment, operation, and maintenance cost, where each of them is described as follows.

3.3.1. Construction Cost. Construction cost of the transformer is composed of construction project cost, installation project cost, equipment acquisition cost, and other cost, called static investment [20, 21]. Generally, construction cost is expressed as

$$C_1 = V_1 \cdot \frac{\gamma\cdot(1+\gamma)^n}{(1+\gamma)^n - 1}, \tag{12}$$

where C_1 is the annual value of the lump-sum investment, V_1 is the lump-sum investment, and γ is the interest rate.

3.3.2. Scrap Value. Scrap value of the transformer can be divided into residual recovery income, early retirement loss cost, and disposal expenses. Its annual value is described as [19]

$$C_2 = -\rho \cdot C_1 + \frac{\left(n - n'\right)}{n} \cdot C_1 + \mathrm{dc} \cdot \frac{\gamma \cdot (1+\gamma)^n}{(1+\gamma)^n - 1}. \quad (13)$$

ρ is the recovery factor, and we set it as 20%~30%. The minus sign expresses that the income is negative relative to the expenditure; dc is the current value of the disposal expense; n is the actual operating life; n' is the expected operating life. When the actual operating life is greater than or equal to the expected life, early retirement loss cost is 0.

3.3.3. Maintenance Cost. The maintenance method adopts scheduling repair; its model is described as

$$V_3 = \sum_{t=1}^{30} M_e \cdot \left(1 - \exp\left(-\left(\frac{t}{15.98}\right)^{1.35}\right)\right), \quad (14)$$

where M_e is early operation and maintenance cost; t is the year number; and V_3 is the total cost of maintenance in 30 years. Use the uniform annual value method for conversion; maintenance cost of each year can be expressed as

$$C_3 = V_3 \cdot \frac{\gamma \cdot (1+\gamma)^n}{(1+\gamma)^n - 1}. \quad (15)$$

Through the above models, the life cycle cost (LLC) is defined as

$$\mathrm{LCC} = C_1 + C_2 + C_3. \quad (16)$$

Take the 110 kv conventional substation [18]. For example, set the expected life of the transformer n' to be 30 years. γ is 8%, the mechanical life of tap is 10000 times, and average annual switching operations are 333 times. For transformer tap, its unit price of one action in the tth year is

$$C_{t\_\mathrm{avg}} = \frac{C_1 + C_2 + C_3}{333}. \quad (17)$$

In this paper, we set $\lambda_3 = C_{1\_\mathrm{avg}}$.

3.4. Objective Function and Constraints. To satisfy the cost savings and security purposes, this paper combines system network loss, voltage dissatisfaction, and switching operation as the optimization objective. Voltage dissatisfaction can be converted into economic cost, so the objective function can be expressed as

$$\min C_{\mathrm{tot}} = C_{P\mathrm{loss}} + C_{\mathrm{VVIO}} + C_{\mathrm{TC}}. \quad (18)$$

The function should satisfy the equality and inequality constraints, where equality constraints reflect the physics of the power system shown as

$$P_{Gi} - P_{Li} = V_i \sum_{j=1}^{N} V_j \left(G_{ij} \cos\delta_{ij} + B_{ij} \sin\delta_{ij}\right),$$
$$Q_{Gi} - Q_{Li} + Q_{Ci} = V_i \sum_{j=1}^{N} V_j \left(G_{ij} \sin\delta_{ij} - B_{ij} \cos\delta_{ij}\right), \quad (19)$$

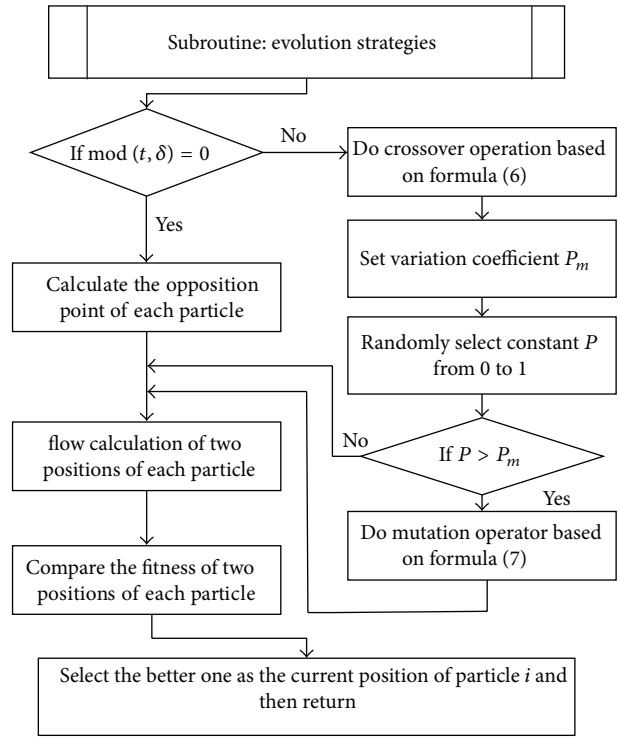


Figure 1: Subroutine of evolution strategies.

where N is node number; P_{Gi} and P_{Li} are generator active power output and active power of load at node i; and Q_{Gi}, Q_{Li}, and Q_{Ci} are generator reactive power input, load reactive power, and reactive power of the compensator capacitor at node i.

Inequality constraints are about the limits of active power, reactive power, voltage, capacitance, and the times of switching operations and are created to ensure system security [12] expressed as

$$\begin{aligned} P_{Gi\min} \le P_{Gi} \le P_{Gi\max} \quad & (i \in N_G), \\ Q_{Gi\min} \le Q_{Gi} \le Q_{Gi\max} \quad & (i \in N_G), \\ V_{i\min} \le V_i \le V_{i\max} \quad & (i \in N), \\ Q_{Ci\min} \le Q_{Ci} \le Q_{Ci\max} \quad & (i \in N_C), \\ T_{i\min} \le T_i \le T_{i\max} \quad & (i \in N_T), \end{aligned} \quad (20)$$

where N_G is the set of generation units; N_T is the set of tap transformers; and N_C is the set of compensator capacitors.

4. Reactive Power Optimization Using the Opposition-Based Improved PSO

Crossover and mutation operation and opposition-based learning are two improvement strategies. We combine and apply them to enhance diversity of particles. The specific process is shown in Figure 1. The main steps are as follows.

Step 1. Define the input data. The input data include the generator voltages and the transformer tap settings, reactive

power of switchable VAR sources, the population size of the particle swarm, the maximum number of iterations, and accelerated constants.

Step 2. Initialize the population. The initial population is generated randomly, which must meet the constraints. Calculate the opposition point $X_i'(t)$ of $X_i(t)$.

Step 3. Do flow calculation. Correct the system parameters for the flow calculation. Get the power system operation parameters, determine whether particles meet the bus voltage and generator reactive power and other constraints, modify the cross border values, and calculate their fitness.

Step 4. Compare the fitness of $X_i(t)$ and $X_i'(t)$; select the better one as the current generation.

Step 5. Record the individual optimal solution and the global optimal solution. For each particle, update the individual optimal solution. Then, select the best solution of the individual optimal solutions as the global optimal solution, $t = t + 1$.

Step 6. Calculate the current flight speed according to formula (4), and fix the particle position according to formula (1).

Step 7. Select the evolutionary strategies. Go into subroutine (see Figure 1); if $\mathrm{mod}(t, \delta) = 0$, calculate the opposition points as candidate solutions. Else, do crossover and mutation operations and generate candidate solutions according to formula (6) and (7).

Step 8. Update the particle position and modify the cross border values. After reactive compensation and transformer tap variables are discrete, do flow calculation (same as Step 3), compare the fitness of the candidate solution and contemporary individual, and select better one as the next generation.

Step 9. Determine whether it is under terminating condition. If the number of iterations at this time t is less than the maximum number of iterations, go to Step 5, or end the iteration and go to Step 10.

Step 10. Output the optimal solution. Optimal solution includes not only the control strategy of the control variables of each node but also the data of state variables, such as the system voltage of every node, system power loss, and generator reactive power output.

5. Results and Discussion

In order to validate the availability of the new opposition-based improved PSO (OIPSO) in solving power system reactive power optimization problems, IEEE 30- and IEEE 118-bus systems are employed to be the simulation studies. Two cases are presented in this section. In the first case, run with minimization of real power loss as the objective function and then compare the results with different methods. In the second case, take system network loss, voltage dissatisfaction, and switching operation as the optimization objective and give the optimal settings of control variables.

Table 1: Comparison of optimal results for different methods.

	Parameters	Losses (p.u.)
GA [22]	CR = 0.6; F = 0.01	0.04650
GSA [23]	G_0 = 100; α = 10	0.04617
DE [24]	F = 0.2; CR = 0.6	0.04550*
OGSA [25]	G_0 = 100; α = 10	0.04498*
PSO [26]	$w_{\max}$ = 0.9; $w_{\min}$ = 0.4; c_1 = 2	0.04814
CLPSO [26]	w_0 = 0.9; p_c = 0.4	0.04721
OIPSO	$w_{\max}$ = 0.9; $w_{\min}$ = 0.4; c_1 = 2; $c_{2\max}$ = 2.05; $c_{2\min}$ = 0.5; $c_{3\max}$ = 5; $c_{3\min}$ = 0.5	**0.04594**

CR is crossover rate; F is mutation rate; G_0 is the gravitational constant; p_c is learning probability, *The reactive compensation and transformer tap variables are continuous.

5.1. IEEE 30-Bus System. IEEE 30-bus system data and operating conditions are given in [21], which has 41 branches, 22 load buses, 6 generators (bus 1, bus 2, bus 5, bus 8, bus 11, and bus 13, while bus 1 is the slack bus and others are PV bus), 4 branches containing four adjustable transformers (branch 9-6, branch 10-6, branch 12-4, and branch 28-27, corresponding to transformers T1, T2, T3, and T4), and 2 shunt capacitors on buses 10 and 24, respectively. In the initial conditions, set the initial generator bus voltages and transformer taps to 1.0 and capacitor values to 0. The total power loss before optimization is 0.0537 p.u. In this paper, set the population size of particle swarm to be 36, and the number of iterations is 1000, δ = 2. In case one, active power loss is studied, and the result is compared with genetic algorithm (GA) [22], gravitational search algorithm (GSA) [23], differential evolution approach (DE) [24], an opposition-based gravitational search algorithm (OGSA) [25], particle swarm optimization (PSO) [26], and comprehensive learning particle swarm optimization (CLPSO) [26]. Do 50 trials and choose the best one shown in Table 1.

Although DE [24] and OGSA [25] have obtained the optimal values 0.04550 and 0.4498, their reactive compensation and transformer tap variables are continuous, which could not satisfy actual operations. The best solution is 0.045936 calculated by the proposed OIPSO algorithm; its average consumption would be 135 s more than the CPU time of PSO (i.e., 130 s [26]), due to its discrete control variables and flow calculation of opposition point. And it improves 14.45%, more than 14% by GSA, and 10.35% by PSO.

In case two, apply OIPSO to the proposed multiobjective problem, which includes system network loss, voltage dissatisfaction of on-load nodes, and switching operations of adjustable transformer. All expenses involved are shown in Table 2. The results are shown in Table 3.

Table 3 shows the optimal value of the control variables of OIPSO solving reactive power optimization. Comparing with PSO, the OIPSO makes small changes in transformer tap and uses little reactive power compensation to improve voltage quality and increase economic efficiency. Meanwhile,

Table 2: All expenses of reactive power optimization.

Unit network loss convert expensed λ_1 (10000 RMB/MW)	Voltage dissatisfaction penalty λ_2 (10000 RMB)	Unit operation cost of transformer tap λ_3 (10000 RMB)
69	3	0.5

Table 3: The values of control variables after optimization.

Variables	Bus number	PSO [26]	OIPSO
V_1	1	1.0995	1.0995
V_2	2	1.0933	1.0945
V_5	5	1.0697	1.0747
V_8	8	1.0719	1.0764
V_{11}	11	1.0480	1.0945
V_{13}	13	1.0945	1.0989
Q_{10}	10	0.3000	0.2000
Q_{24}	24	0.0500	0.0500
T_1	9~6	0.9375	0.9500
T_2	6~10	1.0000	1.0000
T_3	12~4	1.0125	1.0000
T_4	28~27	0.9750	0.9875
P_{loss}	—	0.04702	0.04625
S_V	—	0.001355	0.00573
S_T	—	7	5

the results also reflect that satisfactory interval is necessary, if some voltage values exceed the satisfactory interval, but do not go beyond the constraints, the power system could give up a little of satisfactory degree in exchange for active power loss minimum. This situation would not cause security problems; the dissatisfactory penalty could arouse some attention of operators. The satisfactory interval has a function of forecast.

5.2. IEEE 118-Bus System. To test the potential of OIPSO algorithm in solving bigger systems, IEEE 118-bus system is considered, which has 54 generator buses, 64 load buses, 186 transmission lines, 9 transformer taps, and 14 reactive power sources. The system data and operating conditions are given in [26]. The total power loss before optimization is 1.3359 p.u. For case one, Table 4 shows the comparison of results obtained by GSA [23], DE [25], OGSA [25], PSO [26], CLPSO [26], and OIPSO.

OIPSO discovered the best solution, that is, 1.0518 p.u., in which the improvement is 21.27%, more than 4.36% by GSA, 3.95% by DE, 4.94% by OGSA, 1.26% by PSO, and 2% by CLPSO. Its success rate lags behind CLPSO by 3 points, and CPU process is 1.196 times as PSO does, but OIPSO has more than 20 times as improvement as CLPSO and PSO. The results in this table indicate the superiority of OIPSO. Comparative PSO- and OIPSO-based convergence profiles of power losses for this test system are presented in Figure 2.

Table 4: Comparison of optimal results for different methods.

	P_{loss} (p.u.)	CPU time (s)	Success rate (%)
GSA [23]	1.2776	1199	NR*
DE [25]	1.2832	NR*	NR*
OGSA [25]	1.2699	1101.3	NR*
PSO [26]	1.3191	1215	59
CLPSO [26]	1.3096	1472	73
OIPSO	**1.0518**	1453	70

NR*: not reported.

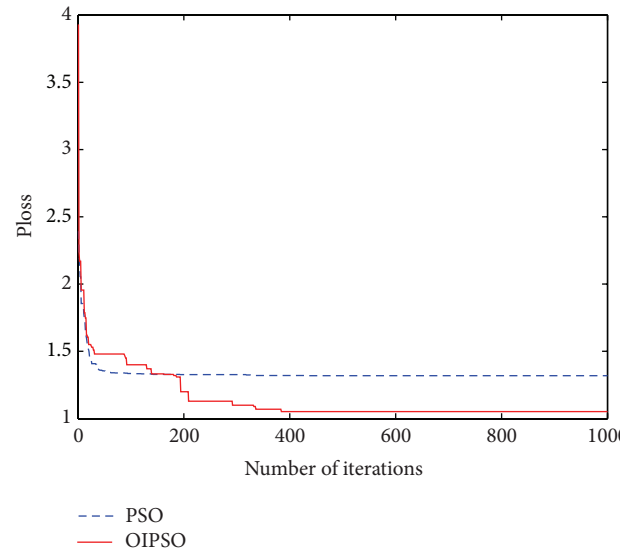


Figure 2: Comparative convergence profiles of power loss.

In case two, the multiobjective results of IEEE 118-bus system are given in Table 5; these values of the parameters provide actual operation variables, which is more effective and practical.

6. Conclusion

In this paper, we developed an opposition-based improved PSO for multiobjective reactive power optimization. The main novelty of the algorithm is the integration of the opposition-based computing into the basic PSO with inertia weight factors, crossover and mutation, and neighborhood model in order to enhance the diversity and produce some additional exploration ability of the population. The performance of the proposed OIPSO algorithm is demonstrated through its calculation on IEEE 30- and 118-bus systems; the comparison results with other algorithms show that OIPSO has better global search ability and fast searching speed. Meanwhile, when the new multiobjective model of reactive power optimization is being built, switching operation cost is defined by the life cycle cost and is emphasized to reflect the actual situation. So, the optimized results are more realistic and more reliable for practical operation.

Table 5: The values of control variables after optimization.

Variables	Bus number	PSO [26]	OIPSO
V_1	1	0.9736	1.0844
V_4	4	1.0145	1.0958
V_6	6	0.9969	1.0977
V_8	8	1.0169	1.0956
V_{10}	10	1.0344	1.0967
V_{12}	12	0.9868	1.0925
V_{15}	15	0.9867	1.0899
V_{18}	18	0.985	1.0929
V_{19}	19	0.9868	1.0893
V_{24}	24	0.9799	1.0917
V_{25}	25	1.0014	1.0981
V_{26}	26	1.0424	1.0955
V_{27}	27	1.002	1.0864
V_{31}	31	1.002	1.0855
V_{32}	32	1.001	1.0866
V_{34}	34	0.9964	1.097
V_{36}	36	0.9881	1.0932
V_{40}	40	0.9699	1.0731
V_{42}	42	1.0282	1.088
V_{46}	46	1.0165	1.0758
V_{49}	49	1.0213	1.0919
V_{54}	54	1.0147	1.081
V_{55}	55	1.0113	1.0819
V_{56}	56	1.0131	1.0814
V_{59}	59	1.0424	1.096
V_{61}	61	1.0252	1.0911
V_{62}	62	1.0284	1.0873
V_{65}	65	1.0377	1.0945
V_{66}	66	1.0381	1.0983
V_{69}	69	1.0161	1.0972
V_{70}	70	0.9551	1.0795
V_{72}	72	0.9793	1.087
V_{73}	73	0.9269	1.0852
V_{74}	74	0.9466	1.0593
V_{76}	76	0.938	1.0639
V_{77}	77	1.0049	1.0777
V_{80}	80	1.0194	1.0918
V_{85}	85	1.0303	1.095
V_{87}	87	1.09	1.0848
V_{89}	89	1.0361	1.0979
V_{90}	90	0.9907	1.0811
V_{91}	91	1.0087	1.0864
V_{92}	92	1.0183	1.0965
V_{99}	99	1.0311	1.084
V_{100}	100	1.0112	1.0975
V_{103}	103	1.0027	1.0885
V_{104}	104	1	1.091
V_{105}	105	1.0101	1.0829
V_{107}	107	0.9857	1.0867
V_{110}	110	0.9663	1.0647
V_{111}	111	0.972	1.0728
V_{112}	112	0.9529	1.0521
V_{113}	113	1.0241	1.0938
V_{116}	116	1.0285	1.0979
Q_5	5	0	−0.3
Q_{34}	34	0.1	0
Q_{37}	37	−0.25	−0.25
Q_{44}	44	0.1	0.05
Q_{45}	45	0.15	0.05
Q_{46}	46	0.2	0.1
Q_{48}	48	0.05	0.1
Q_{74}	74	0.1	0.05
Q_{79}	79	0.15	0.15
Q_{82}	82	0.2	0.2
Q_{83}	83	0.3	0
Q_{105}	105	1	1.05
Q_{107}	107	0.2	0.1
Q_{110}	110	0.05	0.05
T_1	5~8	1.025	1.0125
T_2	25~26	1.0125	1.0125
T_3	17~30	0.9875	0.975
T_4	37~38	0.9875	0.9875
T_5	59~63	1	1
T_6	61~64	1	0.9875
T_7	66~65	1	0.9875
T_8	69~68	0.975	1
T_9	80~81	1.025	0.9875
P_{loss}	—	1.2033	1.1605
S_V	—	0.000585	0.007313
S_T	—	18	8

Conflict of Interests

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 51177039), Doctoral Fund of Ministry of Education of China (Grant no. 20090094110011), the Nantong Science and Technology Project, Nantong, China (Grant no. BK2014024), the Changzhou science and technology project, Changzhou, China (Grant no. CJ20130008), and the Open Project of Jiangsu Province Key Laboratory of Environmental Engineering, Nanjing, China (Grant no. KF2014001).

References

[1] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[2] Y. Shi and R. Eberhart, "Modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation, and IEEE World Congress on Computational Intelligence*, pp. 69–73, May 1998.

[3] A. Ratnaweera, S. Halgamuge, and H. Watson, "Particle swarm optimization with self-adaptive acceleration coefficients," in *Proceedings of the 1st International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 246–268, 2003.

[4] S. Burak Akat and V. Gazi, "Particle swarm optimization with dynamic neighborhood topology: three neighborhood strategies and preliminary results," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '08)*, pp. 1–8, September 2008.

[5] H. S. Lopes and L. S. Coelho, "Particle swarm optimization with fast local search for the blind traveling salesman problem," in *Proceedings of the 5th International Conference on Hybrid Intelligent Systems (HIS '05)*, pp. 245–250, IEEE, November 2005.

[6] W. Jian and X. Chuanpei, "Study on test generation of sequential circuits based on particle swarm optimization and ant algorithm," in *Proceedings of the International Conference on Computer Science and Software Engineering (CSSE '08)*, pp. 149–152, Wuhan, China, December 2008.

[7] X. Wu and G. Huang, "Application of particle filter algorithm in nonlinear constraint optimization problems," in *Proceedings of the 8th International Conference on Natural Computation (ICNC '12)*, pp. 822–826, May 2012.

[8] H. Wang, H. Li, Y. Liu, C. Li, and S. Zeng, "Opposition-based particle swarm algorithm with Cauchy mutation," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 4750–4756, Singapore, September 2007.

[9] K. Massimiliano, "A multi-start opposition-based particle swarm optimization algorithm with adaptive velocity for bound constrained global optimization," *Journal of Global Optimization*, vol. 55, no. 1, pp. 165–188, 2013.

[10] R. Patel, M. M. Raghuwanshi, and L. G. Malik, "Decomposition based multi-objective genetic algorithm (DMOGA) with opposition based learning," in *Proceedings of the 4th International Conference on Computational Intelligence and Communication Networks*, pp. 605–610, November 2012.

[11] Z. Wu, Z. Ni, C. Zhang, and L. Gu, "Opposition based comprehensive learning particle swarm optimization," in *Proceedings of 3rd International Conference on Intelligent System and Knowledge Engineering (ISKE '08)*, pp. 1013–1019, November 2008.

[12] T. Niknam, M. R. Narimani, R. Azizipanah-Abarghooee, and B. Bahmani-Firouzi, "Multiobjective optimal reactive power dispatch and voltage control: a new opposition-based self-adaptive modified gravitational search algorithm," *IEEE Systems Journal*, vol. 7, no. 4, pp. 742–753, 2013.

[13] H. Wang, H. Jiang, K. Xu, and G. Li, "Reactive power optimization of power system based on improved particle swarm optimization," in *Proceedings of the 4th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT '11)*, pp. 606–609, Shandong, China, July 2011.

[14] R. C. Eberhart and J. Kennedy, "New optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, October 1995.

[15] Z. H. Zhan, J. Zhang, Y. Li, and H. S. H. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 6, pp. 1362–1381, 2009.

[16] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition versus randomness in soft computing techniques," *Applied Soft Computing Journal*, vol. 8, no. 2, pp. 906–918, 2008.

[17] P. Umapathy, C. Venkataseshaiah, and M. S. Arumugam, "Particle swarm optimization with various inertia weight variants for optimal power flow solution," *Discrete Dynamics in Nature and Society*, vol. 2010, Article ID 462145, 15 pages, 2010.

[18] S. G. Kim, H. Outhred, and I. Macgill, "Commercialising voltage regulation in nodal electricity spot markets," *International Energy Journal*, vol. 6, no. 1, pp. 219–228, 2005.

[19] Y. Song and J. Li, "Analysis of the life cycle cost and intelligent investment benefit of smart substation," in *Proceedings of the IEEE Innovative Smart Grid Technologies-Asia (ISGT-Asia '12)*, pp. 1–5, IEEE, Tianjin, China, May 2012.

[20] X. Jiang, W. Xuan, and X. Chen, "Research and application of power transformer life cycle cost model," *Journal of Fuzhou University (Natural Science Edition)*, vol. 40, no. 3, pp. 357–361, 2012 (Chinese).

[21] R. Li and S.-Q. Sheng, "An improved particle swarm optimization algorithm for reactive power optimization," in *Asia-Pacific Power and Energy Engineering Conference (APPEEC '11)*, pp. 1–5, March 2011.

[22] S. Durairaj, P. S. Kannan, and D. Devaraj, "Application of genetic algorithm to optimal reactive power dispatch including voltage stability constraint," *Journal of Energy & Environment*, pp. 63–73, 2005.

[23] S. Duman, Y. Sonmez, U. Guvenc, and N. Yorukeren, "Application of gravitational search algorithm for optimal reactive power dispatch problem," in *Proceedings of the International Symposium on INnovations in Intelligent SysTems and Applications (INISTA '11)*, pp. 519–523, June 2011.

[24] A. A. A. E. Ela, M. A. Abido, and S. R. Spea, "Differential evolution algorithm for optimal reactive power dispatch," *Electric Power Systems Research*, vol. 81, no. 2, pp. 458–464, 2011.

[25] B. Shaw, V. Mukherjee, and S. P. Ghoshal, "Solution of reactive power dispatch of power systems by an opposition-based gravitational search algorithm," *International Journal of Electrical Power and Energy Systems*, vol. 55, pp. 29–40, 2014.

[26] K. Mahadevan and P. S. Kannan, "Comprehensive learning particle swarm optimization for reactive power dispatch," *Applied Soft Computing Journal*, vol. 10, no. 2, pp. 641–652, 2010.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 594345, 7 pages
http://dx.doi.org/10.1155/2015/594345

Research Article

Finding Robust Assailant Using Optimization Functions (FiRAO-PG) in Wireless Sensor Network

Piyush Kumar Shukla,[1] Sachin Goyal,[2] Rajesh Wadhvani,[3] M. A. Rizvi,[4] Poonam Sharma,[5] and Neeraj Tantubay[2]

[1]*Department of Computer Science and Engineering, UIT, RGPV, Bhopal, India*
[2]*Department of Information Technology, UIT, RGPV, Bhopal, India*
[3]*Department of Information Technology, MANIT, Bhopal, India*
[4]*Department of Computer Engineering and Applications, NITTTR, Bhopal, India*
[5]*Department of Computer Science & Information Technology, MITS, Gwalior, India*

Correspondence should be addressed to Piyush Kumar Shukla; pphdwss@gmail.com

Received 24 September 2014; Accepted 23 December 2014

Academic Editor: Gerhard-Wilhelm Weber

Wireless sensor network consists of hundreds or thousands of low cost, low power, and self-organizing tiny sensor nodes that are deployed within the sensor network. Sensor network is susceptible to physical attacks due to deprived power and restricted resource capability and is exposed to external environment for transmitting and receiving data. Node capture attack is one of the most menacing attack in the wireless sensor network and may be physically captured by an adversary for extracting confidential information regarding cryptographic keys, node's unique id, and so forth, from its memory to eliminate the confidentiality and integrity of the wireless links. Node capture attack suffers from severe security breach and tremendous network cost. We propose an empirically designed multiple objectives node capture attack algorithm based on optimization functions as an effective solution against the attacking efficiency of node capture attack. Finding robust assailant optimization-particle swarm optimization and genetic algorithm (FiRAO-PG) consists of multiple objectives: maximum node participation, maximum key participation, and minimum resource expenditure to find optimal nodes using PSO and GA. It will leverage a comprehensive tool to destroy maximum portion of the network realizing cost-effectiveness and higher attacking efficiency. The simulation results manifest that FiRAO-PG can provide higher fraction of compromised traffic than matrix algorithm (MA) so the attacking efficiency of FiRAO-PG is higher.

1. Introduction

With the advent of wireless communication technology, sensor network has become a user-perceived network for many applications in the recent years. WSN inheres apportioned low power, low-cost, and self-organizing sensor nodes to get secret information and plays a decisive role to preserve confidentiality and integrity of the radio links in applications like military surveillance, environment monitoring, and many more scenarios. However, due to unattended nature of sensor network, it is highly vulnerable to physical node capture attack [1]. It is an empirically derived comprehensive attack where the adversary physically tampers the sensor node by extracting cryptographic keys and other top-secret information. By leveraging the extracted confidential information, an attacker gathers secret information from the network by eavesdropping communication among the sensor nodes. Node capture is the most vexing problem that jeopardizes the confidentiality, reliability, and security of sensor nodes.

Analyzing the way of mounting the node capture attack can provide threatening models for developing defending techniques against it. Development of effective threatening techniques keeps greater importance because the performance of defending technique directly depends on it. The former node capture attack algorithms proposed suffered drawbacks like attacking modeling and low attacking efficiency. Modeling techniques of attack formalize the mechanism for analyzing the behavior of an adversary. After designing the modeling techniques, the adaptation of an adversary can be

intuitively expressed that exhibits the attacking target and attacking process. Therefore, the modeling of node capture attack can abet heuristic strategies for an adversary. To prototype node capture attack, attacking efficiency is an imperative characteristic that is used to delineate fraction of compromised traffic within the network. Higher attacking efficiency, which represents the network, will be compromised. Therefore, designing an enhanced and optimized way of node capture attack accreting improved attacking efficiency sanctifies its significance.

Multiform researchers have proposed techniques for modeling the node capture attack for its deeper analysis and design of a random key predistribution scheme [2] is applied for configuration of sensor network. Hypothetically, formalization of modeling techniques of the node capture attack can be categorized into an UML methods [1, 3], probabilistic analysis [3, 4], system theoretic approach [5], epidemic theory [6, 7], and vulnerability evaluation approach [8–12].

In certain modeling methods, the attacker randomly captures node to compromise in the communication of a whole sensor network. However, vulnerability evaluation approach has been formalized whereby an attacker can select a node intelligently to compromise the network using vulnerability metric. Vulnerability evaluation methods proposed so far in [8–12] suffer certain drawbacks.

(i) Vulnerability is distinguished as a real number, which cannot precisely outline the destructiveness within the network.

(ii) Attacking efficiency of node capture attack is low, which needs to capture a high number of nodes in the network.

(iii) Some vulnerability approach is restricted for deterministic key protocol, which is unsuitable for mobile network.

(iv) In few vulnerabilities based assailing approaches, node capture attack is modeled from the perspective of relationship among nodes and paths that not provide optimal solution to capture a node.

(v) To evaluate the vulnerability metric, some previous algorithms only provide a focus on the resource expenditure to capture a node, some provide focus on the maximum number of keys captured, and some take route vulnerability into the consideration. Until now, there is no approach designed that takes all the three objectives into the consideration.

To overcome the above problems of vulnerability based approaches and to enhance the attacking efficiency of the node capture attack, we model the node capture attack algorithm that works on the three objectives maximum node participation, maximum key participation, and minimum resource expenditure to find an optimal node using PSO that creates maximum destructiveness in the network and provides higher attacking efficiency at minimum resource expenditure.

2. Related Work

In the literature, various researchers have proposed modeling techniques based on the vulnerability evaluation. In this type of technique, an attacker can intelligently choose a node to attack with eavesdropping on the insecure messages transmitting in the network. In [10], the authors proposed a mathematical model for modeling of node capture attack on the different key establishment protocols within the heterogeneous wireless ad hoc and wireless mesh networks. Node capture attack is modeled using an integer-programming minimization problem which derives the NP-hard set cover problem in which attacks are evaluated with respect to the attacking cost and the benefit of an attack to the attacker. Node capture attack algorithm that receives the expected benefit at less cost for an attacker is mapped to the NP-hard minimization problem. The valuable solution for this problem is estimated with the help of known heuristics that are set coverage and subset coverage. This modeling technique of node capture attack is limited to probabilistic and deterministic key distribution schemes.

In [12], the authors examined the effect of physical node capture attack on the integrity and confidentiality of the network for which they present the node capture attack algorithm as a nonlinear integer programming problem. Because of NP-hardness of the minimization problem, the greedy node captured approximation using vulnerability evaluation protocol (GNAVE) is proposed. GNAVE is an elegant solution for approximating the minimum cost node capture attack. In the GNAVE algorithm, the compromise of the network traffic is mapped as flow of current, which is passed through an electronic circuit. In this algorithm, the route vulnerability metric is proposed, which depends on the routing and cryptographic protocols that can minimize resource expenditure to a certain level. At each step of the GNAVE algorithm, the attacker selects to capture the node with higher values per unit cost to improve the cost-effectiveness of the node capture attack. GNAVE provides increasing attacking efficiency for node capture attack by compromising fewer nodes with higher fraction of compromising traffic of the network. It does not include the execution time into the consideration for destroying the complete network [12].

In [8], Wu et al. proposed a greedy node capture attack algorithm based on the route minimum key set (GNRMK). In GNRMK, sensor network is mapped as flow network to acquire its route minimum key set that shows the vulnerability of the route within the network. The route minimum key set is used to destroy the network with less resource expenditure. It is achieved by evaluating the max-flow of the flow network by using the labeling and adjustment procedure which is based on the Ford-Fulkerson algorithm. Then, a node overlapping to values metric (NOV) is evaluated using the route minimum key sets. A node that has been maximum overlapping value is targeted as a node to be captured within the network. After capturing a node with highest overlapping values, the network topology is dynamically changed due to already compromised links or paths. GNRMK algorithm can only utilize within the static networks because it is exclusively

Table 1: A summary of related symbols and their definitions.

Symbols	Description
N	Set of sensor nodes in the network
N_i	ith sensor node
K	Set of total keys in the key pool
K_i	Set of keys acquired by node n_i
L	Set of links between nodes
$l(i, j)$	Link between node n_i and node n_j
S, D	Set of source and destination nodes
R	Set of routes in the network
W_i	Capturing cost of node n_i
C_n	Set of compromised nodes
$P(R_i)$	Total number of paths of route R_i
$Pk(i, j)$	Number of paths in which node n_j participates in route R_i
F_i	Objective function for node n_i

restricted to deterministic key protocol that is unsuitable for mobile networks.

3. Models and Definitions

The matrix algorithm (MA) is proposed in [9] to perform the node capture attack within the sensor network, when the network is configured with the random key predistribution scheme. In this algorithm, a compromising matrix is designed for evaluating a node that makes the network highly vulnerable by establishing a relationship between nodes and paths. Matrix algorithm takes less resource expenditure with larger destructiveness within the network. MA takes less number of attacking rounds, less execution time, high attacking efficiency, and less resource expenditure [9]. It is limited to the random key predistribution scheme, and it also provides less attention towards the relationship between the attacking efficiency of the node capture attack and attacking cost. It only provides the focus on resource expenditure to capture a specific node that means a node that has less resource expenditures compared to others, it is compromising enough paths in the network, and it will be captured. MA also considers those paths that are not influenced by capturing a particular node that increases computation overheads. Models and definitions.

This section includes the proposed models and various definitions related to our work. Table 1 summarizes the related symbols and their definitions.

3.1. Key Assignment Model. In WSN, the random key predistribution scheme is used to assign set of keys $K_i \in K$ to each sensor $n_i \in N$ that is randomly chooses from key pool. The set of keys shared between nodes n_i and n_j is represented by $K(i, j)$. Neighboring nodes can communicate with each other, when they are located within each other's transmission range r and they have at least one common sharing key $K(i, j)$. The more the number of keys in the set $K(i, j)$ will be, the higher the security of the link will be.

3.2. Network Model. In the network model, wireless sensor network consists of set of N sensor nodes and the network is represented by a directed network graph $G = (N, l)$. After the deployment of sensor nodes, the specific routing protocol is applied to construct multiple routes for transmitting packets from source to destination nodes. A packet from source node $s \in S$ to destination $d \in D$ will traverse one or more paths that depend on the routing protocol. Each of the paths of a particular route is constructed from a set of sequential links (i, j). The link topology of the network is presented by a $N \times N$ matrix (C) where a "c_{ij}" represents the link cost between the nodes, as shown in the following part.

The Connection Cost Matrix for 3 Nodes System. Consider

$$\begin{matrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{matrix} \quad (1)$$

Number link topology matrix is used by specific routing protocols to estimate the possible paths or routes in the network. Two kinds of routing protocols are implemented here that are single path and multipath routing protocols to evaluate the effect of node capture attack algorithm. The single path routing protocol establishes a single path from source to destination, whereas multipath protocol establishes more than one path to transmit packet from source to destination.

3.3. Adversary Model. The node capture attack algorithm is modeled from an attacker's point of view and it is assumed that the attacker has capability and network resources to eavesdrop on the messages passing through the network for capturing a node in the network and extracting cryptographic keys and other pieces of confidential information from nodes memory in polynomial time. It is also assumed that the attacker has knowledge of the key assignment model and routing protocols used in the network like key is represented as label key and adversary has knowledge of assignment of labels of the network keys, paths, and routes implemented by specific routing protocol. In the sensor network, destination nodes are usually implemented with higher security and protection mechanisms. So, it is considered that an attacker cannot intrude them.

The attacker's aim is to propose an efficient attack to compromise the whole network by achieving multiple objectives: minimum resource expenditure, maximum keys, and maximum capture of transmitting packets. To fulfill this goal, the particle swarm optimization technique is used to find network nodes that make sensor network more vulnerable. Therefore, the network is analyzed by an attacker to own the background information of the key assignment protocol and network parameter to model the attacking algorithm.

To compromise the sensor network, it is required for an attacker to extract the keys by capturing a node to break the security, confidentiality, and integrity of the network. To represent the compromise of a link, path, and route in the sensor network the following definitions are illustrated as follows.

Definition I. A link $l(i, j) \in l$ is compromised when the sharing key $K(i, j)$ belongs to the set of keys acquired by an attacker.

Definition II. A path $pi \in P$ is compromised when at least one link which belongs to that path is compromised.

In the single path routing protocol, a route consists of a single path only. So compromise of that single path is equivalent to compromise of a route. But in case of multipath routing, packets are segmented into pieces and then transmitted from separate paths. To compromise such routes the following definition is proposed.

Definition III. A route, $Rs, d \in R$ is compromised when all the paths which belong to that route are compromised.

Particle swarm optimization is a population based computational technique. It learns from the scenario and is used to find a potential solution to an optimization problem. In the context of PSO, a swarm refers to a number of potential solutions to the optimization problem, where each potential solution is referred to as a particle. PSO is initiated by a group of random particles and looks for an optimum value by updating generations. In each round, each particle is updated by tracking two best values: first one is the p_{best} (personal best) value. This is the value of the fitness function; it has been achieved so far. Another one is called the g_{best} (global best). This value is the best value obtained so far by any particle in the population and tracked by the particle swarm optimizer. After finding p_{best} and g_{best}, the particles update its velocity and position with the following equations:

$$\begin{aligned} V_i^{t+1} &= wV_i^t + c_1 \text{rand}\,() \left(P_i - X_i^t\right) \\ &\quad + c_2 \text{rand}\,() \left(P_g - X_i^t\right), \\ X_i^{t+1} &= X_i^t + V_i^{t+1}, \end{aligned} \tag{2}$$

where $i = 1, 2, \ldots, N$, N represents the individual number in the group; w is the inertia coefficient; t represents the iteration number; c_1 and c_2 are learning factors; rand() is uniformly distributed random variables in $[0, 1]$; P_i is particles best position till iteration t; X_i^t is particles current position in iteration t; and P_g represents globally best particle position in iteration t.

The basic procedure of the PSO algorithm is as follows.

(i) Assign initial values to the position and velocity of all particles.

(ii) Evaluate the fitness of each particle according to the desired optimization. So the optimal value of individuals (personal best) and optimal value of swarm (global best) can be obtained.

(iii) Update the position and velocity of the particles.

(iv) Determine whether the conditions meet ends, if not, go to Step 2.

4. Multiple Objectives Node Capture Attack Algorithm Based on PSO (FiRAO-PG)

The aim of the node capture attack algorithm is to capture a set of nodes to compromise the complete network. So, all the paths that belong to different routes should be captured for compromising the whole network. Therefore, the attacker's goal is to compromise maximum possible routes of the network by capturing a limited number of nodes that satisfy multiple objectives which are maximum participation of nodes within the network so that the maximum packets communicated within the network can be eavesdropped, and also maximum keys and minimum resource expenditure. The presented algorithm evaluates the optimal nodes for the node capture using PSO such that only a limited number of nodes capturing compromise the whole network by providing maximum benefit to an attacker.

To analyze the participation of sensor nodes in the network, we calculate the route node participation matrix, which represents the participation of each sensor node in each route of the network. Participation of a node in a specific route manifest that in how many path nodes participated to transfer packets to another node from all the paths belongs to that distinct route. In other words, it represents the participation ratio for each node in the network on the basis of number of paths in which that node belongs among all the paths available in that particular route. We denote the route node participation matrix as $\text{RN} = [\text{RN}(i, j)]R \times N$, where

$$\text{RN}\,(i, j) = \begin{cases} \dfrac{Pk\,(i, j)}{P\,(R_i)} & \text{If Node } n_j \text{ participates in Route } R_i \\ 0 & \text{Otherwise.} \end{cases} \tag{3}$$

$Pk(i, j)$ represents the number of paths of route R_i in which node n_j participates; $P(R_i)$ stands for total number of paths of route R_i.

To achieve another objective that is capturing a node that contains maximum keys together with maximum participation and minimum resource expenditure, we create another key node participation matrix that shows the belonging relationship between keys and nodes. Key node participation matrix $\text{KN} = [\text{KN}(i, j)]K \times N$ can be represented as follows:

$$\text{KN}\,(i, j) = \begin{cases} 1 & \text{If Key } k_i \in n_j \\ 0 & \text{Otherwise.} \end{cases} \tag{4}$$

Here, k_i represents ith key of key pool. To evaluate number of keys a node has in their memory, we calculate the key participation matrix that represents number of key belongs to a particular node of the network. We denote the key participation matrix as $K = [K_j]1 \times N$, where

$$K_j = \begin{cases} \displaystyle\sum_{i=1}^{K} \text{KN}\,(i, j) & \text{If some Keys belong to Node } n_j \\ 0 & \text{Otherwise.} \end{cases} \tag{5}$$

Here, K_j represents the number of keys assigned to node n_j. In the node capture attack, another issue the adversary must pay attention towards is the resource expenditure or energy cost. The adversary seeks for compromising a set of nodes that consumes least energy together with maximum participation and maximum key to wreaking the security of the network. Therefore, we calculate capturing cost matrix that represents the energy cost or resource expenditure for each node in the network. Capturing cost matrix $W = [w_i]1 \times N$ is denoted as follows:

$$W_i = \begin{cases} w_i & \text{Capturing Cost of Node } n_i \\ 0 & \text{Otherwise.} \end{cases} \quad (6)$$

Here, w_i represents the resource expenditure for node n_i. The capturing cost of each node is related to the environment that is exposed by a node and the capability of an attacker. So it is very difficult to elaborate on energy cost in capturing a node. Here, we consider that the resource expenditure of capturing a node ranges between $(0, 1)$.

After evaluating route node participation matrix RN, key participation matrix K, and capturing cost matrix W, we need to find optimal nodes in the network that provide best result on the basis of multiple objectives which are taken into the consideration. To find such kinds of nodes, PSO algorithm is utilized here that provides best optimal nodes that creates maximum destructiveness in the network.

Before applying PSO algorithm, first we evaluate the objective function to achieve the required goal. The objective function can be written as follows:

$$f_j = \sum_{i=1}^{\mathrm{NA}} \sum_{i=1}^{R} \frac{1}{\mathrm{RN}(i, j)} + \frac{1}{K_j} + W_j. \quad (7)$$

Here, f_j represents the objective function of jth node, $\mathrm{RN}(i, j)$ shows participation of jth node in route R_i, K_j stands for keys assessable through jth node in all the possible paths or routes, and W_j represents the capturing cost or resource expenditure of jth node in all the possible paths or routes.

After evaluating objective function, PSO algorithm is initiated to find optimal nodes from the available network nodes, which minimize the value of objective function to provide the best results. In order to find the optimal set of nodes that creates maximum destructiveness in the network using PSO, we define a location mapping equation as

$$X_i = \mathrm{Round}\left(X_i * (\text{Total sensor nodes} - 1)\right) + 1. \quad (8)$$

Equation (8) represents that if the positions of the particles cannot corresponds to node id, then we can find the nearest node id.

The set of compromised nodes is returned by FiRAO-PG that contains node indexes, which provide optimal results based on all the three objectives. Maximum node participation, minimum resource expenditure, and maximum key participation. These three objectives provide following features: (1) it seeks for the maximum participated node that induces maximum destructiveness in the network because if a node has maximum participated value that means it belongs to the higher number of paths and provides maximum capturing of transmitted packets, (2) it takes least resource expenditure to compromise the network, and (3) it acquires maximum keys of key pool that helps to compromise a higher number of paths either directly or partially. The attacking algorithm of node capture attack ends when the whole network is compromised and it returns set of compromised nodes C_n as output of this algorithm. From the perspective of an attacker, the set of compromised nodes C_n (returns by FiRAO-PG) causes maximum destructiveness in the network. But from the defenders point of view, this algorithm provides vulnerable nodes of the network to strengthen the security of the network.

4.1. FiRAO-PG Algorithm. (1) Input: $G(N, l)$, K, w_i.

(2) Output: C_n.

(3) Calculate route node participation matrix using (3).

(4) Calculate key participation matrix using (5) and calculate key node participation matrix using (4).

(5) Calculate capturing cost matrix using (6).

(6) The objective function is then formulated to achieve the required goals using FiRAO-PG algorithm and can be written as

$$f_j = \sum_{i=1}^{\mathrm{NA}} \sum_{i=1}^{R} \frac{1}{\mathrm{RN}(i, j)} + \frac{1}{K_j} + W_j. \quad (9)$$

(7) Optimization algorithm is initiated to find the set of optimal nodes of the available nodes, which minimizes the value of the objective function.

Here, each set of different combinations of nodes are defined as particles of the randomly generated population and dimensions of each particles are defined as number of nodes required to capture. The algorithm works as follows.

Step 1. Initialize population to random size m ($m < n$). Initialize position and velocity of each particle (representing node ids) i to random value.

Step 2. Compute the fitness of each particle using (6) and obtain the optimal value of the individual and of the population.

Step 3. Update position and velocity of each particle using (2), and then adjust position of particles using (8) to find node ids.

Step 4. Test for convergence conditions, if not, go to Step 2; else provide a set of optimal nodes and return index of that nodes to perform node capture attack.

(8) GA algorithm is initiated to find the set of optimal nodes.

Step 1. Initial population random (random selection from nodes).

Step 2. Crossover (node), // a new chromosome is created with 2 parents.

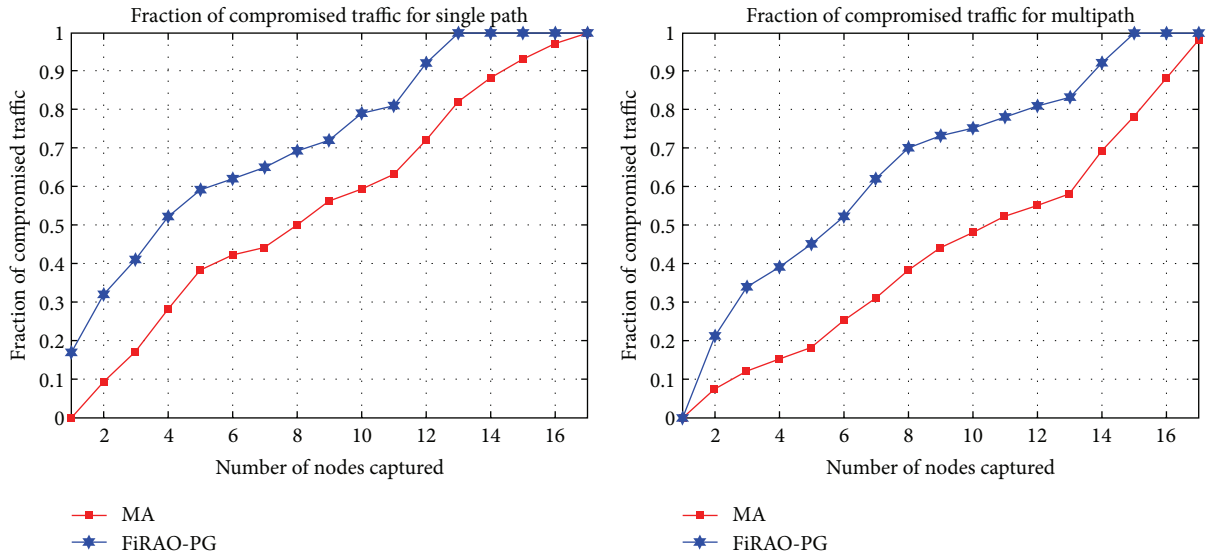


Figure 1: (a) Fraction of compromised traffic versus number of nodes captured for single path routing. (b) Fraction of compromised traffic versus number of nodes captured for multipath routing.

Step 3. Mutation (node), // a chromosome from each parent changes. Then the distance between the new node and the sink is determined.

Step 4. Merge (2 new populations).

Step 5. Sort (node), // (distance to sink, energy).

Step 6. Size (new-population) = size (population).

Step 7. Sort (new population), // the second time.

Step 8. Creating cluster head.

(9) Return C_n based on the best values from PSO and GA.

Table 2: Simulation parameters.

Parameters	Values
Number of sensor nodes	200
Region size	100∗100
Number of source nodes	5
Sensing range	20 m
Number of destination nodes	3
Key pool size	100
Number of keys assigned to a node	20
Set of source and destination nodes	R
Population size	50
Number of iteration	200

5. Result and Analysis

To analyze the performance of multiple objectives node capture attack algorithm based on PSO and GA (FiRAO-PG), we performed the following simulation. The experimental parameters are shown in the Table 2.

In the simulation work, 200 nodes are deployed in the sensor network. From the total deployed nodes, 5 source sensor nodes and 3 destination nodes are randomly selected. Random key predistribution scheme is used to assign keys to different nodes in the sensor network. Keys are assigned randomly from the key pool to each sensor node, when the network is deployed. Sensor nodes located in 20 m can communicate with each other in the simulation. Two kinds of routing protocol are used that are single path routing protocol and multipath routing protocol. An attacking algorithm for node capture attack is analyzed on single path and multipath routing protocol to check influence of this attacking algorithm. The proposed FiRAO-PG algorithm runs 200 iterations. We measure the performance of our proposed work in terms of fraction of compromised traffic.

To show the advantage of our proposed algorithm, we provide a comparison with an MA (matrix algorithm) [9] in terms of fraction of compromised traffic. Both algorithms MA [9] and multiple objectives node capture attack algorithm based on PSO (FiRAO-PG) have same input parameters in the simulation that are $G(N, l)$, K, w_i.

5.1. Fraction of Compromised Traffic. Fraction of compromised traffic represents the ratio of compromised paths among all the paths in the network. Figure 1 illustrates the fraction of compromised traffic of FiRAO-PG and MA for single path and multipath routing. In this experiment, x-coordinate represents the number of nodes captured by an adversary while the y-coordinate indicates the fraction of the traffic that are compromised by an adversary. The fraction of compromised traffic can indicate attacking efficiency of an algorithm. As soon as the fraction of compromised traffic

reaches 1, higher will be the attacking efficiency of that algorithm. Figure 1 illustrates that multiple objectives node capture attack algorithm based on PSO (FiRAO-PG) can quickly approach 1 due to this algorithm that aims to capture node that maximally participates in the network together with keeping maximum keys and consuming less resource expenditure. Therefore, this algorithm causes maximum destructiveness in the sensor network by compromising the maximum number of paths. MA needs comparatively more nodes to break the confidentiality of the network. So, we can conclude that FiRAO-PG provides higher attacking efficiency than MA.

6. Conclusions

We proposed that a FiRAO-PG (finding robust assailant optimization-particle swarm optimization and genetic algorithm) has been used for enhancing the attacking efficiency of the node capture attack in the wireless sensor network. FiRAO-PG takes three objectives into consideration that are maximum key participation, bare minimum resource expenditure, and maximum node participation to find optimal nodes that provide the best combination for all the objectives and causes maximum destructiveness in the network. The simulation result shows that FiRAO-PG provides a higher fraction of compromised traffic, when compared with a matrix algorithm (MA). Therefore, FiRAO-PG provides higher attacking efficiency than MA by capturing a limited number of nodes that compromises whole network.

(i) How to further minimize the number of captured nodes using ACO (ant colony optimization, binary-PSO, simulated annealing) to compromise the whole network to enhance the attacking efficiency of the node capture attack algorithm.

(ii) We can also check performance of FiRAO-PG in the clustered sensor network.

(iii) Performance of the FiRAO-PG can be checked in high mobility networks (VANET, DTN, etc.)

Conflict of Interests

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

Authors' Contribution

Piyush Kumar Shukla deplored the algorithms for this work and generates the idea of how a robust assailant using optimization functions (FiRAO-PG) in wireless sensor network can be designed. Sachin Goyal contributed to implementation of the work using genetic algorithm. Rajesh Wadhvani contributed to summery of the work done in a relevant area till date and also developed understanding of the base paper's work. M. A. Rizvi contributed to summery of the work done in a relevant area till date and also developed understanding of the base paper's work. Poonam Sharma contributed to implementation of the work using particle swam optimization and also purified and tuned the work done using genetic algorithm. Neeraj Tantubay contributed to improving the language of the paper as well as technical content of the paper.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of the IEEE Symposium on Security And Privacy*, pp. 197–213, May 2003.

[3] K. Chan and F. Fekri, "Node compromise attacks and network connectivity," in *Defense Transformation and Net-Centric Systems*, Proceedings of SPIE, Orlando, Fla, USA, April 2011, http://trove.nla.gov.au/work/34974076?citationFormat=BibTeX&selectedversion=NBD42326214.

[4] A. K. Mishra and A. K. Turuk, "Adversary information gathering model for node capture attack in wireless sensor networks," in *Proceedings of the IEEE International Conference on Devices and Communications (ICDeCom '11)*, pp. 1–5, Mesra, India, February 2011.

[5] T. Bonaci, L. Bushnell, and R. Poovendran, "Node capture attacks in wireless sensor networks: a system theoretic approach," in *Proceedings of the 49th IEEE Conference on Decision and Control (CDC '10)*, pp. 6765–6772, Atlanta, Ga, USA, December 2010.

[6] P. De, Y. Liu, and S. K. Das, "Modeling node compromise spread in wireless sensor networks using epidemic theory," in *Proceedings of the IEEE 7th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '06)*, pp. 237–243, June 2006.

[7] P. de, Y. Liu, and S. K. Das, "Deployment-aware modeling of node compromise spread in wireless sensor networks using epidemic theory," *ACM Transactions on Sensor Networks*, vol. 5, no. 3, pp. 1–33, 2009.

[8] G. Wu, X. Chen, M. S. Obaidat, and C. Lin, "A high efficient node capture attack algorithm in wireless sensor network based on route minimum key set," *Security and Communication Networks*, vol. 6, no. 2, pp. 230–238, 2013.

[9] C. Lin and G. Wu, "Enhancing the attacking efficiency of the node capture attack in WSN: a matrix approach," *The Journal of Supercomputing*, vol. 66, no. 2, pp. 989–1007, 2013.

[10] P. Tague and R. Poovendran, "Modeling adaptive node capture attacks in multi-hop wireless networks," *Ad Hoc Networks*, vol. 5, no. 6, pp. 801–814, 2007.

[11] P. Tague and R. Poovendran, "Modeling node capture attacks in wireless sensor networks," in *Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing*, pp. 1221–1224, Urbana, Ill, USA, September 2008.

[12] P. Tague, D. Slater, J. Rogers, and R. Poovendran, "Vulnerability of network traffic under node capture attacks using circuit theoretic analysis," in *Proceedings of the 28th IEEE International Conference on Computer Communications*, pp. 161–165, IEEE, Rio de Janeiro, Brazil, April 2009.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 126404, 17 pages
http://dx.doi.org/10.1155/2015/126404

Research Article

Multiobjective Particle Swarm Optimization Based on PAM and Uniform Design

Xiaoshu Zhu, Jie Zhang, and Junhong Feng

School of Computer Science and Engineering, Guangxi Universities Key Lab of Complex System Optimization and Big Data Processing, Yulin Normal University, Yulin 537000, China

Correspondence should be addressed to Jie Zhang; jgxyzjzj@126.com and Junhong Feng; jgxyfjh@126.com

Received 26 November 2014; Accepted 3 March 2015

Academic Editor: Pandian Vasant

In MOPSO (multiobjective particle swarm optimization), to maintain or increase the diversity of the swarm and help an algorithm to jump out of the local optimal solution, PAM (Partitioning Around Medoid) clustering algorithm and uniform design are respectively introduced to maintain the diversity of Pareto optimal solutions and the uniformity of the selected Pareto optimal solutions. In this paper, a novel algorithm, the multiobjective particle swarm optimization based on PAM and uniform design, is proposed. The differences between the proposed algorithm and the others lie in that PAM and uniform design are firstly introduced to MOPSO. The experimental results performing on several test problems illustrate that the proposed algorithm is efficient.

1. Introduction

Many real-world optimization problems often need to simultaneously optimize multiple objectives that are incommensurable and generally conflicting with each other. They can usually be written as

$$\min_{X\in\Omega} \{f_1(X), f_2(X), \ldots, f_M(X)\}, \quad (1)$$

where $X = (x_1, x_2, \ldots, x_N)$ is a variable vector in a real and N-dimensional space, Ω is the feasible solution space, and M is the number of the objective functions. Since the pioneering attempt of Schaffer [1] to solve multiobjective optimization problems, many kinds of multiobjective evolutionary algorithms (MOEAs), ranging from traditional evolutionary algorithms to newly developed techniques, have been proposed and widely used in different applications [2–4].

Multiobjective evolutionary algorithms, MOEAs, have become well-known methods for solving the multiobjective optimization problems that are too complex to be solved by exact methods. The main challenge for MOEAs is to be satisfied with three goals at the same time: (1) the Pareto optimal solutions are as near to true Pareto front, which means the convergence of MOEAs, (2) the nondominated solutions are evenly scattered along the Pareto front, which means the diversity of MOEAs, and (3) MOEAs obtain Pareto optimal solutions in limited evolution times [5].

The particle swarm optimization algorithm, PSO, and MOEAs are both intelligent optimization algorithms. It was proposed by Eberhart and Kennedy in 1995 [6, 7]. It originates from sharing and exchanging of information in the process of searching food among the bird's individuals. Each individual can benefit from the discovery and flight experience of the others. PSO seems particularly suitable for multiobjective optimization mainly because of the high speed of convergence [8, 9].

PAM is one of k-medoids clustering algorithms based on partitioning methods. It attempts to divide data objects into k partitions. Namely, it can divide a swarm into k different subswarms with different features.

This paper proposed a novel multiobjective particle swarm optimization based on PAM and uniform design, abbreviated as UKMOPSO. It first uses PAM to partition the data points into several clusters, and then the smallest cluster is implemented crossover operator based on the uniform design to generate some new data points. When the size of the Pareto solution is larger than the size of the external archive, PAM is used to determine which Pareto solution is to be

removed or appended. The results of the experimental simulation implemented on several well-known test problems indicate that the proposed algorithm is efficient.

The rest of this paper is organized as follows. Section 2 states the preliminaries of the proposed method. Section 3 presents our method in detail. Section 4 gives the numerical results of the proposed method. The conclusion of the work is made in Section 5.

2. Preliminaries

In this section, we describe some concepts concerning particle swarm optimization, multiobjective particle swarm optimization, PAM, and uniform design.

2.1. Particle Swarm Optimization Algorithms. In d-dimensional search space, the position and velocity of the ith particle are, respectively, represented as $X_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,d}]$ and $V_i = [v_{i,1}, v_{i,2}, \ldots, v_{i,d}]$. The optimal positions of the ith particle and the whole swarm, namely, the individual optimal and the global optimal, are denoted as $Pbest = [p_{i,1}, p_{i,2}, \ldots, p_{i,d}]$ and $Gbest = [p_{g,1}, p_{g,2}, \ldots, p_{g,d}]$, respectively. The individuals or particles in the swarm update their velocities and positions according to the following formulas:

$$v_{i,j}(k+1) = w \cdot v_{i,j}(k) + c_1 \cdot r_1 \cdot \left(Pbest_{i,j}(k) - x_{i,j}(k)\right) + c_2 \times r_2 \times \left(Gbest_j(k) - x_{i,j}(k)\right), \quad (2)$$

$$x_{ij}(k+1) = x_{ij}(k) + v_{ij}(k+1), \quad (3)$$

where the inertia weight coefficient w indicates the ability to maintain the previous speed; the acceleration coefficients c_1 and c_2 are used to coordinate the degrees of tracking individual optimal and global optimal; and r_1 and r_2 are two random numbers drawn from the uniform distribution on the interval $(0, 1)$.

The update equation of the velocity consists of the previous velocity component, a cognitive component and a social component. They are mainly controlled by three parameters: the inertia weight and two acceleration coefficients.

From the theoretical analysis for the trajectory of particles in PSO [10], the trajectory of a particle X_i converges to a weighted mean of P_i and P_g. Whenever the particle converges, it will "fly" to the individual best position and the global best position. According to the update equation, the individual best position of the particle will gradually move closer to the global best position. Therefore, all the particles will converge onto the global best particle's position.

2.2. Multiobjective Particle Swarm Optimization. MOPSO is proposed by Coello et al.; it adopts swarm intelligence to optimize MOPs, and it uses the Pareto optimal set to guide the particle's flight [9].

Particle swarm optimization has been proposed for solving a large number of single objective problems. Many researchers are interested in solving multiobjective problems (MOP) using PSO. To modify a single objective PSO to MOPSO, a guide must be redefined in order to obtain a set of nondominated solutions (Pareto front). In MOPSO, the Pareto optimal solutions should be used to determine the guide for each particle. How to select suitable local guides for attaining both convergence and diversity of solutions becomes an important issue.

There have been some publications to use PSO to solve MOP. A dynamic neighborhood PSO was proposed [11], which optimizes only one objective at a time and uses a scheme similar to lexicographic ordering. In addition, this approach also proposes an unconstrained elite archive named dominated tree to store the nondominated solutions. However, it is a difficult issue for this approach to pick up a best local guide from the set of Pareto optimal solutions for each particle of the population. A strategy for finding suitable local guides for each particle was proposed and named Sigma method. The local guide is explicitly assigned to specify particles according to the Sigma value [12]. This results in the desired diversity and convergence, but it is still not close enough to the Pareto front. On the other hand, an enhanced archiving technique to maintain the best (nondominated) solutions found during the course of a MO algorithm was proposed [13]. It shows that using archives in PSO for MO problems will improve their performance directly. A parallel vector evaluated particle swarm optimization (VEPSO) method for multiobjective problems was proposed [14], which adopted a ring migration topology and PVE system to simultaneously work 2–10 CPUs to find nondominated solutions. In [9], MOPSO method was proposed. It incorporates Pareto dominance and a special mutation operator to solve MO problems [15].

Recently, a hybrid multiobjective algorithm combining both genetic algorithm (GA) and particle swarm optimization (PSO) was proposed [16]. A multiobjective particle swarm optimization based on self-update and grid strategy was proposed for improving the function of Pareto set [17]. A new dynamic self-adaptive multiobjective particle swarm optimization (DSAMOPSO) method is proposed to solve binary-state multiobjective reliability redundancy allocation problems [18], which used a modified nondominated sorting genetic algorithm (NSGA-II) method and a customized time-variant multiobjective particle swarm optimization method to generate nondominated solutions.

The MOPSO method is becoming more popular due to its simplicity to be implemented and its ability to quickly converge to a reasonably acceptable solution for problems in science and engineering.

2.3. PAM Algorithm. There are many clustering methods available in data mining. Typical clustering analysis methods are clustering based on partition, hierarchical clustering, clustering based on density, clustering based on grid, and clustering based on model.

The most frequently used clustering methods based on partition are k-means and k-medoids. In contrast to the k-means algorithm, k-medoids chooses data points as centroids, which make k-medoids method more robust than k-means in the presence of noise and outliers. The reason is that k-medoids method is less influenced by outliers or other

Algorithm: PAM, a k-medoids algorithm for partitioning based on medoid or central objects.
Input:
k: the number of clusters,
D: a data set containing n objects.
Output: A set of k clusters.
Method:
(1) Arbitrarily choose k objects in D as the initial representative objects or seeds;
(2) Repeat
(3) Assign each remaining object to the cluster with the nearest representative object;
(4) Randomly select a nonrepresentative object, o random;
(5) Compute the total cost, S, of swapping representative object, o_j, with o random;
(6) if $S < 0$ then swap o_j with o random to form the new set of k representative objects;
(7) Until no change;

Algorithm 1: PAM algorithm.

extreme values than k-means. PAM (Partitioning Around Medoids) is the first and the most frequently used k-medoids algorithms. It is shown in Algorithm 1.

PAM constructs k partitions (clusters) of the given dataset, where each partition represents a cluster. Each cluster may be represented by a centroid or a cluster representative which is some sort of summary description of all the objects contained in a cluster. It needs to determine k partitions for n objects. The process of PAM is by and large as follows. Firstly, randomly select k representative objects, and cluster other objects to the same group as the representative object according to the minimum distances between the representative object and other objects. Then try to replace these representative objects with other nonrepresentative objects in order to minimize squared error. All the possible pairs of objects are analyzed, where one object in each pair is considered a representative object and the other is not. The total cost of the clustering is calculated for each such combination. An object will be replaced with such an object having minimized squared error. The set of best objects in each cluster after iteration forms the representative objects for the next iteration. The final set of representative objects is the respective centroids of the clusters [19–22].

2.4. Uniform Design

2.4.1. Uniform Design. The main objective of uniform design is to sample a small set of points from a given set of points such that the sampled points are uniformly scattered [23].

Let there be n factors and q levels per factor. When n and q are given, the uniform design selects q from q^n possible combinations, such that these q combinations are uniformly scattered over the space of all possible combinations. The selected q combinations are expressed as a uniform array $U(n, q) = [U_{i,j}]q \times n$, where $U_{i,j}$ is the level of the jth factor in the ith combination and can be calculated by the following formula

$$U_{i,j} = \left(i\sigma^{j-1} \bmod q\right) + 1, \tag{4}$$

where σ is a parameter given in Table 1.

2.4.2. Improved Generation of Initial Population. An algorithm for dividing the solution space and an algorithm for generating the initial population have been designed [23]. However, Algorithm 2 in [23] considers only the dividing of the solution space, not the dividing of the N-dimension space. This will bring about some serious problems. If we assume that, in Step 2, $Q_0 = 5$ and $N = 10$, then $U(N, Q_0)$ is impossible to be generated because Q_0 must be larger than N. Namely, Algorithm 2 in [23] is only suitable for the low dimensional problem. In order to overcome the shortcomings, the dividing of the N-dimension space is introduced to improve the algorithm. The improved algorithm can be suitable for not only low dimensional but also high dimensional problem. The improved algorithm is shown as follows.

Algorithm A (improved generation of initial population)

Step 1. Judge whether Q_0 is valid or not, as it must be found in the 1st column of Table 1. If it is not valid, it stops and shows error messages, otherwise it continues.

Step 2. Execute Algorithm 1 in [23] to divide $[l, u]$ into S subspaces $[l(1), u(1)], [l(2), u(2)] \cdots [l(S), u(S)]$.

Step 3.1. Judge whether Q_0 is more than N, if yes, turn to Step 3.2, otherwise turn to Step 3.3.

Step 3.2. Quantize each subspace, and apply the uniform array $U(N, Q_0)$ to sample Q_0 points.

Step 3.3. Divide N-dimension space into $\lfloor N/N_0 \rfloor$ parts, where N_0 is an integer more than 1 and less than Q_0 and is generally taken as Q_0-1. Among $\lfloor N/N_0 \rfloor$ parts, the 1st part corresponds to the dimension from 1 to N_0, and the 2nd part corresponds to the dimension from $N_0 + 1$ to $2 * N_0$, and so forth. If the remainder R of N/N_0 is not equal to 0, then a plus part corresponds to the dimension from $\lfloor N/N_0 \rfloor * N_0 + 1$ to N, whose length is surely less than N_0. Repeat to execute Step 3.4 for each part.

Step 3.4. Quantize each subspace, and then apply uniform array $U(N_0, Q_0)$ to sample Q_0 points. It is noteworthy that

Table 1: Values of the parameter σ for different number of factors and different number of levels per factor.

Number of levels per factors	Number of factors	σ
5	2~4	2
7	2~6	3
11	2~10	7
13	2	5
	3	4
	4~12	6
17	2~16	10
19	2~3	8
	4~18	14
23	2, 13~14, 20~22	7
	8~12	15
	3~7, 15~19	17
29	2	12
	3	9
	4~7	16
	8~12, 16~24	8
	13~15	14
	25~28	18
31	2, 5~12, 20~30	12
	3~4, 13~19	22

$U(N_0, Q_0)$ is replaced with $U(R, Q_0)$ in the plus part, where the remainder R is equal to $N - \lfloor N/N_0 \rfloor * N_0$.

2.4.3. Crossover Operator Based on the Uniform Design. The crossover operator based on the uniform design acts on two parents. It quantizes the solution space defined by these parents into a finite number of points, and then it applies the uniform design to select a small sample of uniformly scattered points as the potential offspring.

For any two parents p_1 and p_2, their minimal values and the maximal values of each dimension are used to form the novel solution space $[l_{\text{parent}}, u_{\text{parent}}]$. Each domain of $[l_{\text{parent}}, u_{\text{parent}}]$ is quantized into Q_1 levels, where Q_1 is a predefined prime number. Then the uniform design is applied to select a sample of points as the potential offspring. The details of the algorithm can be referred to [23].

3. The Proposed Algorithm

3.1. Elitist Selection or Elitism [4]. Elitism means that elite individuals cannot be excluded from the mating pool of the population. A strategy presented can always include the best individual of the current population in the next generation in order to prevent the loss of good solutions found so far. This strategy can be extended to copy the best n individuals to the next generation. This is explanation of the elitism. In MOP, elitism plays an important role.

Two strategies are often used to implement elitism. One maintains elitist solutions in the population; the other stores elitist solutions into an external secondary list or external archive and reintroduces them to the population. The former copies all nondominated solutions in the current population to the next population and then fills the rest of the next population by selecting from the remaining dominated solutions in the current population. The latter uses an external secondary list or external archive to store the elitist solutions. External list stores the nondominated solutions found. It will be updated in the next generation by means of removing elitist solutions dominated by a new solution or adding the new solution if it is not dominated by any existing elitist solution.

```
population paretocreate (pop)
Input: pop indicates the population.
Output: pareto indicates the non-dominated solutions.
  pareto ← pop;
  for ∀ chr1 ∈ pop;
    for ∀ chr2 ∈ pareto ∧ chr2 ≠ chr1;
      if chr1 dominate chr2
        pareto ← pareto − chr2;
      end if
    end for
  end for
  return pareto;
```

Algorithm 2: Pseudocode of selecting elitist.

The work adopts the second strategy, namely, storing elitist solutions to an external secondary list. Its advantage is that it can preserve and dynamically adjust all the nondominated solutions set till the current generation. The pseudocodes of selecting elitist and updating elitist are, respectively, shown in Algorithms 2 and 3.

3.2. Selection Mechanism for the Swarm Based on Uniform Design. This paper adopts the uniform design to select the best T_1 points ($T_1 < T$) from the T points and acquires their objectives fit$V$. The detailed steps are as follows.

Algorithm B

Step 1. Calculate each of M objectives for each of the T points; normalize each of objectives $f_i(x)$ as follows:

$$h_i x = \frac{f_i(x)}{\max_{y \in \psi} \{|f_i(y)|\}}, \tag{5}$$

where ψ is a set of points in the current population and $h_i x$ is the normalized objective.

Step 2. Apply the uniform design to generate the D_0 weight vectors $w_1, w_2, \ldots, w_{D_0}$; each of them is used to compose one fitness function by the following formula, where D_0 is a design parameter and it is prime:

$$\text{fit}_i = w_{i,1} h_1(x) + w_{i,2} h_2(x) + \cdots + w_{i,D_0} h_{D_0}(x), \quad 1 \le i \le D_0. \tag{6}$$

```
population paretoupdate(offspring, pareto)
Input: offspring indicates the offsprings after performing the crossover operator
       and mutation operator; pareto indicates the non-dominated solutions.
Output: pareto indicates the non-dominated solutions.
  offspring ← call paretocreate(offspring);
  for ∀ chr1 ∈ offspring;
    nondominated ← true;
    for ∀ chr2 ∈ pareto;
      if chr1 dominate chr2
        pareto ← pareto − chr2;
      else if chr2 dominate chr1
        nondominated ← false;
      else
        continue
      end if
    end for
    if nondominated
      pareto ← pareto ∪ chr1
    end if
  end for
  return pareto;
```

Algorithm 3: Pseudocode of updating elitist.

Step 3. Based on each of fitness functions, evaluate the quality of the T points. Assume the remainder T_1/D_0 is R_0. For the first R_0 fitness functions, select the best $\lceil T_1/D_0 \rceil$ points; for the rest of fitness functions, select the best $\lfloor T_1/D_0 \rfloor$ points. Overall, a total of T_1 points are selected. The objectives of these selected points are correspondingly stored into fit$V$.

3.3. Selection Mechanism for Gbest Based on PAM. In MOPSO, *Gbest* plays an important role in guiding the entire swarm toward the global Pareto front [24]. In contrast to the single objective PSO having only one global best *Gbest*, MOPSO has multiple Pareto optimality solutions which are nondominated each other. How to select a suitable *Gbest* for each of particles from Pareto optimal solutions is one very key issue.

The paper presents the selection mechanism for *Gbest* based on PAM as follows.

Algorithm C. Assume the population and the numbers of particles are denoted as pop and Npop, and the Pareto optimality and the numbers of Pareto optimality are denoted as pareto and NP.

Step 1. Acquire the number of cluster according to the following formula:

$$K = K_{\min} + (K_{\max} - K_{\min}) \cdot \left(\frac{t}{\max G} \right), \tag{7}$$

where $K_{\min}$ and $K_{\max}$, respectively, denote the minimal and maximal value of K, namely, $K \in [K_{\min}, K_{\max}]$; t and $\max G$ indicate the tth iteration and the maximal iteration number. The formula can acquire the linearly increasing number of cluster so as to accord with the process from the coarse search to the elaborate search.

Step 2. If $NP \leq K$, then for each particle in pop, find the nearest Pareto optimal solution as its *Gbest*. Otherwise, turn to Step 3.

Step 3. Perform PAM described in Section 2.3 to partition Pareto and pop into K clusters, respectively, the cluster centroids of which are denoted as $C_1 = \{c_{1,1}, c_{1,2}, \ldots, c_{1,K}\}$ and $C_2 = \{c_{2,1}, c_{2,2}, \ldots, c_{2,K}\}$.

Step 4. For each $c_{2,j} \in C_2$, find the nearest $c_{1,i} \in C_1$. For all the particles in the cluster represented by $c_{2,j}$, their *Gbest* randomly takes one of Pareto optimal solutions in the cluster represented by $c_{1,i}$.

3.4. Adjustment of Pareto Optimal Solutions Based on the Uniform Design and PAM. The number of Pareto optimal solutions in the external archive will increasingly enlarge with evolution; namely, the size of the external archive will become very large with the iteration number. This will increase the computation complexity and the execution time of an algorithm. Therefore, the size of the external archive must be controlled. In the meanwhile, Pareto optimal solutions in the external archive are possibly close to each other in the objective space. If the solutions are close to each other, they are nearly the same choice. It is desirable to find the Pareto optimal solutions scattered uniformly over the Pareto front, so that the decision maker can have a variety of choices. Therefore, how to control the size of the external archive and select representative Pareto optimal solutions scattered uniformly over the Pareto front is a key issue.

The paper presents the adjustment algorithm of Pareto optimal solutions based on the uniform design and PAM. The algorithm firstly implements PAM to partition the Pareto front into K clusters in the objective space and then

implements the uniform crossover operator on the minimal cluster so as to generate more Pareto optimality in the minimal cluster; finally it keeps all the points in the lesser clusters and discards some points in the larger clusters. The detailed steps of the algorithm are shown as follows.

Algorithm D. Assume the Pareto optimality, their values of M functions, and the numbers of Pareto optimality are denoted as pareto, Fpareto, and NP. The size of the external archive is assumed as $NP\lim$. The number of clusters is K.

Step 1. If the numbers of data points in the maximal and the minimal clusters have too many differences or are both very small, select all the data points from the minimal cluster and the same number of points from the cluster which is the nearest to the minimal cluster. For each pair of data points, perform the uniform crossover operator described in Section 2.4.3. Update pareto, Fpareto, and NP according to Algorithm 3.

Step 2. If $NP > NP\lim$, implement PAM to partition the Fpareto into K clusters in the objective space and let $N\,\text{div} = NP\lim/K$; it indicates the average points to select from each of K clusters. We can classify three situations according to the number of points in each cluster and N div as follows.

(i) Situation 1: the numbers of points in all clusters are larger than or equal to N div.

(ii) Situation 2: the number of data points is larger than or equal to N div only in one cluster.

(iii) Situation 3: the number of the clusters, in which the number of data points is larger than N div, is larger than 1 and less than K.

Step 3. For Situation 1, sort all clusters according to their containing points in ascending order. Select N div points from the previous $K-1$ clusters, and select the remainder points, $NP\lim - N\,\text{div} * (K-1)$, from the last cluster. Turn to Step 7.

Step 4. For Situation 2, keep the points of all the clusters, in which the number of the points is less than or equal to N div, and the remainder points are selected from that only cluster. Turn to Step 7.

Step 5. For Situation 3, keep all the points of all the clusters, in which the number of points is less than or equal to N div. Turn to Step 6 to select the remainder point.

Step 6. Recalculate N div, and let $N\,\text{div} = \text{rem}P/K'$, where rem$P$ and K' indicate the number of the remainder points and clusters. According to the new N div, turn to Step 3 or Step 4.

Step 7. Save the selected $NP\lim$ points and terminate the algorithm.

3.5. Thoughts on the Proposed Algorithm. In MOP, for M objectives $f_1, f_2, \ldots, f_M$ and any two points x and y in the feasible solution space Ω, if each objective is satisfied with $f_i(x) \le f_i(y)$ and at least one objective is satisfied with $f_i(x) < f_i(y)$, namely, x is at least as good as y with respect to all the objectives, and x is strictly better than y with respect to at least one objective, then we say that x dominates y. If no other solution is strictly better than x, then x is called a nondominated solution or Pareto optimal solution.

In single objective problems, there is only one objective to be optimized. Therefore, the global best (*Gbest*) of the whole swarm is only one. In multiobjective problems, there are multiple consistent or conflicting objectives to be optimized. Therefore, there exist very large or infinite numbers of solutions which cannot dominate each other. These solutions are nondominated solutions or Pareto optimal solutions. Therefore, *Gbest* of the whole swarm is more than one. How to select suitable *Gbest* is a very key issue.

When it is not possible to find all these solutions, it may be desirable to find as many solutions as possible in order to provide more choices to the decision maker. However, if the solutions are close to each other, they are nearly of the same choice. It is desirable to find the Pareto optimal solutions scattered uniformly over the Pareto front, so that the decision maker can have a variety of choices. The paper introduces uniform design to ensure that the acquired solutions scatter uniformly over the Pareto front in the objective space.

For the MOPSO, the diversity of the population is a very important factor. It has a key impact on the convergence of an algorithm and uniformly distribution of Pareto optimal solution. It can effectively get rid of the premature of the algorithms. The paper introduces PAM clustering algorithms to maintain the diversity of the population. The particles in the same cluster have similar features, whereas ones in different clusters have dissimilar features. Thus, choosing particles from different clusters may necessarily increase the diversity of the population.

3.6. Steps of the Proposed Algorithm

Step 1. According to the population size Npop, determine the number of subintervals S and the population size Q_0 in each subinterval, such that $Q_0 * S$ is more than Npop; Q_0 is a prime and must exist in Table 1. Execute Algorithm A in Section 2.4.2 to generate a temporary population containing $Q_0 * S \ge N\text{pop}$ points.

Step 2. Perform Algorithm B described in Section 3.2 to select the best Npop points from the $Q_0 * S$ points as the initial population pop and acquire their objectives fit$V$.

Step 3. Initialize the speed and *Pbest* of the particle i by $V[i] = \text{pop}[i]$ and $Pbest[i] = \text{pop}[i]$; $fPbest[i] = \text{fit}V[i]$ where $i = 1 \cdots N\text{pop}$; $fPbest[i]$ denotes objectives of $Pbest[i]$.

Step 4. According to Algorithm 2, select elitist or Pareto optimal solutions from pop and store them to the external secondary list, pareto, the size of which is assumed as NP.

Step 5. Choose the suitable *Gbest* for each of the particles from pareto in terms of Algorithm C described in Section 3.3.

Step 6. Update the position pop and velocity V for each of the particles, respectively, using formulas (2) and (3), where formula (2) is modified as follows:

$$v_{i,j}(k+1) = w \cdot v_{i,j}(k) + c_1 \cdot r_1 \cdot \left(Pbest_{i,j}(k) - x_{i,j}(k)\right) + c_2 \times r_2 \times \left(Gbest_{i,j}(k) - x_{i,j}(k)\right). \quad (8)$$

Step 7. For the jth dimensional variable of the ith particle, if it goes beyond its lower boundary or upper boundary, then it takes the jth dimensional value of its corresponding boundary and the jth dimensional value of its velocity takes the opposite value.

Step 8. Calculate each of M objectives for each of the particles, and update them into fit$V$.

Step 9. Update *Pbest* of each particle as follows.

For the ith particle, if fit$V[i]$ dominates $fPbest[i]$, then let $fPbest[i]$ = fit$V[i]$ and $Pbest[i]$ = pop$[i]$; otherwise, $Pbest[i]$ and $fPbest[i]$ are kept unchanged. If neither of them is dominated by the other, then randomly select one of them as $fPbest[i]$ and update its corresponding $Pbest[i]$.

Step 10. Update the external archive pareto storing Pareto optimality and its size *NP* according to Algorithm 3.

Step 11. Implement Algorithm D described in Section 3.4 to adjust the Pareto optimal solutions such that the number of Pareto optimal solutions is less than or equal to the size of the external archive *NP* lim.

Step 12. If the stop criterion is satisfied, terminate algorithm; otherwise, turn to Step 5 and continue.

4. Numerical Results

In order to evaluate the performances of the proposed algorithm, we compare it with two outstanding algorithms, UMOGA [23] and NSGA-II [25].

4.1. Test Problems. Several well-known multiobjective test functions are used to test the performance of the proposed algorithm.

Biobjective problem: FON [25, 26], KUR [25, 27], ZDT1, ZDT2, ZDT3, and ZDT6 [25, 28–31].

Three-objective problems: DTLZ1 and DTLZ [2, 32].

The definitions of them are as follows.

FON is defined as

$$f_1(x) = 1 - \exp\left(-\sum_{i=1}^{3}\left(x_i - \frac{1}{\sqrt{3}}\right)^2\right),$$
$$f_2(x) = 1 - \exp\left(-\sum_{i=1}^{3}\left(x_i + \frac{1}{\sqrt{3}}\right)^2\right), \quad (9)$$

where $x_i \in [-4, 4]$.

This test function has a nonconvex Pareto optimal front.

KUR is defined as

$$f_1(x) = \sum_{i=1}^{n-1}\left(-10\exp\left(-0.2 \cdot \sqrt{x_i^2 + x_{i+1}^2}\right)\right),$$
$$f_2(x) = \sum_{i=1}^{n}\left(|x_i|^{0.8} + 5 \cdot \sin x_i^3\right), \quad (10)$$

where $n = 3$ and $x_i \in [-5, 5]$.

This test function has a nonconvex Pareto optimal front, in which there are three discontinuous regions.

ZDT1 is defined as

$$f_1(x_1) = x_1,$$
$$g(x_2, \ldots, x_n) = 1 + \frac{9\sum_{i=2}^{n} x_i}{(n-1)}, \quad (11)$$
$$h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}},$$

where $n = 30$ and $x_i \in [0, 1]$.

This test function has a convex Pareto optimal front.

ZDT2 is defined as

$$f_1(x_1) = x_1,$$
$$g(x_2, \ldots, x_n) = 1 + \frac{9\sum_{i=2}^{n} x_i}{(n-1)}, \quad (12)$$
$$h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^2,$$

where $n = 30$ and $x_i \in [0, 1]$.

This test function has a nonconvex Pareto optimal front.

ZDT3 is defined as

$$f_1(x_1) = x_1,$$
$$g(x_2, \ldots, x_n) = 1 + \frac{9\sum_{i=2}^{n} x_i}{(n-1)}, \quad (13)$$
$$h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} - \left(\frac{f_1}{g}\right)\sin(10\pi f_1),$$

where $n = 30$ and $x_i \in [0, 1]$.

This test function represents the discreteness feature. Its Pareto optimal front consists of several noncontiguous convex parts. The introduction of the sine function in h causes discontinuity in the Pareto optimal front.

ZDT6 is defined as

$$f_1(x_1) = 1 - \exp(-4x_1)\sin^6(6\pi x_1),$$
$$g(x_2, \ldots, x_n) = 1 + 9\left(\frac{\sum_{i=2}^{n} x_i}{(n-1)}\right)^{0.25}, \quad (14)$$
$$h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^2,$$

where $n = 10$ and $x_i \in [0, 1]$.

This test function has a nonconvex Pareto optimal front. It includes two difficulties caused by the nonuniformity of the search space. Firstly, Pareto optimal solutions are nonuniformly distributed along the global Pareto optimal front (the front is biased for solutions in which $f_1(x_1)$ is near one). Secondly, the density of the solutions is the lowest near the Pareto optimal front and the highest away from the front.

DTLZ1 is defined as

$$f_1(x) = 0.5x_1x_2(1 + g(x)),$$
$$f_2(x) = 0.5x_1(1 - x_2)(1 + g(x)),$$
$$f_3(x) = 0.5(1 - x_1)(1 + g(x)),$$
$$g(x) = 100 \cdot \left[5 + \sum_{i=3}^{n}\left((x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\right)\right], \quad (15)$$

where $n = 7$ and $x_i \in [0, 1]$.

This test function has difficulty in the convergence to the Pareto optimal hyperplane and contains $(11^5 - 1)$ local Pareto optimal fronts.

DTLZ2 is defined as

$$f_1(x) = (1 + g(x))\cos(0.5\pi x_1)\cos(0.5\pi x_2),$$
$$f_2(x) = (1 + g(x))\cos(0.5\pi x_1)\sin(0.5\pi x_2),$$
$$f_3(x) = (1 + g(x))\sin(0.5\pi x_1), \quad (16)$$
$$g(x) = \sum_{i=3}^{n}(x_i - 0.5)^2,$$

where $n = 12$ and $x_i \in [0, 1]$.

The Pareto optimal solutions of this test function must lie inside the first octant of the unit sphere in a three-objective plot. It is more difficult than DTLZ1.

4.2. Parameter Values. The parameter values of the proposed algorithm, UKMOPSO, are adopted as follows.

(i) *Parameters for PSO*: the linearly descending inertia weight [33, 34] is within the interval [0.1, 1], and the acceleration coefficients c_1 and c_2 are both taken as 2.

(ii) *Parameter for PAM*: the minimal and maximal numbers of clusters are $K_{\min} = 2$ and $K_{\max} = 10$.

(iii) *Population size*: the population size Npop is 200.

(iv) *Parameters for the uniform design*: the number of subintervals S is 64; the number of the sample points or the population size of each subinterval Q_0 is 31; set $D_0 = 31$ and $Q_1 = 5$.

(v) *Stopping condition*: the algorithm terminates if the number of iterations is larger than the given maximal generations of 20.

All the parameter values in UMOGA [23] are set equal to the original values in [23]; the different and additional parameter values between UMOGA and UKMOPSO are as follows: the number of subintervals S is 16; $F = N$ (the number of variables); $D_1 = 7$; $p_m = 0.02$.

NSGA-II in [25] adopted a population size of 100, a crossover probability of 0.8, a mutation probability of $1/n$ (where n is the number of variables), and maximal generations of 250. In order to make the comparisons fair, we have used population size of 100 and maximal generations of 40 in NSGA-II, so that the total number of function evaluations in NSGA-II, UKMOPSO, and UMOGA is the same.

4.3. Performance Metric. Based on the assumption that the true Pareto front of a test problem is known, many kinds of performance metrics have been proposed and used by many researchers such as [2, 9, 24, 25, 29, 35–40]. Three of them are adopted in the paper to compare the performance of the proposed algorithm with them. The first metric is C metric [29, 38]. It is taken as the quantitative metric of the solution quality and often used to show that the outcomes of one algorithm dominating the outcomes of another algorithm. It is defined as

$$C(A, B) = \frac{|b \in B : \exists a \in A : a \prec b|}{|B|}, \quad (17)$$

where $a \prec b$ represents b being dominated by a; A represents the Pareto front obtained by Algorithm A, and B represents the Pareto front obtained by Algorithm B in a typical run. Furthermore, $|B|$ represents the number of elements of B.

The metric value $C(A, B) = 1$ means that all points in B are dominated by or equal to points in A. In contrast, $C(A, B) = 0$ means that none of the points in B are covered by the ones in A.

The second metric is the IGD metric (namely, Inverted Generational Distance) [39–41]. It is the mean value of the distances from the set of solutions uniformly distributed in the true Pareto front to the set of solutions obtained by an algorithm in objective space. Let P^* be a set of uniformly distributed points along the true PF (Pareto Front). Let A be an approximate set to the PF; the average distance from P^* to A is defined as

$$\mathrm{IGD}(A, P^*) = \frac{\sum_{v \in P^*} d(v, A)}{|P^*|}, \quad (18)$$

where $d(v, A)$ is the minimum Euclidean distance between v and the points in A. If $|P^*|$ is large enough to represent the PF very well, $\mathrm{IGD}(A, P^*)$ could measure both the diversity and convergence of A in a sense. To have a low value of $\mathrm{IGD}(A, P^*)$, the set A must be very close to the PF and cannot miss any part of the whole PF.

The smaller the IGD value for the set of obtained solutions is, the better the performance of the algorithm will be.

The third measure is the measure of maximum spread (MS) [2, 24, 35], which is proposed in [42, 43]. It can measure how well the true Pareto front ($\mathrm{PF}_{\mathrm{true}}$) is covered by the discovered Pareto front ($\mathrm{PF}_{\mathrm{known}}$) through hyperboxes

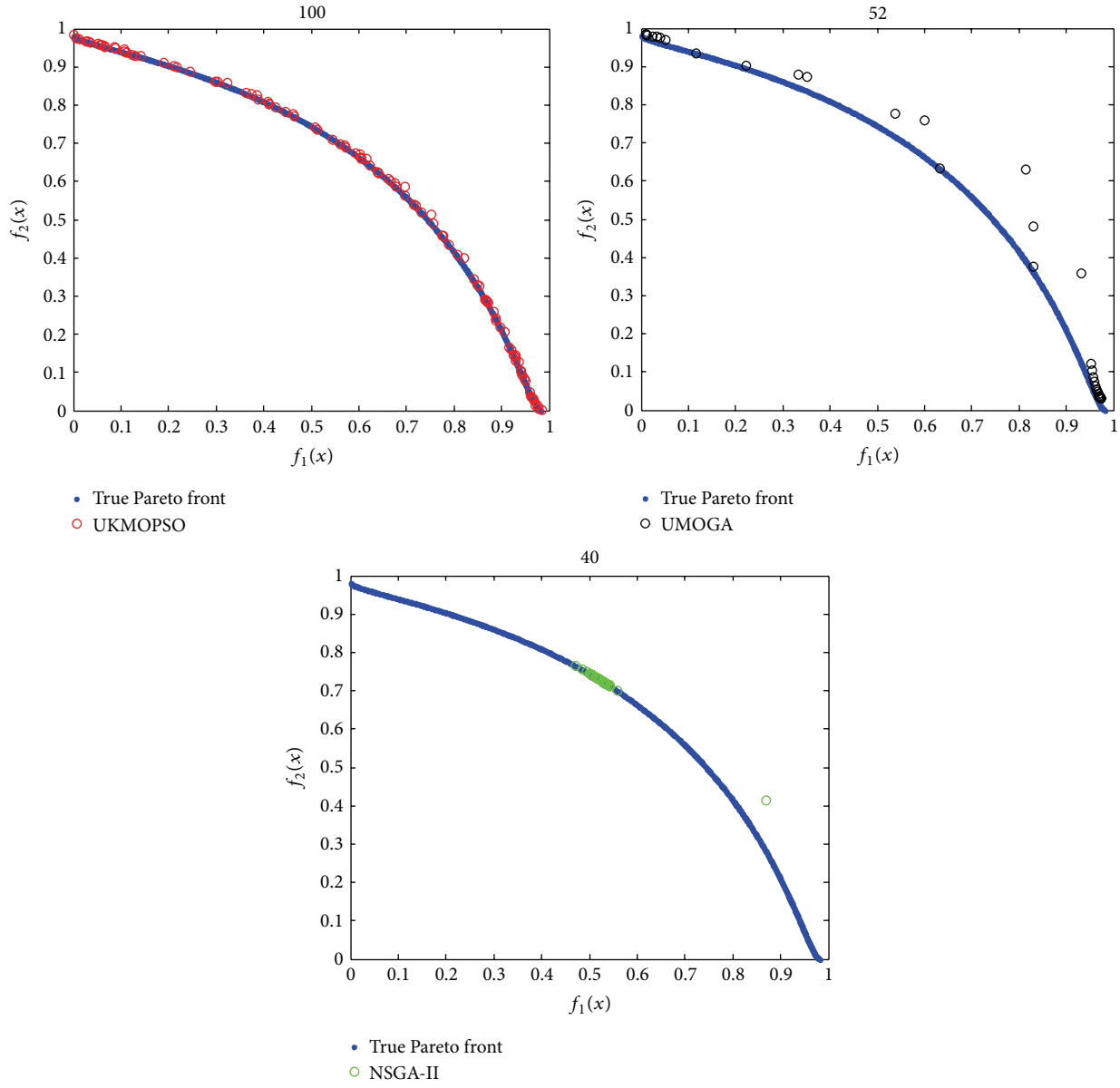


FIGURE 1: Pareto fronts obtained by different algorithms for test problem FON.

formed by the extreme function values observed in the PF_{true} and PF_{known}. It is defined as

$$MS = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\delta_i},$$
$$\delta_i = \left(\frac{\min\left(f_i^{max}, F_i^{max}\right) - \max\left(f_i^{min}, F_i^{min}\right)}{F_i^{max} - F_i^{min}}\right)^2, \quad (19)$$

where M is the number of objectives, f_i^{max} and f_i^{min} are the maximum and minimum values of the ith objective in PF_{known}, respectively, and F_i^{max} and F_i^{min} are the maximum and minimum values of the ith objective in PF_{true}, respectively.

Note that if $f_i^{min} \geq F_i^{max}$, then $\delta_i = 0$. Algorithms with larger MS values are desirable and MS = 1 means that the true Pareto front is totally covered by the obtained Pareto front.

4.4. Results. For each test problem, we perform the proposed algorithm (called UKMOPSO) for 30 independent runs and compare its performance with UMOGA [23] and NSGA-II [25]. The values of several metrics, the C metric, IGD metric, and MS metric, are shown in Tables 2, 3, 4, and 5. The Pareto fronts obtained by several algorithms implemented on several test functions are illustrated in Figures 1–6.

For brevity, C(UKMOPSO, UMOGA), C(UMOGA, UKMOPSO), C(UKMOPSO, NSGA-II), and C(NSGA-II, UKMOPSO) are, respectively, marked as C_{12}, C_{21}, C_{13}, and C_{31}.

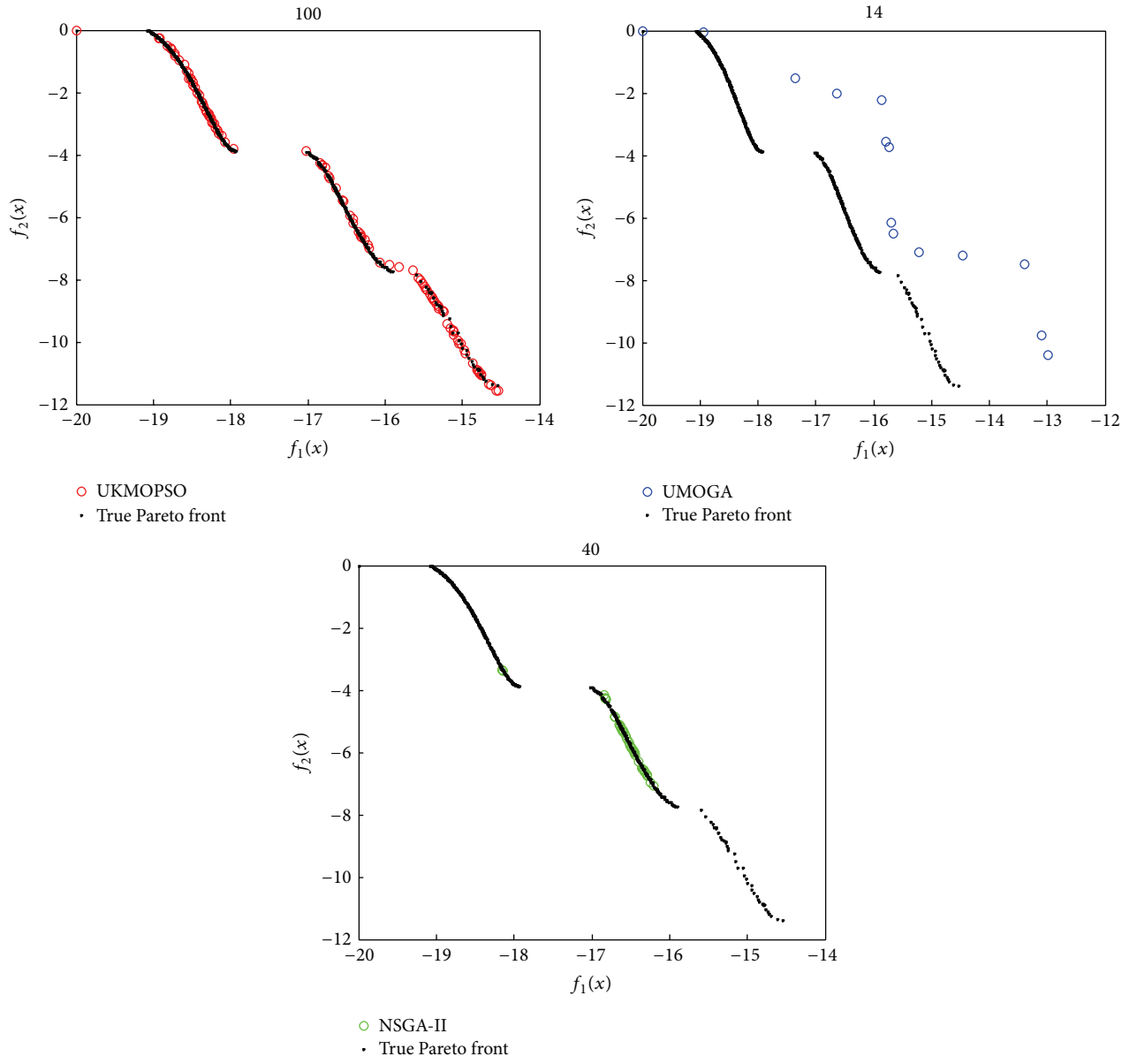


Figure 2: Pareto fronts obtained by different algorithms for test problem KUR.

Table 2: Comparison of C metric between UKMOPSO and UMOGA.

	C (UKMOPSO, UMOGA)	C (UMOGA, UKMOPSO)
FON	0.769	0.01
KUR	0.929	0.01
ZDT1	0.12	0.13
ZDT2	0.063	0.03
ZDT3	0.139	0.098
ZDT6	0.063	0.01
DTLZ1	1	0
DTLZ2	1	0

Table 3: Comparison of C metric between UKMOPSO and NSGA-II.

	C (UKMOPSO, NSGA-II)	C (NSGA-II, UKMOPSO)
FON	0.025	0.01
KUR	0.075	0.11
ZDT1	1	0
ZDT2	1	0
ZDT3	1	0
ZDT6	1	0
DTLZ1	0	0.13
DTLZ2	0.025	0.06

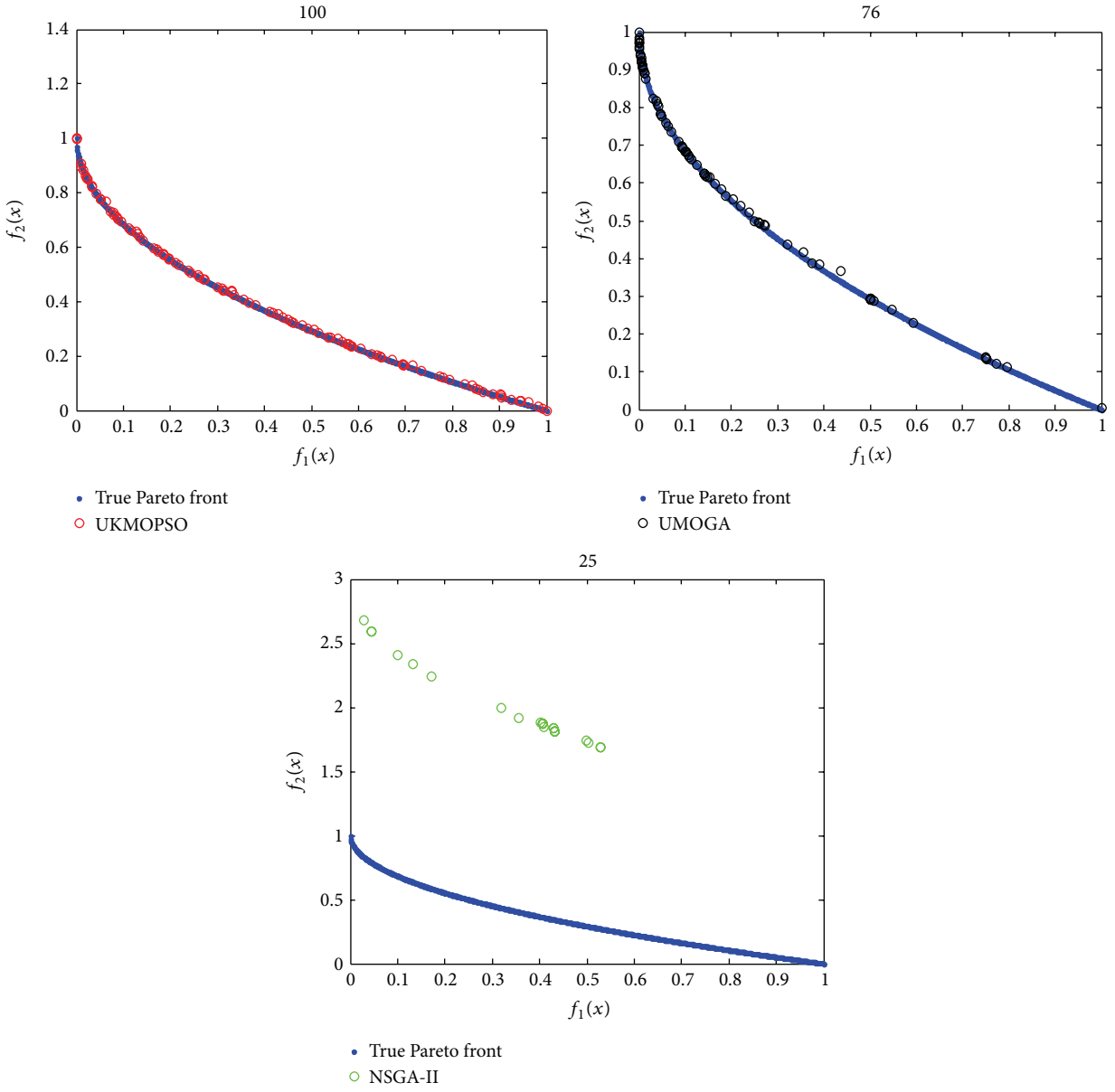


Figure 3: Pareto fronts obtained by different algorithms for test problem ZDT1.

Table 4: Comparison of IGD metric.

	UKMOPSO	UMOGA	NSGA-II
FON	0.0072	0.0432	0.3707
KUR	0.0668	1.1576	1.2323
ZDT1	0.0067	0.0265	1.3933
ZDT2	0.0067	0.0727	2.1409
ZDT3	0.1964	0.2151	0.2816
ZDT6	0.0033	0.0139	5.9223
DTLZ1	7.4827	22.965	5.4182
DTLZ2	0.1123	0.4319	0.2724

Table 5: Comparison of maximum spread (MS) metric.

	UKMOPSO	UMOGA	NSGA-II
FON	0.9769	0.9754	0.2368
KUR	0.9789	0.9580	0.3414
ZDT1	0.9429	0.9984	0.5495
ZDT2	0.9999	0.9986	0.0019
ZDT3	0.9276	0.9275	0.7963
ZDT6	1	0.9974	0.3738
DTLZ1	1	1	0.6588
DTLZ2	1	1	0.7125

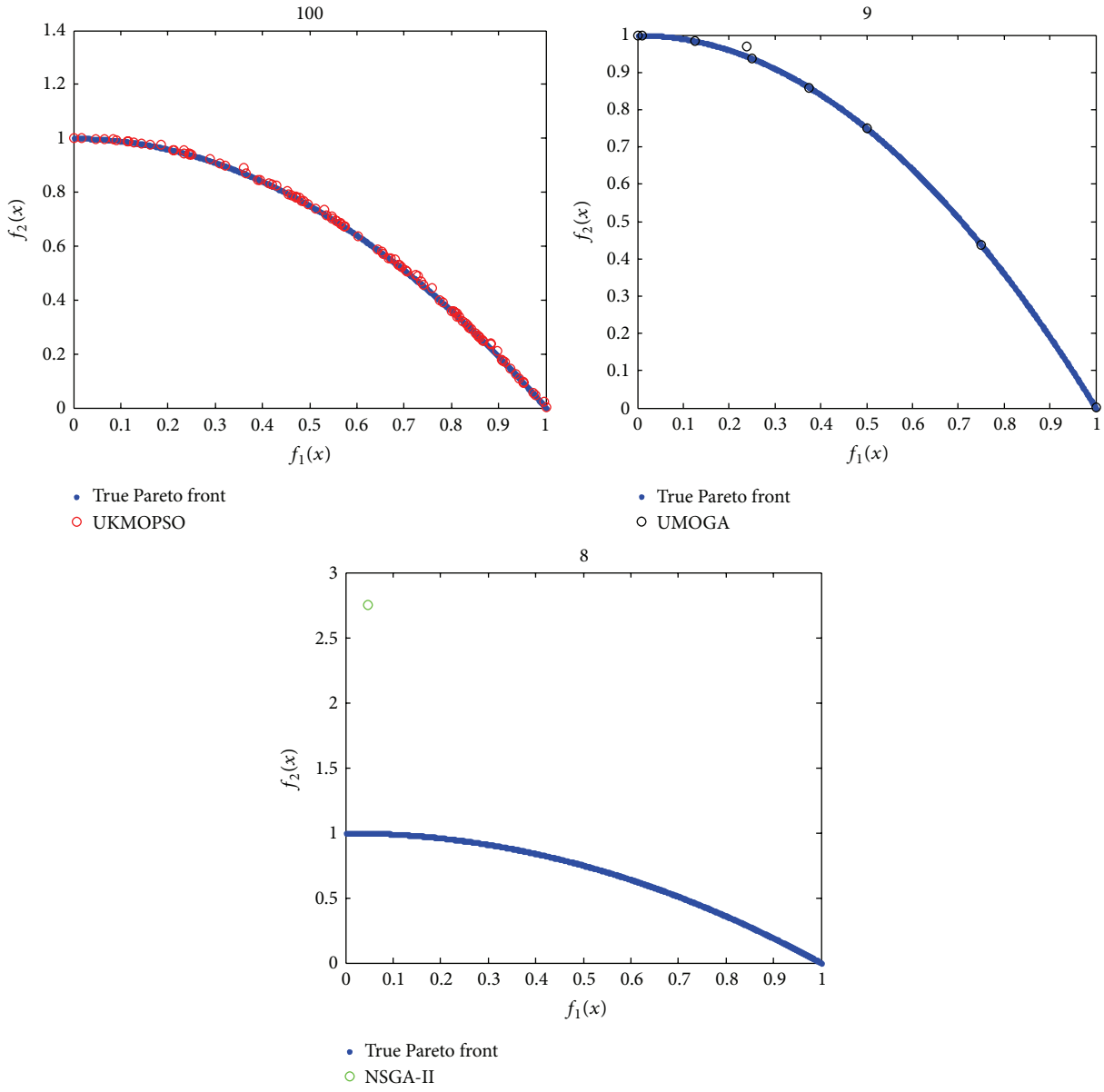


Figure 4: Pareto fronts obtained by different algorithms for test problem ZDT2.

As shown in Table 2, for the test functions, Fon and KUR, $C_{12} = 0.769$ and $C_{12} = 0.929$ mean that 76.8% and 92.9% of the solutions obtained by UMOGA are dominated by those obtained by UKMOPSO. Similarly, $C_{21} = 0.01$ means that 1% of the solutions obtained by UKMOPSO are dominated by those obtained by UMOGA. For DTLZ1 and DTLZ2, $C_{12} = 1$ and $C_{21} = 0$ mean that all solutions obtained by UMOGA are dominated by those obtained by UKMOPSO, but none of solutions from UKMOPSO is dominated by those from UMOGA. It means that the solution quality via UKMOPSO is much better than that via UMOGA for the above test functions. For ZDT1, ZDT2, ZDT3, and ZDT6, the values of the C metrics do not have too many differences between UKMOPSO and UMOGA. It means the solution qualities via both are almost identical.

From Table 3, we can see that for ZDT1, ZDT2, ZDT3, and ZDT6, $C_{13} = 1$ and $C_{31} = 0$ mean that all solutions obtained by NSGA-II are dominated by those obtained by UKMOPSO, but none of solutions from UKMOPSO is dominated by those from NSGA-II. It means that the solution quality via UKMOPSO is much better than that via NSGA-II for the above test functions. For the rest of the test functions, the solution qualities via both are almost identical.

In Table 4, for all test problems, the IGD values of UKMOPSO are the smallest among UKMOPSO, UMOGA, and NSGA-II. This means the PF found by UKMOPSO is the nearest to the true PF compared with the PF obtained by the other two algorithms; namely, the performance of UKMOPSO is the best in the three algorithms. The IGD values of UMOGA are all larger than those of NSGA-II for

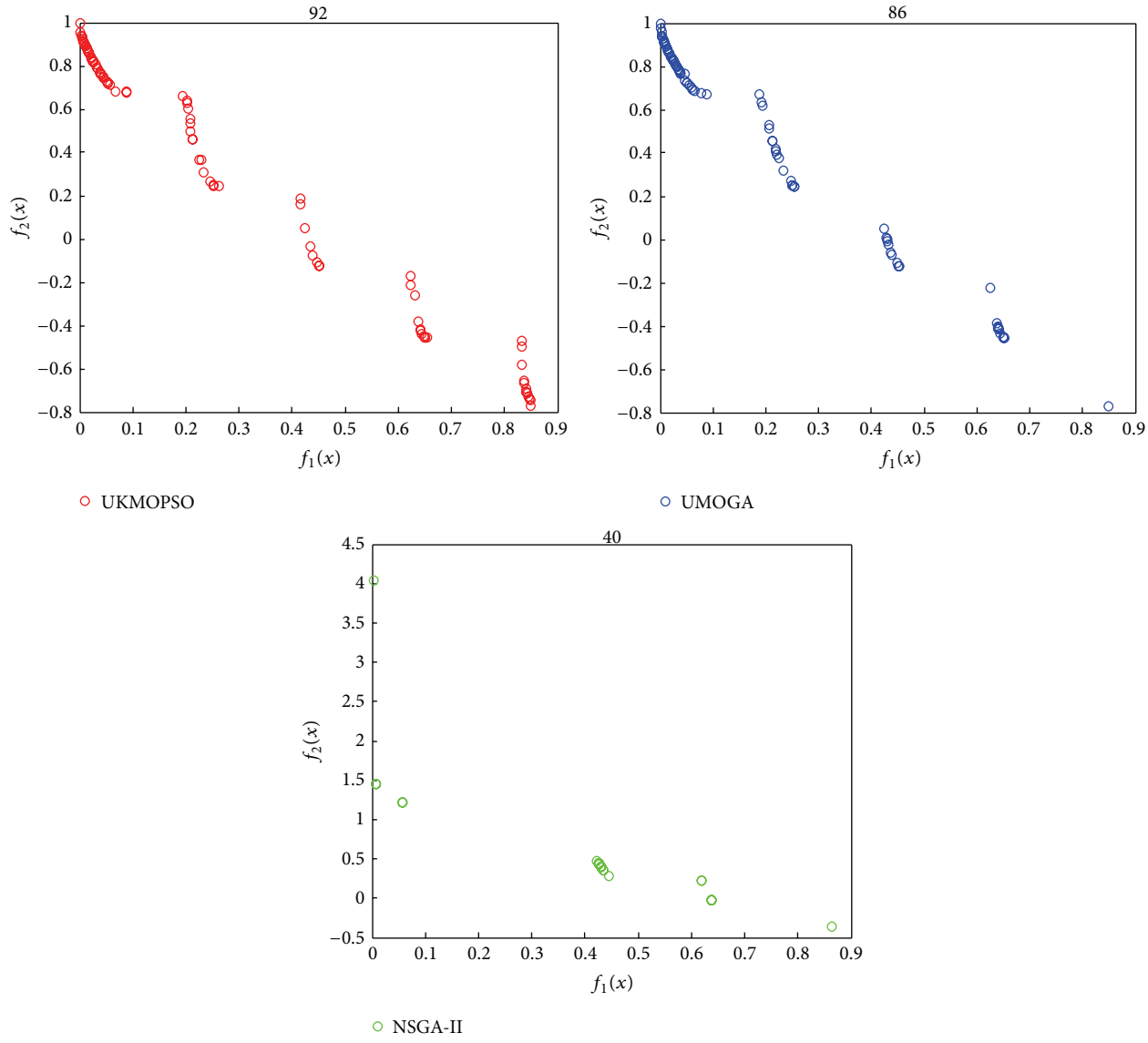


FIGURE 5: Pareto fronts obtained by different algorithms for test problem ZDT3.

all test functions. It means that the PF found by UMOGA is closer to the true PF than the PF obtained by NSGA-II.

From Table 5, we can see that all the MS values of UKMOPSO are almost close to 1 for each test function. It means that almost all the true PF is totally covered by the PF obtained by UKMOPSO. UMOGA is similar to UKMOPSO. The MS values of NSGA-II are much lesser than UKMOPSO and UMOGA, especially MS = 0.0019 for ZDT2.

Figures 1–6 all demonstrate that the PF found by UKMOPSO is the nearest to the true PF and scatters most uniformly in those obtained by three algorithms. Most of the points in the PF found by UKMOPSO overlap the points in the true PF. It means the solution quality obtained by UKMOPSO is very high.

4.5. Influence of the Uniform Crossover and PAM. In order to find out the influence of the uniform crossover on the proposed algorithm, we compare the distribution and number of Pareto optimal solutions before and after performing the uniform crossover. One of the simulating results on the test function ZDT1 is shown in Figure 7.

From Figure 7, it can be seen that, before and after performing the uniform crossover, the number of data points in the 1st cluster and the 2nd cluster varies from 7 to 15 and from 19 to 26, respectively; namely, many new Pareto optimal solutions having not been found before performing the uniform crossover are generated, and the differences of the data points between two clusters have been decreased. This will directly influence the uniformity of the PF and acquire more uniform Pareto solutions.

PAM is used to determine which Pareto solutions are to be removed from or be inserted into the external archive. This is to maintain the diversity of Pareto optimal solutions. The

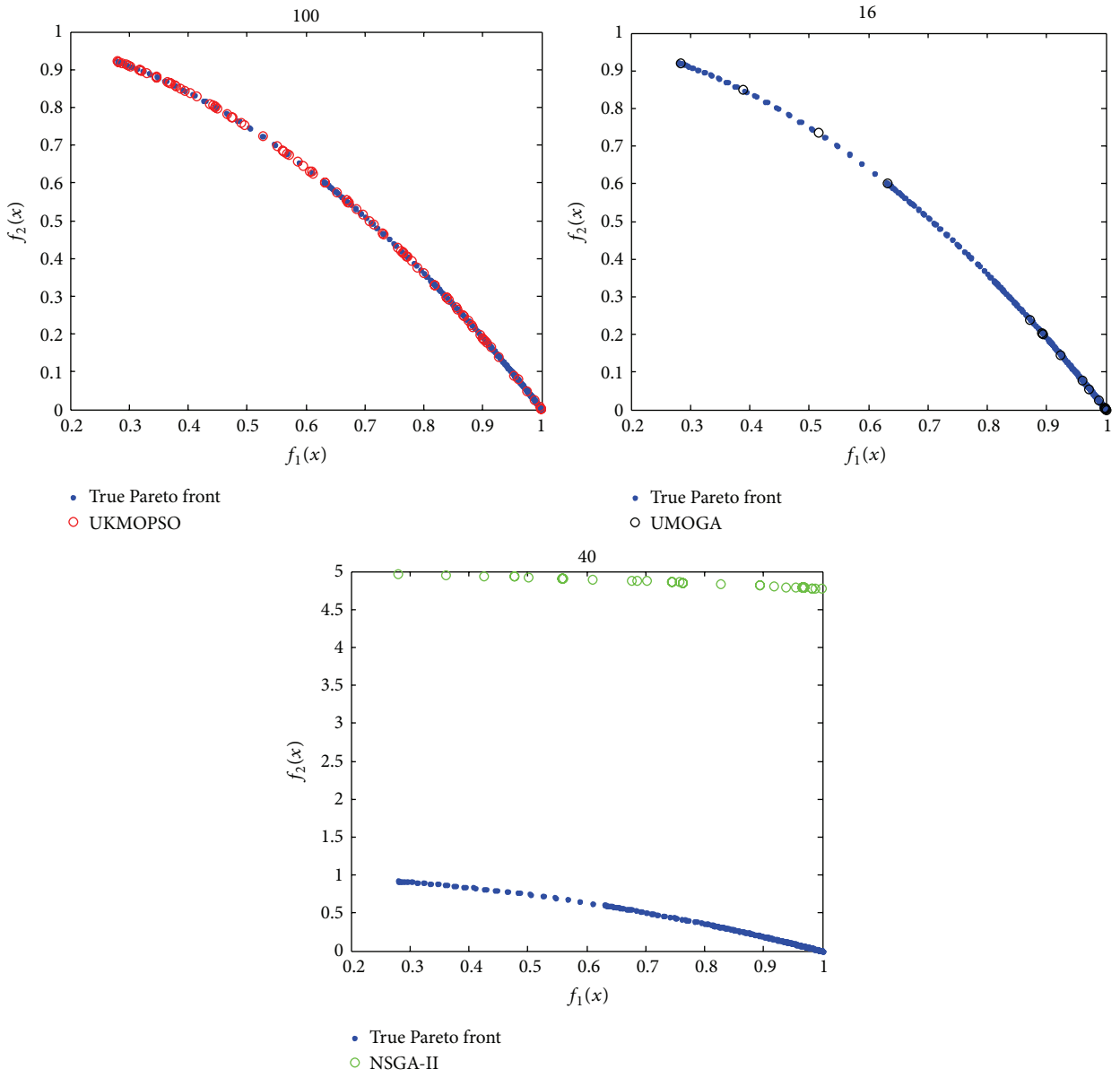


Figure 6: Pareto fronts obtained by different algorithms for test problem ZDT6.

diversity can be computed by the "distance-to-average-point" measure [44] defined as

$$\text{diversity}(S) = \frac{1}{|S|}\sum_{i=1}^{|S|}\sqrt{\sum_{j=1}^{N}\left(p_{ij} - \overline{p}_j\right)^2}, \tag{20}$$

where S is the population, $|S|$ is the swarm size, N is the dimensionality of the problem, p_{ij} is the j'th value of the i'th particle, and $\overline{p}_j$ is the j'th value of the average point $\overline{p}$.

If the number of Pareto optimality is less than or equal to the size of the external archive, PAM has no influence on the proposed algorithm. Otherwise, it is used to select the different type of Pareto optimality from several clusters. Therefore, The diversity of Pareto optimality will certainly increase. We monitor the diversity of Pareto optimality before and after performing PAM on ZDT1 at a certain time. The values are, respectively, 87.62 and 117.52. This fully demonstrates that PAM can improve the diversity of Pareto optimality.

5. Conclusion and Future Work

In this paper, a multiobjective particle swarm optimization based on PAM and uniform design is presented. It firstly implements PAM to partition the Pareto front into K clusters in the objective space; and then it implements the uniform crossover operator on the minimal cluster so as to generate more Pareto optimality in the minimal cluster. When the size of the Pareto solution is larger than that of the external archive, PAM is used to determine which Pareto solutions are to be removed from or be inserted into the external archive.

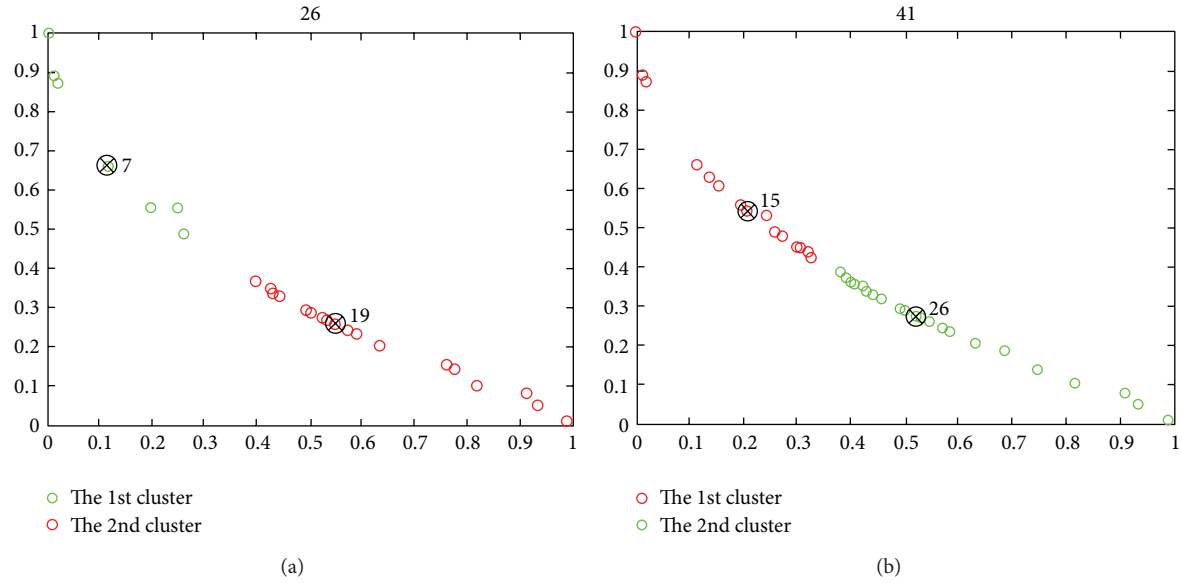


Figure 7: Pareto optimal solutions before (a) and after (b) performing the uniform crossover.

Finally, it keeps all the points in the lesser clusters and discards some points in the larger clusters. This can ensure that each of the clusters will contain approximately the same data points. Therefore, the diversity of the Pareto solutions will increase, and they can scatter uniformly over the Pareto front. The results of the experimental simulation performed on several well-known test problems indicate that the proposed algorithm obviously outperforms the other two algorithms.

This algorithm is going on for further enhancement and improvement. One attempt is to use a more efficient or approximate clustering algorithm to speed up the execution time of this algorithm. Another attempt is to extend its application scopes.

Conflict of Interests

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

Acknowledgments

This research was supported by Guangxi Natural Science Foundation (no. 2013GXNSFAA019337), Guangxi Universities Key Project of Science and Technology Research (no. KY2015ZD099), Key project of Guangxi Education Department (no. 2013ZD055), Scientific Research Staring Foundation for the PHD Scholars of Yulin Normal University (no. G2014005), Special Project of Yulin Normal University (no. 2012YJZX04), and Key Project of Yulin Normal University (no. 2014YJZD05). The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References

[1] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the 1st International Conference on Genetic Algorithms*, pp. 93–100, 1985.

[2] L. Tang and X. Wang, "A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 1, pp. 20–45, 2013.

[3] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Parallel Problem Solving from Nature PPSN VI*, vol. 1917 of *Lecture Notes in Computer Science*, pp. 849–858, Springer, Berlin, Germany, 2000.

[4] J. Zhang, Y. Wang, and J. Feng, "Attribute index and uniform design based multiobjective association rule mining with evolutionary algorithm," *The Scientific World Journal*, vol. 2013, Article ID 259347, 16 pages, 2013.

[5] S. Jiang and Z. Cai, "Faster convergence and higher hypervolume for multi-objective evolutionary algorithms by orthogonal and uniform design," in *Advances in Computation and Intelligence*, vol. 6382 of *Lecture Notes in Computer Science*, pp. 312–328, Springer, Berlin, Germany, 2010.

[6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.

[7] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, IEEE, Nagoya, Japan, October 1995.

[8] L. Yang, "A fast and elitist multi-objective particle swarm algorithm: NSPSO," in *Proceedings of the IEEE International Conference on Granular Computing (GRC '08)*, pp. 470–475, August 2008.

[9] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE*

Transactions on Evolutionary Computation, vol. 8, no. 3, pp. 256–279, 2004.

[10] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[11] H. Xiaohui and R. Eberhart, "Multiobjective optimization using dynamic neighborhood particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation (CEC '02)*, vol. 2, pp. 1677–1681, May 2002.

[12] 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 (SIS '03)*, pp. 26–33, Indianapolis, Ind, USA, April 2003.

[13] T. Bartz-Beielstein, P. Limbourg, J. Mehnen, K. Schmitt, K. E. Parsopoulos, and M. N. Vrahatis, "Particle swarm optimizers for Pareto optimization with enhanced archiving techniques," in *Proceedings of the Congress on Evolutionary Computation (CEC '03)*, vol. 3, pp. 1780–1787, December 2003.

[14] K. E. Parsopoulos, D. K. Tasoulis, and M. N. Vrahatis, "Multiobjective optimization using parallel vector evaluated particle swarm optimization," in *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA '04)*, vol. 2, pp. 823–828, February 2004.

[15] S.-J. Tsai, T.-Y. Sun, C.-C. Liu, S.-T. Hsieh, W.-C. Wu, and S.-Y. Chiu, "An improved multi-objective particle swarm optimizer for multi-objective problems," *Expert Systems with Applications*, vol. 37, no. 8, pp. 5872–5886, 2010.

[16] A. A. Mousa, M. A. El-Shorbagy, and W. F. Abd-El-Wahed, "Local search based hybrid particle swarm optimization algorithm for multiobjective optimization," *Swarm and Evolutionary Computation*, vol. 3, pp. 1–14, 2012.

[17] J. Wang, W. Liu, W. Zhang, and B. Yang, "Multi-objective particle swarm optimization based on self-update and grid strategy," in *Proceedings of the 2012 International Conference on Information Technology and Software Engineering*, vol. 211 of *Lecture Notes in Electrical Engineering*, pp. 869–876, Springer, Berlin, Germany, 2013.

[18] K. Khalili-Damghani, A.-R. Abtahi, and M. Tavana, "A new multi-objective particle swarm optimization method for solving reliability redundancy allocation problems," *Reliability Engineering & System Safety*, vol. 111, pp. 58–75, 2013.

[19] J. Zhang, Y. Wang, and J. Feng, "Parallel particle swarm optimization based on PAM," in *Proceedings of the 2nd International Conference on Information Engineering and Computer Science (ICIECS '10)*, Wuhan, China, December 2010.

[20] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2006.

[21] L. F. Ibrahim, "Using of clustering algorithm CWSP-PAM for rural network planning," in *Proceedings of the 3rd International Conference on Information Technology and Applications (ICITA '05)*, vol. 1, pp. 280–283, IEEE, July 2005.

[22] P. Chopra, J. Kang, and J. Lee, "Using gene pair combinations to improve the accuracy of the PAM classifier," in *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM '09)*, pp. 174–177, November 2009.

[23] Y.-W. Leung and Y. Wang, "Multiobjective programming using uniform design and genetic algorithm," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 30, no. 3, pp. 293–304, 2000.

[24] D. Liu, K. C. Tan, C. K. Goh, and W. K. Ho, "A multiobjective memetic algorithm based on particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics*, vol. 37, no. 1, pp. 42–50, 2007.

[25] 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.

[26] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms. II. Application example," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 28, no. 1, pp. 38–47, 1998.

[27] F. Kursawe, "A variant of evolution strategies for vector optimization," in *Parallel Problem Solving from Nature*, vol. 496 of *Lecture Notes in Computer Science*, pp. 193–197, Springer, Berlin, Germany, 1991.

[28] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[29] H.-L. Liu, Y. Wang, and Y.-M. Cheung, "A Multi-objective evolutionary algorithm using min-max strategy and sphere coordinate transformation," *Intelligent Automation & Soft Computing*, vol. 15, no. 3, pp. 361–384, 2009.

[30] N. Chakraborti, B. S. Kumar, V. S. Babu, S. Moitra, and A. Mukhopadhyay, "A new multi-objective genetic algorithm applied to hot-rolling process," *Applied Mathematical Modelling*, vol. 32, no. 9, pp. 1781–1789, 2008.

[31] K. Deb, "Multi-objective genetic algorithms: problem difficulties and construction of test problems," *Evolutionary Computation*, vol. 7, no. 3, pp. 205–230, 1999.

[32] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization*, Advanced Information and Knowledge Processing, pp. 105–145, Springer, London, UK, 2005.

[33] Y. Shi and R. Eberhart, "Modified particle swarm optimizer," in *Proceedings of the IEEE World Congress on Computational Intelligence*, pp. 69–73, May 1998.

[34] C. S. Feng, S. Cong, and X. Y. Feng, "A new adaptive inertia weight strategy in particle swarm optimization," in *Procedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 4186–4190, IEEE, Singapore, September 2007.

[35] C. K. Goh, K. C. Tan, D. S. Liu, and S. C. Chiam, "A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design," *European Journal of Operational Research*, vol. 202, no. 1, pp. 42–54, 2010.

[36] P. Cheng, J.-S. Pan, L. Li, Y. Tang, and C. Huang, "A survey of performance assessment for multiobjective optimizers," in *Proceedings of the 4th International Conference on Genetic and Evolutionary Computing (ICGEC '10)*, pp. 341–345, December 2010.

[37] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[38] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[39] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II," *IEEE*

Transactions on Evolutionary Computation, vol. 13, no. 2, pp. 284–302, 2009.

[40] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," Working Report, Department of Computing and Electronic Systems, University of Essex, 2008.

[41] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.

[42] C.-S. Tsou, H.-H. Fang, H.-H. Chang, and C.-H. Kao, "An improved particle swarm Pareto optimizer with local search and clustering," in *Simulated Evolution and Learning*, vol. 4247 of *Lecture Notes in Computer Science*, pp. 400–407, 2006.

[43] U. Wickramasinghe and X. Li, "Choosing leaders for multiobjective PSO algorithms using differential evolution," in *Proceeding of the 7th International Conference Simulated Evolution and Learning*, vol. 5361 of *Lecture Notes in Computer Science*, pp. 249–258, Springer, Berlin, Germany, 2008.

[44] T. Krink, J. S. Vesterstrom, and J. Riget, "Particle swarm optimisation with spatial particle extension," in *Proceedings of the Congress on Evolutionary Computation (CEC '02)*, pp. 1474–1479, May 2002.