Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 5831362, 17 pages
https://doi.org/10.1155/2019/5831362

Research Article

Modified Antipredatory Particle Swarm Optimization for Dynamic Economic Dispatch with Wind Power

Kai Chen, Li Han, Shuhuan Wang, Junjie Lu, and Liping Shi

School of Electrical and Power Engineering, China University of Mining and Technology, Xuzhou 221008, China

Correspondence should be addressed to Li Han; dannyli717@163.com

Received 22 May 2019; Revised 11 August 2019; Accepted 29 August 2019; Published 23 October 2019

Academic Editor: Gaetano Zizzo

A modified antipredatory particle swarm optimization (MAPSO) algorithm with evasive adjustment behavior is proposed to solve the dynamic economic dispatch problem of wind power. The algorithm adds the social avoidance inertia weight to the conventional antipredatory particle swarm optimization (APSO) speed update formula. The size of inertia weight is determined by the distance between the global worst particle and other particles. After normalizing the distance, the inertia weight is controlled within the ideal range by using the characteristics of sigmoid function and linear decreasing method, which improves the ability of particles to avoid the worst solution. Then, according to the characteristics of the acceleration coefficient which can adjust the local and global searching ability of particles, acceleration coefficients of nonlinear change strategy is proposed to improve the searching ability of the algorithm. Finally, the proposed algorithm is applied to several benchmark functions and power grid system models, and the results are compared with those reported using other algorithms, which prove the effectiveness and superiority of the proposed algorithm.

1. Introduction

As environmental issues become more prominent, wind energy, as an important renewable energy source, has been paid more and more attention. However, the volatility and randomness of wind power make the dynamic economic dispatching (DED) model more and more complicated, which also brings many difficulties to the solution of the model. Therefore, the method for solving the dynamic economic dispatching model has always been a research hotspot [1, 2].

In recent years, artificial intelligence algorithms have been widely used in dynamic economic dispatching models. For example, genetic algorithm (GA) [3–9], simulated annealing algorithm (SA) [10], tabu search algorithm (TS) [11, 12], differential evolution algorithm (DE) [13–19], particle swarm optimization (PSO) [20–34], artificial bee colony algorithm (ABC) [35], artificial immune system algorithm (AIS) [36, 37], evolutionary programming algorithm (EP) [38–40], complementary quadratic programming algorithm (cQP) [41], biogeography-based optimization algorithm (BBO) [42, 43], teaching learning-based optimization algorithm (TLBO) [44, 45], charged system search algorithm (CSSA) [46], flower pollination algorithm (FPA) [47], rooted tree optimization (RTO) [48], synergic predator-prey optimization (SPPO) [49], kinetic gas molecule optimization (KGMO) [50], grey wolf optimization (GWO) [51], modified crow search algorithm (MCSA) [52], Maclaurin series-based Lagrangian algorithm (MSL) [53], pattern search algorithm (PS) [54], harmony search algorithm (HS) [55], covariance matrix adapted evolution strategy algorithm (CMAES) [56], improved bacterial foraging algorithm (IBFA) [57], and mine blast algorithm (MBA) [58].

In the intelligent optimization algorithm, particle swarm optimization is favored by scholars because of its advantages of low parameter setting, fast convergence, robustness, and easy implementation. Some scholars have studied the acceleration coefficients c_{1g} and c_{2g} of particle swarm optimization. For example, Snganthan [59] indicated that a better solution can be obtained when the two acceleration coefficients are constant. Trelea [60] further used the convergence factor to improve the convergence ability of the

basic PSO and proves the correlation between the two acceleration coefficients. On this basis, Venter and Sobieszczanski-Sobieski [61] proved through experiments that a small cognitive coefficient and a large social coefficient can improve the performance of the PSO algorithm. However, the acceleration coefficients of the abovementioned literatures for particle swarm optimization are all the fixed value, and the influence of the acceleration coefficient of the variation strategy on the performance of the algorithm is not considered. In response to this, Ratnaweera et al. [62] proposed time-varying acceleration coefficients (TVACs), which is to increase the global search ability of the algorithm by using larger individual cognitive coefficients and smaller social coefficients in the early iteration (c_{1g} is larger, and c_{2g} is smaller), and as the number of iterations increases, the individual cognitive coefficient decreases linearly and the social coefficient increases linearly so that the algorithm can obtain smaller individual cognitive coefficients and larger social coefficients to speed up the convergence rate of PSO (c_{1g} is smaller, and c_{2g} is larger). This change process allows the particles to be distributed in every corner of the search space at an early stage and can quickly converge to the global best at a later stage. However, the linear change strategy makes the value of the acceleration coefficient too uniform in the whole change interval, and the speed of change does not change according to the number of iterations, which cause the particles to fall into prematureness in advance and difficult to jump out in the later stage. Therefore, this linear change strategy still cannot effectively raise the performance of the algorithm. Another part of the literature studies the movement characteristics, displacement, and velocity update formula of particles. Kheshti et al. [63] proposed a double-inertia weighted particle swarm optimization algorithm which uses two different inertia weights at the beginning and the end of the algorithm. Both of inertia weights enhance the local and global search ability of the particles and improve the accuracy of the algorithm. Zou et al. [64] used the new displacement update formula to guide the particle's search activity, expanding the particle's search space and reducing the possibility of particles falling into local optimum. Elsayed et al. [65] added a crossover operation followed by a greedy selection process while replacing the mean best position of the particles with the personal best position of each particle in the velocity updating equation of random drift particle swarm, which increases the diversity of population and improves the performance of the algorithm. Yao et al. [66] introduced quantum computing theory and used quantum bit and angle to depict the state of particles rather than using particle position and velocity of the classical PSO, which shows a stronger search ability and quicker convergence speed. However, these studies all make improvements from the aspects of the optimal solution of particles and do not consider the influence of the worst solution on particle optimization. Thus, Selvakumar Thanushkodi [67] added the worst solution idea to the particle swarm optimization and proposed the anti-predatory particle swarm optimization. The algorithm likens the local worst solution and the global worst solution found by the particles in the iteration to the natural predator. The particles will dodge the predator and avoid the worst solution during the optimization process. However, in each iteration, the particles cannot judge whether to make evasive behavior or the degree of evasion according to the distance between the particles and the worst solution at each iteration; that is to say, no matter how far the particles is from the worst solution, it will make the same degree of evasive behavior, which inevitably increases the complexity of the algorithm and the limitation of search space.

Synthesizing above all researches, this paper proposes a modified antipredatory particle swarm optimization (MAPSO) based on the evasive adjustment behavior of the particles. Firstly, a nonlinear change strategy is introduced to solve the problem that the linear change strategy of acceleration coefficient makes the particles prone to precocity. Then, before the global worst position part of the antipredation particle swarm velocity formula, the avoiding inertia weight is added to adjust the degree of dodge of the particles. The equation of avoiding inertia weight considers the monotonicity of the sigmoid function and introduces the linear decreasing coefficients to control the avoiding inertia weight within the ideal range of variation. Finally, the effectiveness of the proposed algorithm is verified by test functions and standard power system cases.

2. Mathematical Model for Economic Dispatch considering Wind Power

2.1. Cost Function. As a renewable and clean energy source, wind power does not generate fuel costs. Therefore, on the premise of not considering wind power fuel cost, the goal of dynamic economic dispatching of power system with wind farms is to reasonably distribute the output of each generator set in the power grid to minimize the total cost of power generation during dispatching under the condition of meeting the constraints of load and unit constraints. Usually, the objective of economic dispatch of power systems with wind power is usually to minimize the sum of fuel cost of thermal power units, valve-point effect cost, overestimation cost, and underestimation cost of wind power uncertainty. The total cost expression is as follows:

$$\min f(P, w) = \sum_{n=1}^{N} F(P_n) + \sum_{m=1}^{M} R_m(w_m), \tag{1}$$

where $f(P, w)$ is the total cost of dispatching in single period, F is the fuel cost, R is the cost of wind power uncertainty, n is the number of thermal power units, N is the total number of thermal power units, m is the number of wind farms, M is the total number of wind farms, P_n is the active power output of the nth thermal power unit in a single period, and w_m is the actual output of the mth wind farm.

The objective function of dynamic economic dispatch needs to consider the sum of costs of each period. The expression is as follows:

$$\min C = \sum_{h=1}^{H} f_h(P, w), \tag{2}$$

where h is the dispatching periods, H is the total number of dispatching periods, and C is the total cost of dynamic economic dispatch in H period.

2.1.1. Fuel Cost. The total fuel cost includes fuel cost of thermal power units and energy consumption cost of steam turbine valve-point effect, and the expression is set as

$$F(P_n) = a_n P_n^2 + b_n P_n + c_n + \left| d_n \sin\left[e_n \left(P_{n,\min} - P_n \right) \right] \right|, \tag{3}$$

where a_n, b_n, and c_n are the fuel cost coefficient, d_n and e_n are the valve-point effect cost coefficients, and $P_{n,\min}$ is the lower limit of the active power output of the nth thermal power unit.

2.1.2. Wind Power Cost. The expressions of overestimation and underestimation cost caused by the randomness of wind power are as follows:

$$R_m(w) = C_{\mathrm{ow},m}\left(w_m - w_{\mathrm{av},m} \right) + C_{\mathrm{uw},m}\left(w_{\mathrm{av},m} - w_m \right), \tag{4}$$

where $C_{\mathrm{ow},m}$ and $C_{\mathrm{uw},m}$ are the penalty coefficients for overestimation and underestimation of wind power cost, respectively, and $w_{\mathrm{av},m}$ is the available output of the mth wind farm.

2.2. Constraints

2.2.1. System Active Power Balance Constraints. The power balance constraint equation considering the network loss is expressed as

$$\sum_{n=1}^{N} P_n = P_{\mathrm{L}} + P_{\mathrm{D}}. \tag{5}$$

The power balance constraint equation considering the network loss and wind power is expressed as

$$\sum_{n=1}^{N} P_n + \sum_{m=1}^{M} w_m = P_{\mathrm{L}} + P_{\mathrm{D}}, \tag{6}$$

where P_{D} is the predicted value of active power in single load period and P_{L} is the network loss in single load period. The network loss calculation formula is

$$P_{\mathrm{L}} = \sum_{i}^{N} \sum_{j}^{N} P_i B_{i,j} P_j + \sum_{i}^{N} P_i B_{i,0} + B_{0,0}, \tag{7}$$

where $B_{i,j}$, $B_{i,0}$, and $B_{0,0}$ are network loss parameters.

2.2.2. Unit Active Output Constraints

$$P_{n,\min} < P_n < P_{n,\max}, \tag{8}$$

$$0 < w_m < w_{m,\max}, \tag{9}$$

where $P_{n,\max}$ and $P_{n,\min}$ are, respectively, the upper and lower limits of the output of the nth unit and $w_{m,\max}$ is the maximum installed capacity of the mth wind farm.

2.2.3. Unit Ramp Rate Constraints

$$\begin{cases} P_{n,h} - P_{n,h-1} < U_{\mathrm{Rn}} \Delta T, \\ P_{n,h-1} - P_{n,h} < D_{\mathrm{Rn}} \Delta T, \end{cases} \tag{10}$$

where U_{Rn} and D_{Rn} are the up and down ramp rates of the nth unit, respectively, and ΔT is the dispatch time interval and takes 1 h.

2.2.4. System Spinning Reserve Constraints

(1) Positive spinning reserve capacity constraint with wind power system:

$$\begin{cases} w_{\mathrm{av}} \times w_{\mathrm{u}}\% + P_{\mathrm{D}} \times L_{\mathrm{u}}\% < \sum_{n=1}^{N} U_n, \\ U_n = \min\left(P_{n,\max} - P_n, U_{\mathrm{Rn}} T_{10} \right). \end{cases} \tag{11}$$

(2) Negative spinning reserve capacity constraint with wind power system:

$$\begin{cases} \left(w_{\max} - w_{\mathrm{av}} \right) \times w_{\mathrm{d}}\% + P_{\mathrm{D}} \times L_{\mathrm{d}}\% < \sum_{n=1}^{N} D_n, \\ D_n = \min\left(P_n - P_{n,\min}, D_{\mathrm{Rn}} T_{10} \right), \end{cases} \tag{12}$$

where $w_{\mathrm{u}}\%$ and $w_{\mathrm{d}}\%$ are, respectively, the demand coefficients of wind power prediction errors for positive and negative spinning reserve; $L_{\mathrm{u}}\%$ and $L_{\mathrm{d}}\%$ are, respectively, the demand coefficients of the system load prediction errors for positive and negative spinning reserve; U_n and D_n are, respectively, the positive and negative spinning reserve capacity provided by the nth thermal power unit; and T_{10} is the response time of spinning reserve, taking 10 minutes.

3. Modified Antipredation Particle Swarm Optimization Algorithm

3.1. Antipredation Particle Swarm Optimization (APSO). Antipredation particle swarm optimization (APSO) algorithm regards the particle optimization process as the foraging process of birds. Birds will try their best to avoid the attack of natural enemies while searching for food in nature. APSO adds the position of the natural enemies as the worst position to the PSO model and divides the cognitive experience in the PSO algorithm into two parts: good cognitive experience and bad cognitive experience. The former is the best location memory of the particle's own experience, and the latter is the memory of the worst position (predator's position) experienced by the particle itself. Similarly, social

experience is divided into two parts: good social experience and bad social experience. The former is the best position memory experienced by the group, and the latter is the worst position memory experienced by the group. Thus, the speed and displacement update formula of APSO algorithm is obtained as follows:

$$V_i^{t+1} = \omega V_i^t + c_{1g} r_1 \left(pbest_i^t - x_i^t\right) + c_{1b} r_2 \left(x_i^t - pworst_i^t\right) + c_{2g} r_3 \left(gbest^t - x_i^t\right) + c_{2b} r_4 \left(x_i^t - gworst^t\right), \tag{13}$$

$$x_i^{t+1} = x_i^t + V_i^{t+1}, \tag{14}$$

where V_i^t is the velocity of the ith particle at the tth iteration; ω is the inertia weight, indicating the degree of succession of the particle to the tth iteration speed; c_{1g} is the acceleration coefficient of the particle flying to its own best position; c_{1b} is the acceleration coefficient of the particle flying away its own worst position; c_{2g} is the acceleration coefficient of the best position of the particle flying to the group; c_{2b} is the acceleration coefficient of the worst position of the particle flying away from the group; r_1, r_2, r_3, and r_4 are random numbers uniformly distributed on [0, 1]; $pbest_i^t$ is the individual optimal value of the ith particle at the tth iteration; $pworst_i^t$ is the individual worst value of the ith particle at the tth iteration; $gbest^t$ is the global optimal value of the particle swarm at the tth iteration; $gworst^t$ is the global worst value of the particle swarm at the tth iteration; and x_i^t is the displacement of the ith particle at the tth iteration.

In equation (13), the velocity update formula is composed of five parts: the first part reflects the influence of the current velocity on the particle, which is related to the current state of the particle; the second part reflects that the particle is affected by its own best position, namely, the particle's good cognitive experience; the third part reflects the influence of its own worst position, i.e., the bad cognitive experience of particles; the fourth part reflects that particles are affected by the best position of the group, that is, the good social experience of the particle; and the fifth part means the influence of particles on the worst position of the group, i.e., the bad social experience of particles.

3.2. Modified Antipredation Particle Swarm Optimization (MAPSO)

3.2.1. Nonlinear Time-Varying Acceleration Coefficient. In order to improve the performance of the algorithm, Ratnaweeera et al. proposed a linear strategy to adjust the acceleration coefficients c_{1g} and c_{2g} and experimentally obtained that c_{1g} is linearly decreasing from 2.5 to 0.5 and c_{2g} is linearly increasing from 0.5 to 2.5 (the black dotted part in Figure 1), and the algorithm has the best optimization effect. The reason for adopting of linear strategy is that, at the beginning of iteration, particles need to occupy every corner of the search space by relying on their own cognitive experience and maintain the diversity of particles, so a larger c_{1g} and a smaller c_{2g} are set. At the later stage of iteration, the global optimal value approaches the convergence value, and each particle needs to approach the optimal solution and search around it. At this time, c_{1g} is small, while c_{2g} is large. However, the linear time-varying acceleration coefficient changes too fast in the early iteration so that the particles have not yet undergone too many local searches and have to move closer to the global optimal solution on a large scale. In general, the global optimal value of the population in the early search is often the local optimal value, especially for the multipeak problems, which lead to the premature convergence of the particles to the local optimum; while the speed of change of the acceleration coefficient slows down in the late iteration, which makes the particles learn too much from themselves, and is rarely affected by other particles, further aggravating the premature algorithm. Based on this, this paper proposes a nonlinear time-varying acceleration coefficient method as a new adjustment strategy for c_{1g} and c_{2g} (the solid line part of Figure 1). The improved expression is as follows:

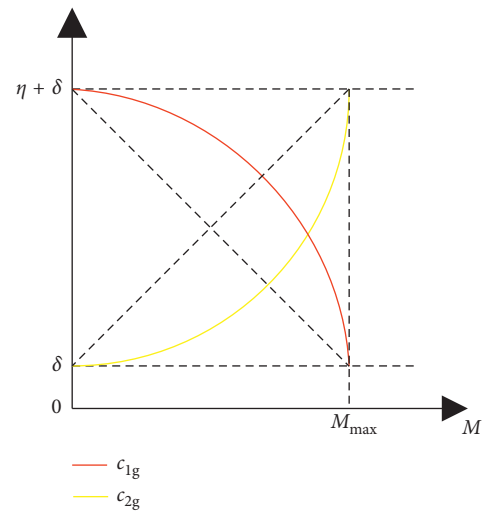

Figure 1: Change trend chart of c_{1g} and c_{2g}.

$$c_{1g} = \eta \cdot \sqrt{1 - \left(\frac{\theta}{\theta_{\max}}\right)^2} + \delta, \tag{15}$$

$$c_{2g} = (\eta + \delta) - \eta \cdot \sqrt{1 - \left(\frac{\theta}{\theta_{\max}}\right)^2}, \tag{16}$$

where the nonlinear parameters η and δ are used to limit the range of variation of c_{1g} and c_{2g}. Since the nonlinear strategy has the same trend as the linear strategy, only the rate of change is different, so this paper also uses the same range of variation as the linear variation strategy, and in order to satisfy it, the η and δ are supposed to be 2 and 0.5, respectively; θ is the current number of iterations; and $\theta_{\max}$ is the total number of iterations.

Figure 1 intuitively shows the change process of linear acceleration coefficient and nonlinear acceleration coefficient.

As can be seen from Figure 1, at the initial iteration stage of the algorithm, c_{1g} and c_{2g} change slowly so that the

particles can have more time to learn from themselves at the beginning of the iteration and are less affected by other particles. In this way, the particles can avoid blindly moving closer to other particles, which can effectively ameliorate the precocity phenomenon; in the later stage of iteration, c_{1g} and c_{2g} change faster, and the social experience of particles accumulates rapidly, which accelerate the convergence speed of particles to the global optimal solution. In summary, the setting of the nonlinear acceleration coefficient can improve the premature phenomenon of the particles in the early stage and speed up the convergence of the particles to the optimal solution in the later stage.

3.2.2. Adjustment of Avoid Inertia Weight ω_{gw}. c_{2b} in APSO is used to determine the extent to which particles avoid the global worst solution. It is defined as a constant in this algorithm, which may cause particles to avoid the global worst solution even when they are far away.

Figure 2(a) shows the optimization process of two particles i and j in APSO. The superscript t represents the optimization result of the tth iteration, and the superscript $t+1$ represents the optimization result of the $(t+1)$th iteration; black circle represents the global worst solution. It can be seen from the optimization result of the tth iteration that the particle i is closer to *gworst* and the particle j is farther away from *gworst*. In APSO, in the process of avoiding the global worst solution *gworst*, the particle does not adjust the avoidance degree according to the distance between itself and *gworst*, so the avoidance extent of i and j is the same. In fact, the particle j is far from *gworst* and does not need to make the same avoidance extent as particle i, or it may cause the problem that the particle j in Figure 2(a) oversteps the global optimal solution due to its excessive avoidance extent.

In order to solve this problem, the inertia weight ω_{gw} is defined based on the distance between the particle and the global worst solution *gworst* to adjust the evading degree of the particle. As shown in Figure 2(b), when the particle j is far away from *gworst*, the particle's evasive inertia weight ω_{gw} is small, and the degree of evasive behavior of the particle is small, avoiding the problem that the particle j is over the optimal solution in Figure 2(a); when the particle i is closer to *gworst*, the particle's evasive inertia weight ω_{gw} is large, and the degree of evasive behavior of the particle is large, thereby increasing the search range of the particles i.

The expression of ω_{gw} is shown in the following equation:

$$\omega_{gw} = \alpha - \beta \cdot \frac{1}{1 + e^{-\mu \cdot l_i^{\text{normal}}}}, \tag{17}$$

where

$$l_i^{\text{normal}} = 2 \cdot \frac{l_i - l_{\min}}{l_{\max} - l_{\min}} - 1, \tag{18}$$

$$l_i = \left| gworst^t - x_i^t \right|, \quad i = 1, 2, \ldots, I, \tag{19}$$

where α and β are linearly decreasing coefficients; μ is the inclination coefficient; l_i is the distance between the particle i and the global worst solution *gworst*; $l_{\min}$ and $l_{\max}$ are the minimum and maximum distances of the global worst solution to the particle swarm, respectively; l_i^{normal} is the normalized distance of l_i; and I is the number of particles.

When the particles are moving, ω_{gw} and l_i should exhibit an inverse relationship, i.e., the distance between the particles and *gworst* is too close (l_i is small), then ω_{gw} is larger, which makes the particles are completely away from *gworst*; on the contrary, ω_{gw} is smaller, making the particles slightly away from *gworst*. However, due to the monotonous increment of the sigmoid function itself, the inverse relationship cannot be met. Therefore, the linear decreasing strategy is added to the sigmoid function in equation (17) and the values of α and β are defined, which not only makes ω_{gw} show an inverse ratio with the change of the distance but also ω_{gw} is controlled within an ideal range related to α and β.

The fractional part in equation (17) is the sigmoid function, and the variation of ω_{gw} depends on l_i, which is the global worst value and the distance between the particles. However, if l_i^{normal} is directly expressed in l_i, then in the actual process, the difference in dimension and order of magnitude caused by different optimization models will cause l_i to be too large and the fractional part can only be taken to 1. In order to avoid this phenomenon, this paper normalizes l_i and controls it between [−1, 1], The normalized value is expressed by l_i^{normal}, as shown in equation (18).

Since the value of l_i^{normal} is between [−1, 1], the value range of sigmoid function is the red bold curve ($\overset{\frown}{\text{AB}}$) of Figure 3. However, this red part is only a small part of the whole value range of sigmoid function ($\overset{\frown}{\text{CABD}}$). As can be seen from equation (17), this phenomenon will lead to a very small range of values of ω_{gw}, so it is necessary to multiply a coefficient μ before l_i^{normal} to expand the range of the domain of the sigmoid function so that the range of the sigmoid function can be taken as wide as possible.

The red bold curve ($\overset{\frown}{\text{AB}}$) in Figure 3 is the range of the sigmoid function before the coefficient μ is defined, and the region ($\overset{\frown}{\text{CABD}}$) represented by $\mu \cdot l_i^{\text{normal}}$ in the figure almost covers the entire value range of the sigmoid function, so it is reasonable to define the coefficient μ.

Based on the analysis in this section, the velocity and displacement expressions of MAPSO are given by the following equations:

$$\begin{aligned} V_i^{t+1} = {} & \omega V_i^t + c_{1g} r_1 \left(pbest_i^t - x_i^t \right) + c_{1b} r_2 \left(x_i^t - pworst_i^t \right) \\ & + c_{2g} r_3 \left(gbest^t - x_i^t \right) + \omega_{gw} c_{2b} r_4 \left(x_i^t - gworst^t \right), \end{aligned} \tag{20}$$

$$x_i^{t+1} = x_i^t + V_i^{t+1}. \tag{21}$$

Equation (20) changes the trend of c_{1g} and c_{2g} into a nonlinear strategy based on the speed update formula of APSO and adds ω_{gw} before the global worst position, which not only improves the premature of the particles but also expands the search scope, allowing the algorithm to more effectively find the optimal solution to the problem.

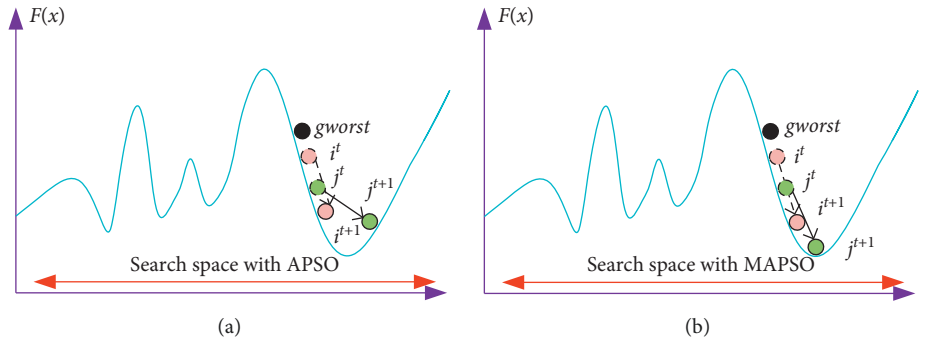

Figure 2: The evasive principle of (a) APSO and (b) MAPSO.

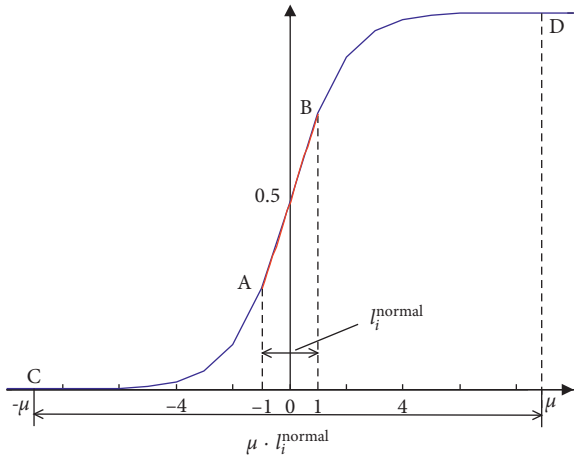

Figure 3: The value area of sigmoid function.

4. Constraint Processing and Algorithm Solving Steps

4.1. Constraints Processing. When dealing with DED model of power system, it is necessary to comprehensively consider the output distribution of units and the dynamic adjustment of units in different periods. Therefore, the upper and lower power bounds of thermal power units are dynamically adjusted before the dispatching calculation in each period to satisfy the unit ramp and power limitation constraints. Combined with equations (8) and (10), the following treatment is carried out:

$$\begin{cases} P_{n,h,\min} = \max\left[P_{n,h,\min}, \left(P_{n,h-1} - D_{\mathrm{Rn}}\right)\right], \\ P_{n,h,\max} = \min\left[P_{n,h,\max}, \left(P_{n,h-1} + U_{\mathrm{Rn}}\right)\right], \\ P_{n,h,\min} \le P_{n,h} \le P_{n,h,\max}. \end{cases} \tag{22}$$

4.2. Algorithm Steps and Flow Chart. According to the dynamic economic dispatch model of wind farms in this paper, the steps of MAPSO are as follows:

Step 1: initialize the particle population and randomly generate the initial velocity and displacement of each particle, input system, unit parameters, and algorithm control parameters. This paper uses linear decreasing inertia weight (LDIW) [68]:

$$\omega = \omega_{\max} + \left(\omega_{\min} - \omega_{\max}\right) \times \frac{\theta}{\theta_{\max}}, \tag{23}$$

where $\omega_{\max}$ is set to 0.9 and $\omega_{\min}$ is set to 0.4.

Step 2: limit the initial velocity and displacement of each particle according to the constraints processing of Section 4.1 so that the particle swarm can search in the feasible region.

Step 3: calculate the fitness value of the initial position of the particle and retain the individual optimal and worst positions *pbest* and *pworst*, global optimal, and worst positions *gbest* and *gworst*.

Step 4: calculate ω_{gw} according to equation (17) and update the displacement and velocity of each particle by equations (20) to (21).

Step 5: calculate the fitness value, update *pbest*, *pworst*, *gbest*, and *gworst*, and compare with the initial fitness value of step 3, and if the updated fitness value is dominant, it is retained.

Step 6: repeat steps 4 to 5 until the iteration terminates.

Step 7: output the optimal result.

Flow chart is shown in Figure 4.

5. Case Analysis

Benchmark functions and economic dispatch cases were used in this section to test and verify the MAPSO. The algorithm runs on the personal computer (Windows 7, 64 bits, Intel Core i5 2.6 GHz, 4G RAM) version of MATLAB 2014a.

5.1. Benchmark Functions. In order to verify the effectiveness of the MAPSO algorithm, five test functions are used to prove the search performance of the MAPSO.

(1) Sphere function:

$$f_1(x) = \sum_{i=1}^{n} x_i^2, \quad x \in [-100, 100]^n. \tag{24}$$

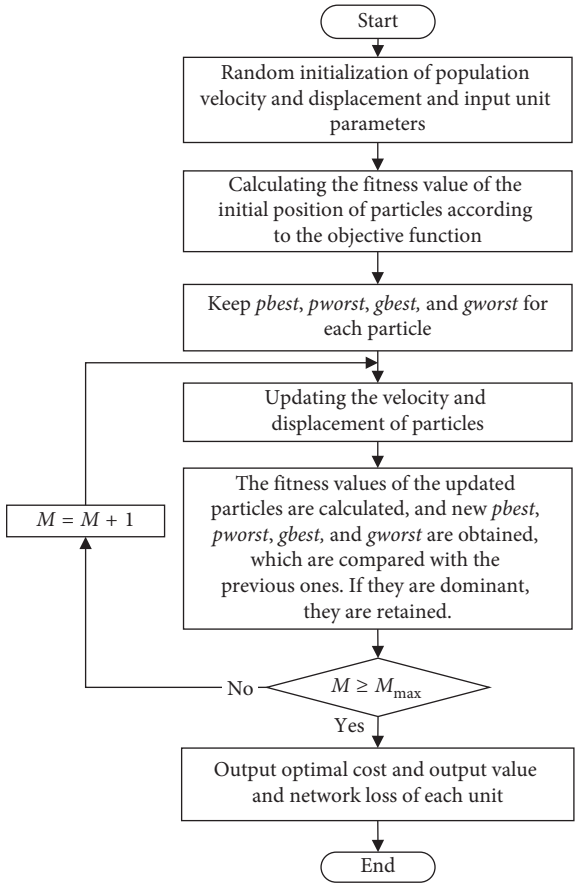

Figure 4: Flow chart of algorithm.

(2) Schwefel function:

$$f_2(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|, \quad x \in [-10, 10]^n. \tag{25}$$

(3) Rastrigin function:

$$f_3(x) = \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right), \quad x \in [-5.12, 5.12]^n. \tag{26}$$

(4) Griewank function:

$$f_4(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad x \in [-600, 600]^n. \tag{27}$$

(5) Styblinski function:

$$f_5(x) = \frac{\sum_{i=1}^{n} x_i^4 - 16x_i^2 + 5x_i}{2}, \quad x \in [-5, 5]^n. \tag{28}$$

Among these test functions, the minimum value of f_1~f_4 is 0 and the minimum value of f_5 is $-39.16599n$, where n is the dimension of the function (for example, when $n = 30$, the minimum value is about -1170). The algorithm parameters are set as follows: the total number of particles is 40, the maximum number of iterations is 1000, and the function dimension is 30. The time-varying acceleration coefficients particle swarm optimization (TVAC-PSO), the nonlinear time-varying acceleration coefficients particle swarm optimization (NTVAC-PSO), the time-varying acceleration coefficients antipredator particle swarm optimization (TVAC-APSO), the time-varying acceleration coefficients antipredator particle swarm optimization (NTVAC-APSO), and the proposed MAPSO are used to calculate the benchmark functions 50 times independently and compared the optimal value, average value, and standard deviation of the results, as shown in Table 1.

It can be seen from Table 1 that, in terms of the average value and the optimal solution, the results of the MAPSO are superior to the other four algorithms, even multimodal functions such as f_3 and f_4, which have multiple local minimum values, perform equally well. In terms of standard deviation, the standard deviation of f_1 and f_2 is 0, and in the function f_3–f_5, the results of MAPSO are also smaller than the other four algorithms. Taking f_5 as an example, the standard deviation of MAPSO is 34.7. Compared with TVAC-PSO, NTVAC-PSO, TVAC-APSO, and NTVAC-APSO, the standard deviation of MAPSO decreases by 6.8, 3.8, 0.3, and 2.5, respectively. This shows the stability of the MAPSO algorithm and proves the superiority of the MAPSO algorithm in the global optimal solution search ability.

In order to visually demonstrate the convergence characteristics of MAPSO and PSO, Figure 5 shows the convergence curves of two PSO variants and MAPSO.

As can be seen from Figure 5 that the iteration times of the MAPSO algorithm are the least when it converges to the optimal solution, which means that the proposed algorithm can not only improve the convergence speed but also increase the accuracy. In addition, during the iteration process, the fluctuation of the MAPSO convergence curve is relatively more frequent than other PSO variants, which shows that the MAPSO algorithm can expand the search space of particles and improve the activity of the particles so that the particles are not easy to fall into local optimum.

5.2. Economic Dispatch. In order to further prove the superiority of the MAPSO algorithm, the performance of the algorithm is validated by analyzing and simulating the standard 5 units and standard 10 units of IEEE 39-bus in this section. The parameters of the algorithm are set as follows: the number of particles is 30; the total number of iterations is 100; c_{1g} and c_{2g} are set according to formula (15)~(16); c_{1b} and c_{2b} are set to 0.4 and 0.2, respectively; ω is updated according to formula (23); α and β are taken to 0.4 and 0.2, respectively; and μ is taken to 9.

5.2.1. MAPSO Test of 5 Generators Unit System without Wind Power. Taking the five-unit power system without wind power as an example, the dispatching model considers fuel cost (3) and constraints (5), (7), (8), and (10), dispatching

Table 1: Comparison of five algorithms for benchmark functions.

Number	Function	Method	Average value	Optimal value	Standard deviation
1	Sphere	TVAC-PSO	$3.24E-01$	$1.25E-01$	$1.38E-01$
		NTVAC-PSO	$1.38E-01$	$4.79E-02$	$7.37E-02$
		TVAC-APSO	$0.00E+00$	$0.00E+00$	$0.00E+00$
		NTVAC-APSO	$0.00E+00$	$0.00E+00$	$0.00E+00$
		MAPSO	**0.00E + 00**	**0.00E + 00**	$0.00E+00$
2	Schwefel	TVAC-PSO	$4.32E+00$	$2.04E+00$	$1.11E+00$
		NTVAC-PSO	$3.59E+00$	$2.02E+00$	$1.07E+00$
		TVAC-APSO	$0.00E+00$	$0.00E+00$	$0.00E+00$
		NTVAC-APSO	$0.00E+00$	$0.00E+00$	$0.00E+00$
		MAPSO	**0.00E + 00**	**0.00E + 00**	$0.00E+00$
3	Rastrigin	TVAC-PSO	$5.57E+01$	$2.33E+01$	$1.39E+01$
		NTVAC-PSO	$4.98E+01$	$3.24E+01$	$1.12E+01$
		TVAC-APSO	$4.18E+01$	$1.53E+01$	$1.13E+01$
		NTVAC-APSO	$4.10E+01$	$1.37E+01$	$1.08E+01$
		MAPSO	**3.70E + 01**	**1.06E + 01**	$1.01E+01$
4	Griewank	TVAC-PSO	$3.85E-02$	$1.76E-02$	$1.08E-02$
		NTVAC-PSO	$1.31E-02$	$7.70E-03$	$5.45E-03$
		TVAC-APSO	$1.23E-02$	$3.10E-03$	$1.42E-02$
		NTVAC-APSO	$8.63E-03$	$3.10E-03$	$6.46E-03$
		MAPSO	**8.34E − 03**	**2.40E − 03**	$3.49E-03$
5	Styblinski	TVAC-PSO	$-9.52E+02$	$-1.07E+03$	$4.15E+01$
		NTVAC-PSO	$-9.71E+02$	$-1.04E+03$	$3.85E+01$
		TVAC-APSO	$-1.00E+03$	$-1.08E+03$	$3.50E+01$
		NTVAC-APSO	$-1.00E+03$	$-1.08E+03$	$3.72E+01$
		MAPSO	**−1.02E + 03**	**−1.11E + 03**	$3.47E+01$

period $H = 24$ h, and time interval is 1 h. The parameters of units, loads, and network losses in the system are obtained from [10]. The dynamic economic dispatching model is optimized 30 times by using the MAPSO algorithm, and the optimal total cost is 43439 USD. The output, network loss, and cost distribution of each unit within 24 hours are shown in Table 2.

In order to highlight the performance of the MAPSO algorithm, Table 3 lists the optimal results of the proposed algorithm compared with those of other algorithms.

It can be clearly seen that the proposed algorithm is superior to other algorithms in terms of average, maximum, and minimum, which further illustrates the effectiveness of the algorithm for model solving.

5.2.2. MAPSO Test of 10 Generators Unit System without Wind Power. The dispatching model is the same as the ten-unit model. The parameters of units, loads, and network losses in the system are obtained from [69]. The dynamic economic dispatching model is run 30 times by using the MAPSO algorithm, and the optimal total cost is 2474831 USD. The output, network loss, and cost distribution of each unit within 24 hours are shown in Table 4.

The MAPSO algorithm is compared with the algorithms used in the same model and data in recent years [19, 37, 57, 69–71]. The specific results are shown in Table 5.

From Table 5, it can be seen that, compared with the IMOEA/D-CH algorithm, MAMODE algorithm, HMO-DE-PSO algorithm, and RCGA/NSGA-II algorithm, the cost of MAPSO is reduced by 5387 USD, 17638 USD, 9187 USD, and 4187 USD. Similarly, compared with traditional algorithms such as AIS, PSO, and EP, the cost savings are also considerable. In terms of computing time, MAPSO only takes 12.26 seconds, which is much lower than MAMODE and RCGA/NSGA-II. Compared with traditional algorithms, the computing time of MAPSO is also significantly reduced. Compared with IBFA, the computing time of both algorithms is close, but the cost is reduced by 6920 USD. Synthesizing the above data analysis., it can be seen that the proposed MAPSO algorithm is more effective and superior for solving high-dimensional and multiconstrained dynamic economic dispatching models. It can not only save time but also find more economical and reasonable dispatching schemes for decision makers in the process of dispatching without losing accuracy while guaranteeing speed.

5.2.3. MAPSO Test of 10 Generators Unit System with Wind Power. In this test system, a wind farm connected to the grid is considered. The total installed capacity of the wind farm connected to the grid is 100 MW, with a total of 50 wind turbines. The dispatching model considers fuel cost (3) and constraints (6)–(10), dispatching period $H = 24$ h, and time interval is 1 h. The parameters of units and network losses in the system are obtained from [70]. The predicted values of wind power and system load in each period are shown in Table 6 w_u% and w_d% are 0.2 and 0.3, respectively, while L_u% and L_d% are 0.05, respectively. The dynamic economic dispatching model is optimized 30 times by using the MAPSO algorithm, and the optimal total cost is 2355671 USD. The output, network loss, and cost distribution of each unit within 24 hours are shown in Table 7. The power

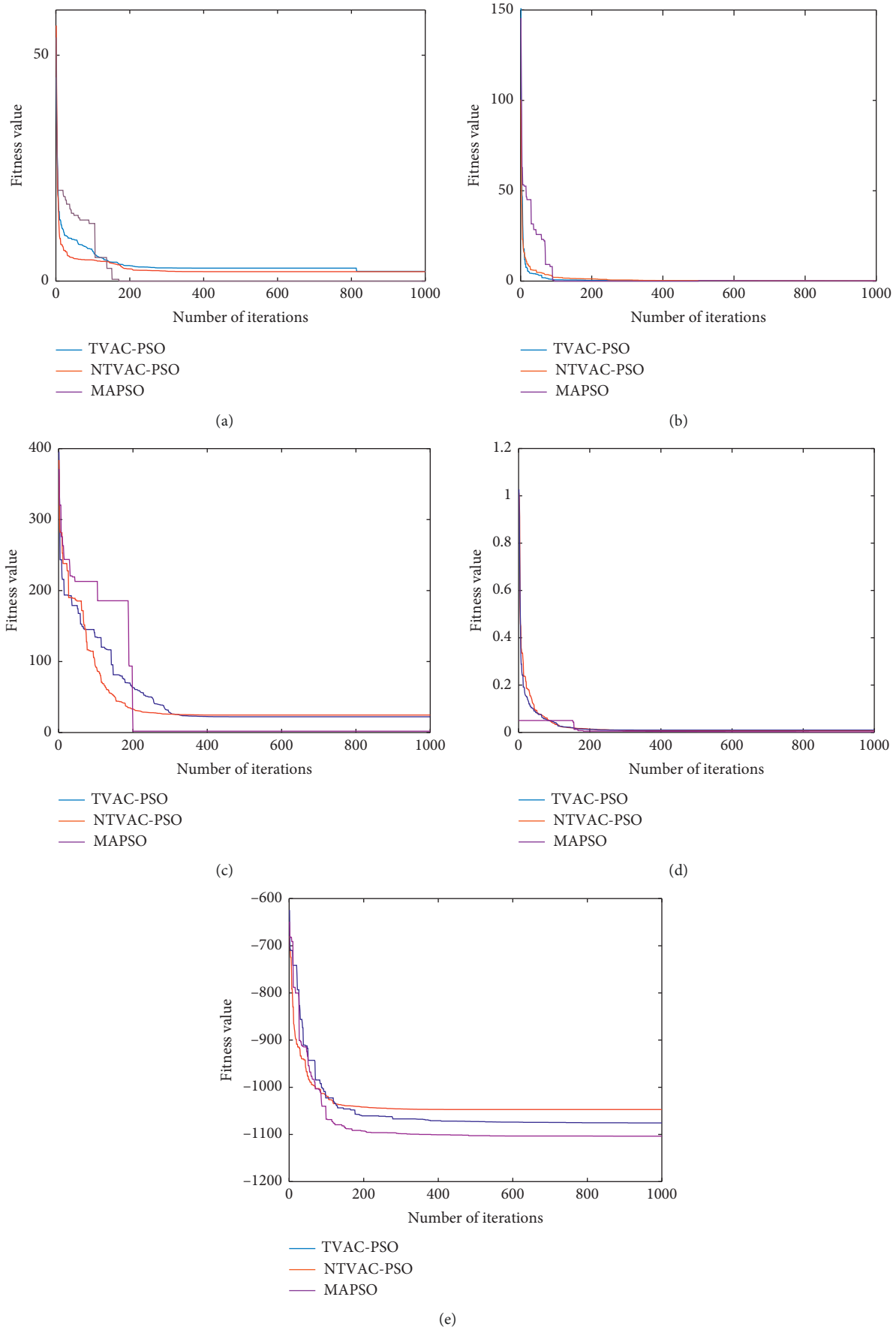

Figure 5: Convergence curve of optimal value of MAPSO and PSO: convergence characteristics of (a) f_1, (b) f_2, (c) f_3, (d) f_4, and (e) f_5.

Table 2: Unit output, network loss, and cost during each period of no wind five-unit system.

Period of time	Output of thermal generating units in different periods (MW)					Network loss (MW)	Cost (USD)
	G1	G2	G3	G4	G5		
1	22.66	98.54	112.67	40.00	139.76	3.64	1265.98
2	10.00	98.54	112.67	78.06	139.76	4.03	1444.85
3	10.00	92.43	112.67	124.91	139.76	4.76	1389.84
4	10.14	98.54	112.67	174.91	139.76	4.49	1659.36
5	10.00	92.52	112.67	209.81	139.76	6.77	1587.79
6	15.09	98.54	152.67	209.82	139.76	7.88	1857.62
7	23.63	98.54	112.67	209.82	189.76	8.42	1965.28
8	12.71	98.54	112.67	209.82	229.52	9.27	1797.29
9	35.55	98.55	126.70	209.82	229.52	10.14	2043.16
10	64.02	98.54	112.67	209.82	229.52	10.57	1996.61
11	75.00	104.04	112.68	209.82	229.52	11.06	2038.04
12	75.00	98.54	112.67	236.01	229.52	11.74	2206.63
13	64.03	98.54	112.67	209.81	229.52	10.57	1996.61
14	49.63	98.54	112.67	209.82	229.52	10.18	1977.71
15	19.63	91.59	112.67	209.81	229.52	9.23	1862.85
16	10.00	75.20	112.67	159.81	229.52	7.21	1892.85
17	10.00	87.59	112.67	124.91	229.52	6.69	1615.27
18	33.64	98.54	129.22	124.91	229.52	7.83	1854.72
19	47.46	98.54	112.67	174.91	229.52	9.10	2061.87
20	64.02	98.54	112.67	209.82	229.52	10.57	1996.62
21	39.36	98.54	112.67	209.82	229.52	9.91	1944.65
22	10.00	98.54	112.67	209.82	181.89	7.92	1863.93
23	20.18	98.54	112.67	161.74	139.76	5.89	1695.06
24	10.00	82.13	110.71	124.91	139.76	4.50	1424.73
Total	192.39	741.76	2300.68	2772.78	4252.66	4703.01	43439.34

Table 3: Comparison of simulation results.

Algorithm	Minimum cost (USD)	Average cost (USD)	Maximum cost (USD)
MSL [53]	49216.81	—	—
SA [10]	47356	—	—
IRCGA [9]	47185	—	—
PS [54]	46530	—	—
DE [9]	45800	—	—
GA [35]	44862.42	44921.76	45893.95
AIS [36]	44385.43	44758.84	45553.77
HS [55]	44367.23	—	—
PSO [35]	44253.24	45657.06	46402.52
ABC [35]	44045.83	44064.73	44218.64
EAPSO [34]	43820	44082	44982
CMAES [56]	43536	43915	44191
MAPSO	**43439.34**	**43751.13**	**44034**

distribution of each unit is shown in Figure 6, which verifies that it satisfies the power balance constraints.

Similarly, in order to verify the search performance of the MAPSO algorithm, the optimal results are compared with those of IMOEA/D-CH and NSGA-II-CH algorithms in [70], as shown in Table 8.

It can be seen from Table 8 that the MAPSO algorithm is still superior to the other two algorithms in solving the dynamic economic model with wind power. Compared with the IMOEA/D-CH algorithm and NSGA-II-CH algorithm, the cost is reduced by 4129 USD and 20229 USD, respectively. Therefore, the advantages of the MAPSO algorithm are reflected in the results.

In order to embody the rationality of wind power integration into the power system, Table 9 gives the comparison of the optimization results of the dispatching model before and after the application of the MAPSO algorithm to wind power integration.

From Table 9, it is clear that the dispatching model with wind farms is lower than that without wind farms, regardless of network loss or cost. In this paper, Tables 4 and 7 are used to extract three periods from 24 periods for comparative analysis: for example, in the first moment, the network loss of the model without wind farms is 19.54 MW, and that of the model with wind farms is 17.75 MW, which is reduced by 1.79 MW and 9.2%. In the 16th period, the network loss of

Table 4: Unit output, network loss, and cost during each period of no wind ten-unit system.

Period of time	Output of thermal generating units in different periods (MW)										Network loss (MW)	Cost (USD)
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10		
1	150.00	135.00	73.00	148.42	164.72	140.39	90.25	120.00	23.82	10.00	19.54	61859
2	150.00	135.00	73.00	98.42	214.72	160.00	117.31	120.00	53.82	10.00	22.48	65257
3	150.00	135.00	131.70	147.78	243.00	140.85	130.00	120.00	48.44	40.00	28.61	72718
4	150.00	135.00	197.70	178.95	243.00	160.00	130.00	120.00	72.05	55.00	35.56	80414
5	150.00	135.00	277.70	228.95	223.73	143.62	128.03	115.31	64.08	53.20	39.59	84957
6	150.00	135.00	340.00	278.95	243.00	153.16	130.00	120.00	71.43	55.00	48.22	93839
7	150.00	210.38	340.00	280.62	243.00	160.00	130.00	120.00	74.16	47.03	53.23	100183
8	150.00	257.45	340.00	300.00	243.00	160.00	130.00	120.00	80.00	55.00	58.61	106791
9	230.00	337.45	340.00	300.00	243.00	160.00	130.00	120.00	80.00	54.41	70.70	123188
10	306.22	379.12	340.00	298.26	243.00	159.62	130.00	120.00	80.00	44.76	79.72	136391
11	311.06	459.12	340.00	300.00	243.00	160.00	130.00	120.00	75.42	54.54	88.07	148757
12	391.06	435.70	340.00	300.00	243.00	160.00	130.00	120.00	67.94	55.00	92.73	156969
13	311.06	425.77	340.00	292.12	243.00	160.00	130.00	120.00	80.00	55.00	84.72	143974
14	231.06	345.77	340.00	300.00	243.00	160.00	130.00	120.00	80.00	44.38	70.77	123555
15	151.06	265.77	340.00	300.00	243.00	160.00	130.00	117.59	80.00	47.15	58.65	107111
16	150.00	185.77	268.46	250.00	225.38	158.69	130.00	120.00	54.77	55.00	43.90	90665
17	150.00	135.73	309.78	244.61	218.43	148.98	127.98	120.00	24.77	39.62	39.69	84755
18	150.88	135.00	340.00	294.61	243.00	155.13	130.00	117.58	54.77	55.00	48.21	93681
19	192.12	215.00	340.00	300.00	243.00	160.00	130.00	120.00	80.00	55.00	58.48	106556
20	272.12	295.00	340.00	300.00	243.00	160.00	130.00	120.00	80.00	55.00	70.64	123440
21	260.65	319.08	340.00	300.00	243.00	159.19	130.00	120.00	71.19	52.74	70.85	124113
22	180.65	239.08	260.00	300.00	238.33	123.65	130.00	120.00	41.19	43.98	49.30	98371
23	150.00	159.39	180.00	252.32	243.00	125.10	130.00	90.00	20.00	13.98	32.25	77864
24	150.00	135.00	122.09	231.39	223.82	101.22	130.00	83.04	22.69	10.00	25.58	69405
Total	4788	5746	6653	6225	5640	3630	3064	2804	1481	1061	1290	2474813

Table 5: Algorithm optimization results and comparison in 24 h.

Algorithm	Computing time (s)	Cost (USD)
MAPSO	**12.26**	**2474813**
IMOEA/D-CH [70]	—	2480200
MAMODE [19]	505	2492451
IBFA [57]	5.2	2481733
HMO-DE-PSO [71]	—	2484000
RCGA/NSGA-II [69]	1080	2516800
AIS [37]	53.56	2519700
PSO [37]	68.47	2572200
EP [37]	72.68	2585400

Table 6: Wind power and system load forecast at different periods.

Period of time	w_{av}	P_D
1	55	1036
2	50	1110
3	65	1258
4	48	1406
5	38	1480
6	48	1628
7	55	1702
8	48	1776
9	32	1924
10	20	2022
11	40	2106
12	50	2150
13	65	2072
14	72	1924
15	90	1776
16	100	1554
17	85	1480
18	68	1628
19	60	1776
20	70	1972
21	75	1924
22	90	1628
23	80	1332
24	75	1184

the model without wind farms is 43.9 MW, and that of the model with wind farms is 38.15 MW, which is reduced by 5.75 MW and 13.1%. In the 24th period, the network loss of the model without wind farms was 25.58 MW, and that of the model with wind farms was 22.15 MW, reduced by 3.43 MW and 13.4%, and the total network loss of the 24 periods was reduced by 95.1 MW. Similarly, in these three periods, the cost savings of wind power grid-connected models are 2726 USD, 7271 USD, and 3633 USD, respectively. The total cost of 24 periods decreased by 119142 USD. It can be seen that the addition of wind power is very beneficial to the efficiency and cost of the system operation.

5.2.4. MAPSO Test of 10 Generators Unit System considering Wind Power Uncertainty. Considering the randomness of wind power, the model adds overestimation and underestimation of penalty costs on the basis of Section 5.2.3; therefore, the model considers (3) and (4) and constraints (6)–(10), dispatching period $H = 24$ h, and time interval is 1 h. The parameters of units and network losses in the system are obtained from [70]. The total installed capacity of the

Table 7: Unit output, network loss, and cost during each period of wind ten-unit system.

Period of time	Output of thermal generating units in different periods (MW)										Network loss (MW)	Cost (USD)
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10		
1	150.00	135.00	73.00	60.00	159.31	160.00	111.17	120.00	20.00	10.11	17.75	59133
2	150.00	135.00	115.96	60.00	209.31	160.00	130.00	90.00	20.00	10.00	20.58	62973
3	150.99	135.00	176.52	110.00	171.37	147.17	130.00	120.00	50.00	27.42	25.69	69439
4	150.00	135.00	256.52	160.00	221.37	160.00	128.69	109.93	20.27	49.28	33.27	78288
5	150.00	135.00	277.17	210.00	243.00	160.00	130.00	111.27	28.33	34.64	37.48	83064
6	151.34	138.96	340.00	260.00	243.00	160.00	106.44	120.00	58.33	46.98	45.41	91564
7	150.00	137.52	340.00	287.57	241.86	160.00	130.00	120.00	80.00	49.25	49.30	94644
8	170.31	201.55	340.00	300.00	233.82	160.00	130.00	120.00	80.00	46.99	54.97	102375
9	250.13	281.55	340.00	300.00	243.00	160.00	130.00	120.00	80.00	55.00	67.81	119564
10	299.50	361.55	340.00	300.00	243.00	160.00	130.00	120.00	80.00	45.96	77.83	133788
11	301.93	440.45	340.00	300.00	243.00	160.00	130.00	120.00	60.51	55.00	84.42	144372
12	374.93	387.89	340.00	300.00	243.00	160.00	130.00	120.00	80.00	51.28	87.35	147870
13	313.60	345.89	340.00	298.78	243.00	160.00	130.00	119.01	80.00	55.00	78.21	134511
14	237.77	265.89	340.00	297.55	243.00	160.00	130.00	120.00	79.98	41.94	64.59	115564
15	157.77	185.89	340.00	295.06	225.21	160.00	125.30	120.00	80.00	49.05	52.14	99136
16	150.00	135.00	260.00	245.60	224.16	145.60	130.00	102.97	55.58	43.54	38.15	83394
17	150.21	135.00	201.58	195.60	243.00	160.00	129.99	120.00	51.71	43.23	34.94	79544
18	150.00	156.04	281.58	232.25	243.00	160.00	125.77	120.00	80.00	55.00	44.05	90295
19	150.00	236.04	340.00	282.25	243.00	160.00	130.00	118.98	63.64	46.41	54.38	102082
20	230.00	316.04	340.00	300.00	243.00	160.00	130.00	120.00	80.00	52.06	68.78	120282
21	211.14	287.77	340.00	300.00	235.31	160.00	124.40	120.00	80.00	55.00	64.46	115275
22	150.00	207.77	260.00	250.00	221.80	155.23	130.00	120.00	50.45	35.82	43.15	90214
23	150.00	135.00	181.34	200.00	179.14	136.87	128.09	116.27	20.45	33.11	28.17	72575
24	150.00	135.00	141.08	158.79	134.22	131.00	130.00	87.42	50.45	12.97	22.15	65728
Total	4650	5166	6645	5704	5373	3756	3060	2776	1430	1005	1195	2355671

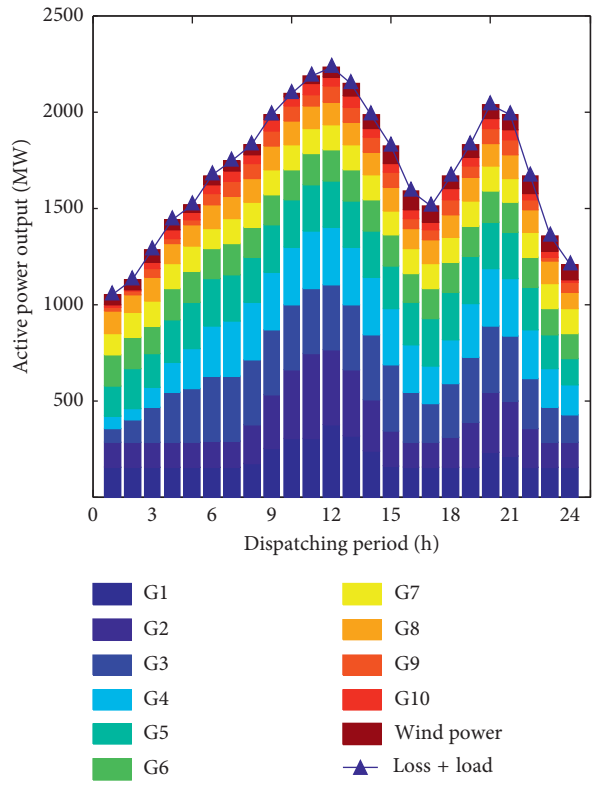

Figure 6: Power balance constraint verification.

wind farm connected to the grid is 100 MW, with a total of 50 wind turbines. The predicted values of wind power and system load in each period are shown in Table 6. w_u% and w_d% are 0.2 and 0.3, respectively, while L_u% and L_d% are 0.05, respectively. $C_{ow,m}$ and $C_{uw,m}$ are 30 and 5, respectively. The dynamic economic dispatching model is run 30 times by using the MAPSO algorithm, and the optimal total cost is 2368764 USD. The thermal power and wind power output, network loss, and cost distribution of each unit within 24 hours are shown in Table 10.

It can be seen from Table 10 that the optimal output, network loss, and cost of each unit in each dispatching period considering wind power uncertainty. The total optimal output of ten thermal generators (G1–G10) and one wind farm (W) under each load is equal to the system load, which satisfies the power balance constraints and the upper and lower power limits of each unit, so the results obtained by MAPSO are reasonable.

Table 8: Comparison of optimization results of each algorithm.

Algorithm	Cost (USD)
MAPSO	**2355671**
IMOEA/D-CH [70]	2359800
NSGA-II-CH [70]	2377700

Table 9: Data comparison of two models.

Models	Total network loss (MW)	Total cost (USD)
Model without wind power	1290.1	2474813
Model with wind power	1195.0	2355671

Table 10: Unit output, network loss, and cost of a ten-unit system considering wind power uncertainty.

Period of time	Output of the thermal power and wind power in different periods (MW)											Network loss (MW)	Cost (USD)
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	W		
1	150.0	135.0	73.0	60.0	189.4	57.0	130.0	79.1	71.3	55.0	54.6	18.2	60057
2	160.6	135.0	135.5	60.0	159.9	107.0	112.3	109.1	41.3	55.0	54.8	20.6	64758
3	150.0	135.0	215.5	106.4	182.9	122.5	82.3	120.0	53.4	51.1	65.3	26.0	70225
4	150.0	135.0	241.0	156.4	223.9	143.0	112.3	115.6	64.6	38.8	58.3	32.8	78433
5	150.0	135.0	321.0	204.3	238.6	127.5	95.3	113.0	80.0	10.0	43.3	37.7	83397
6	221.0	135.0	333.9	243.1	188.6	160.0	125.3	120.0	80.0	10.0	56.4	45.5	93751
7	150.0	177.9	316.2	293.1	238.6	160.0	128.0	109.3	76.1	40.0	61.2	49.1	96149
8	150.0	257.9	340.0	300.0	232.0	160.0	130.0	101.5	59.4	51.8	49.0	55.5	104448
9	230.0	334.9	340.0	300.0	243.0	160.0	130.0	101.9	80.0	35.3	37.0	68.0	121455
10	281.1	370.3	340.0	295.8	243.0	160.0	126.1	120.0	71.4	55.0	35.9	76.5	133219
11	361.1	372.2	337.0	300.0	239.7	155.5	122.2	117.3	67.8	55.0	60.8	82.6	143785
12	319.6	401.9	340.0	300.0	240.6	160.0	121.4	120.0	74.6	55.0	100.0	82.7	142804
13	300.0	354.8	340.0	300.0	242.1	160.0	130.0	119.5	80.0	25.0	96.6	75.7	132993
14	241.0	274.8	340.0	300.0	232.6	160.0	108.9	107.7	80.0	55.0	87.8	63.9	116751
15	161.0	208.4	340.0	300.0	243.0	129.3	130.0	120.0	50.0	55.0	91.2	52.5	99796
16	150.0	135.0	310.4	250.0	229.7	147.4	120.7	100.8	20.0	28.6	100.0	38.4	84286
17	150.0	135.0	340.0	200.0	179.7	138.7	130.0	99.2	20.0	34.3	88.2	35.4	81265
18	150.0	183.8	330.1	250.0	229.7	160.0	100.0	101.6	42.2	55.0	69.9	44.5	92439
19	184.9	225.5	340.0	300.0	206.1	160.0	130.0	120.0	52.1	50.2	61.2	54.5	103093
20	227.9	301.0	340.0	294.0	243.0	160.0	130.0	120.0	80.0	42.0	100.0	66.3	118000
21	213.0	307.2	340.0	300.0	220.9	129.6	130.0	120.0	76.3	49.6	100.0	63.3	114992
22	150.0	227.2	275.6	256.3	216.0	126.4	100.0	120.0	80.0	19.6	100.0	43.1	91493
23	150.0	147.2	195.6	247.8	166.0	76.4	86.4	120.0	77.8	10.0	83.0	28.6	74200
24	150.0	140.9	115.6	197.8	175.0	82.3	86.4	106.5	65.8	10.0	76.1	22.5	66973
Total	4701	5366	6940	5815	5204	3303	2798	2682	1544	946	1730	1184	2368764

Table 11: Data comparison of two models.

Models	Total cost (USD)
Not considering wind power uncertainty	2355671
Considering wind power uncertainty	2368764

In order to reflect the rationality of considering the randomness of wind power, Table 11 lists the data comparison results considering and not considering the randomness of wind power dispatching models.

From Table 11, it is obviously that the cost of model considering wind power uncertainty is 13093 USD higher than that of model not considering wind power uncertainty. This is because the former model adds overestimation and underestimation costs to the original model. Although the cost increases, the impact of wind power stochastic characteristics on the grid cannot be neglected in the actual wind power grid operation. Therefore, the slight increase in costs is reasonable. On the contrary, it also shows that the MAPSO algorithm has good search performance in solving the problem of the wind power stochastic model.

6. Conclusions

Dynamic economic dispatching models are built. Fuel cost of thermal power units, valve-point effect cost, and overestimation and underestimation costs of wind power uncertainty are considered in the objective function of the model. Network loss and positive and negative spinning reserve demand added by uncertainty of wind power grid connection are considered in the constraints. For solving the model, a modified antipredator particle swarm optimization algorithm is proposed, and the following work is done:

(1) The algorithm considers the distance between the global worst particle and other particles, introduces the inertia weight ω_{gw} to control the degree of particle movement, and constructs the formula by using the characteristics of sigmoid function, normalization, and linear decreasing. In view of the shortcomings of time-varying acceleration coefficient, a nonlinear time-varying acceleration coefficient is proposed, which improves the local search and global search ability of the particle.

(2) In order to verify the effectiveness of the improved algorithm, several benchmark functions and power grid system models are used to analyze the proposed algorithm. The simulation results show that the MAPSO algorithm is practical and superior in model solving.

Abbreviations

MAPSO: Modified antipredatory particle swarm optimization
APSO: Antipredatory particle swarm optimization
DED: Dynamic economic dispatching
PDF: Probability density function
GA: Genetic algorithm
SA: Simulated annealing
DE: Differential evolution
ABC: Artificial bee colony
AIS: Artificial immune system

EP: Evolutionary programming
MSL: Maclaurin series-based Lagrangian
PS: Pattern search
HS: Harmony search
CMAES: Covariance matrix adapted evolution strategy
IBFA: Improved bacterial foraging algorithm
TVAC: Time-varying acceleration coefficients
NTVAC: Nonlinear time-varying acceleration coefficient
LDIW: Linear decreasing inertia weight
IRCGA: Improved real-coded genetic algorithm
EAPSO: Enhanced adaptive particle swarm optimization
IMOEA/D-CH: Improved multiobjective evolutionary algorithm based on decomposition with constraints handling
MAMODE: Multiobjective differential evolution with expanded double selection and adaptive random restart
RCGA/NSGA-II: Real-coded genetic algorithm and nondominated sorting genetic algorithm-II
HMO-DE-PSO: Hybrid multiobjective DE and PSO
NSGA-II-CH: Nondominated sorting genetic algorithm-II with constraints handling.

Nomenclature

$f(P, w)$: The total cost of dispatching in single period
$F(P_n)$: The fuel cost of thermal unit n
a_n, b_n, and c_n: The fuel cost coefficients of thermal unit n
$E(P_n)$: The energy consumption cost caused by the valve-point effect of thermal unit n
d_n and e_n: The coefficients of thermal unit n due to valve-point effect
P_n: The active power output of thermal unit n
$P_{n,\max}$ and $P_{n,\min}$: The maximum and minimum limits of the output of thermal unit n
N: The total number of thermal units
$w_{\mathrm{av},m}$: The available output of the mth wind farm
w_m: The actual output of the mth wind farm
$U_m(w)$: The penalty cost of overestimation of wind power output
$D_m(w)$: The penalty cost of underestimation of wind power output
H: The total number of dispatching periods
C: The total cost of dynamic economic dispatch in H period
M: The total number of wind farms
P_{D}: The predicted value of active power in single load period
P_{L}: The network loss in single load period
$B_{i,j}$, $B_{i,0}$, and $B_{0,0}$: The loss coefficients of network loss
$w_{m,\max}$: The maximum installed capacity of the mth wind farm
U_{Rn} and D_{Rn}: The up and down ramp rates of the nth unit, respectively
ΔT: The dispatch time interval
$L_{\mathrm{u}}\%$ and $L_{\mathrm{d}}\%$: The demand coefficients of the system load prediction errors for positive and negative spinning reserve
$w_{\mathrm{u}}\%$ and $w_{\mathrm{d}}\%$: The demand coefficients of wind power prediction errors for positive and negative spinning reserve, respectively
U_n and D_n: The positive and negative spinning reserve capacity provided by the nth thermal unit, respectively
T_{10}: The response time of spinning reserve
k and c: Shape factor and scale factor of Weibull distribution, respectively
w_r: The rated capacity of the wind farm
V, v_i, v_r, and v_o: Actual, cut-in, rated, and cut-out wind speeds, respectively
$P\{\cdot\}$: The probability of occurrence of events
k_r and kp: The penalty coefficients for overestimation and underestimation
V_i^t: The velocity of the ith particle at the tth iteration
ω: Inertia weight of the particle
$\omega_{\max}$ and $\omega_{\min}$: The maximum and minimum inertia weight of the particle
$c_{1\mathrm{g}}$ and $c_{1\mathrm{b}}$: The acceleration coefficients of the particle flying to its own best position and flying away its own worst position, respectively
$c_{2\mathrm{g}}$ and $c_{2\mathrm{b}}$: The acceleration coefficients of the best position of the particle flying to the group and the worst position of the particle flying away from the group, respectively
r_1, r_2, r_3, and r_4: The random number uniformly distributed on [0, 1], respectively
$pbest_i^t$ and $pworst_i^t$: The individual optimal value and worst value of the ith particle at the tth iteration, respectively
$gbest^t$ and $gworst^t$: The global optimal value and worst value of the particle swarm at the tth iteration
x_i^t: The displacement of the ith particle at the tth iteration
η and δ: The nonlinear parameters
θ and $\theta_{\max}$: The current number and total number of iterations, respectively
ω_{gw}: The particle's evasive inertia weight
A and β: The linear decreasing coefficients of ω_{gw}
μ: The inclination coefficient of ω_{gw}
l_i: The distance between the particle i and the global worst solution $gworst$
$l_{\min}$ and $l_{\max}$: The minimum and maximum distances of the global worst solution to the particle swarm, respectively
l_i^{normal}: The normalized value of l_i
I: The number of particles.

Data Availability

The data used to support the findings of this study will be provided by the author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This study was funded by National Natural Science Foundation of China (61703404) and China Postdoctoral Science Foundation (2016M591956).

References

[1] F. Hu, K. J. Hughes, D. B. Ingham, L. Ma, and M. Pourkashanian, "Dynamic economic and emission dispatch model considering wind power under energy market reform: a case study," *International Journal of Electrical Power & Energy Systems*, vol. 110, pp. 184–196, 2019.

[2] Y. Liu and N.-K. C. Nair, "A two-stage stochastic dynamic economic dispatch model considering wind uncertainty," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 2, pp. 819–829, 2016.

[3] D. C. Walters and G. B. Sheble, "Genetic algorithm solution of economic dispatch with valve point loading," *IEEE Transactions on Power Systems*, vol. 8, no. 3, pp. 1325–1332, 1993.

[4] H. Barati and M. Sadeghi, "An efficient hybrid MPSO-GA algorithm for solving non-smooth/non-convex economic dispatch problem with practical constraints," *Ain Shams Engineering Journal*, vol. 9, no. 4, pp. 1279–1287, 2018.

[5] L. Mellouk, M. Boulmalf, A. Aaroud, K. Zine-Dine, and D. Benhaddou, "Genetic algorithm to solve demand side management and economic dispatch problem," *Procedia Computer Science*, vol. 130, pp. 611–618, 2018.

[6] T. C. Bora, V. C. Mariani, and L. d. S. Coelho, "Multi-objective optimization of the environmental-economic dispatch with reinforcement learning based on non-dominated sorting genetic algorithm," *Applied Thermal Engineering*, vol. 146, pp. 688–700, 2019.

[7] C. H. Ram Jethmalani, S. P. Simon, K. Sundareswaran, P. Srinivasa Rao Nayak, and N. P. Padhy, "Real coded genetic algorithm based transmission system loss estimation in dynamic economic dispatch problem," *Alexandria Engineering Journal*, vol. 57, no. 4, pp. 3535–3547, 2018.

[8] M. Nemati, M. Braun, and S. Tenbohlen, "Optimization of unit commitment and economic dispatch in microgrids based on genetic algorithm and mixed integer linear programming," *Applied Energy*, vol. 210, pp. 944–963, 2018.

[9] J. K. Pattanaik, M. Basu, and D. P. Dash, "Improved real coded genetic algorithm for dynamic economic dispatch," *Journal of Electrical Systems and Information Technology*, vol. 5, no. 3, pp. 349–362, 2018.

[10] C. K. Panigrahi, P. K. Chattopadhyay, R. N. Chakrabarti, and M. Basu, "Simulated annealing technique for dynamic economic dispatch," *Electric Power Components and Systems*, vol. 34, no. 5, pp. 577–586, 2006.

[11] S. Pothiya, I. Ngamroo, and W. Kongprawechnon, "Application of multiple tabu search algorithm to solve dynamic economic dispatch considering generator constraints," *Energy Conversion and Management*, vol. 49, no. 4, pp. 506–516, 2008.

[12] W. Sa-ngiamvibool, S. Pothiya, and I. Ngamroo, "Multiple tabu search algorithm for economic dispatch problem considering valve-point effects," *International Journal of Electrical Power & Energy Systems*, vol. 33, no. 4, pp. 846–854, 2011.

[13] X. Shen, D. X. Zou, X. Zhang, Q. Zhng, and P. Xiao, "A phase-based adaptive differential evolution algorithm for the economic load dispatch considering valve-point effects and transmission losses," *Mathematical Problems in Engineering*, vol. 2018, Article ID 4585403, 24 pages, 2018.

[14] J. S. Dhaliwal and J. S. Dhillon, "Profit based unit commitment using memetic binary differential evolution algorithm," *Applied Soft Computing*, vol. 81, Article ID 105502, 2019.

[15] M. Ghasemi, M. Taghizadeh, S. Ghavidel, and A. Abbasian, "Colonial competitive differential evolution: an experimental study for optimal economic load dispatch," *Applied Soft Computing*, vol. 40, pp. 342–363, 2016.

[16] H. Gustavo and Czaikoski, "Differential evolution with ensemble of mutation operators in GPGPUs for economic load dispatch," *Revista de Informática Teórica e Aplicada*, vol. 23, pp. 2310–2332, 2016.

[17] J. X. V. Neto, G. Reynoso-Meza, T. H. Ruppel, V. C. Mariani, and L. d. S. Coelho, "Solving non-smooth economic dispatch by a new combination of continuous GRASP algorithm and differential evolution," *International Journal of Electrical Power & Energy Systems*, vol. 84, pp. 13–24, 2017.

[18] D. Zou, S. Li, X. Kong, H. Ouyang, and Z. Li, "Solving the dynamic economic dispatch by a memory-based global differential evolution and a repair technique of constraint handling," *Energy*, vol. 147, pp. 59–80, 2018.

[19] X. Jiang, J. Zhou, H. Wang, and Y. Zhang, "Dynamic environmental economic dispatch using multiobjective differential evolution algorithm with expanded double selection and adaptive random restart," *International Journal of Electrical Power & Energy Systems*, vol. 49, pp. 399–407, 2013.

[20] A. Shukla and S. N. Singh, "Advanced three-stage pseudo-inspired weight-improved crazy particle swarm optimization for unit commitment problem," *Energy*, vol. 96, pp. 23–36, 2016.

[21] Q. Qin, S. Cheng, Q. Zhang, Y. Wei, and Y. Shi, "Multiple strategies based orthogonal design particle swarm optimizer for numerical optimization," *Computers & Operations Research*, vol. 60, pp. 91–110, 2015.

[22] M. K. Modi, A. Swarnkar, N. Gupta, K. R. Niazi, and R. C. Bansal, "Stochastic economic load dispatch with multiple fuels using improved particle swarm optimization," *IFAC-PapersOnLine*, vol. 48, no. 30, pp. 490–494, 2015.

[23] Q. Qin, S. Cheng, X. Chu, X. Lei, and Y. Shi, "Solving non-convex/non-smooth economic load dispatch problems via an enhanced particle swarm optimization," *Applied Soft Computing*, vol. 59, pp. 229–242, 2017.

[24] M. Mohammadian, A. Lorestani, and M. M. Ardehali, "Optimization of single and multi-areas economic dispatch problems based on evolutionary particle swarm optimization algorithm," *Energy*, vol. 161, pp. 710–724, 2018.

[25] M. Gholamghasemi, E. Akbari, M. B. Asadpoor, and M. Ghasemi, "A new solution to the non-convex economic load dispatch problems using phasor particle swarm optimization," *Applied Soft Computing*, vol. 79, pp. 111–124, 2019.

[26] H. Lu, P. Sriyanyong, Y. H. Song, and T. Dillon, "Experimental study of a new hybrid PSO with mutation for economic dispatch with non-smooth cost function," *International Journal of Electrical Power & Energy Systems*, vol. 32, no. 9, pp. 921–935, 2010.

[27] J. Sun, V. Palade, X.-J. Wu, W. Fang, and Z. Wang, "Solving the power economic dispatch problem with generator constraints by random drift particle swarm optimization," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 222–232, 2014.

[28] B. K. Panigrahi, V. Ravikumar Pandi, and S. Das, "Adaptive particle swarm optimization approach for static and dynamic economic load dispatch," *Energy Conversion and Management*, vol. 49, no. 6, pp. 1407–1415, 2008.

[29] A. P. Lin and W. Sun, "Multi-leader comprehensive learning particle swarm optimization with adaptive mutation for economic load dispatch problems," *Energies*, vol. 12, no. 1, p. 116, 2019.

[30] G. Xiong and D. Shi, "Orthogonal learning competitive swarm optimizer for economic dispatch problems," *Applied Soft Computing*, vol. 66, pp. 134–148, 2018.

[31] G. Yuan and W. Yang, "Study on optimization of economic dispatching of electric power system based on hybrid intelligent algorithms (PSO and AFSA)," *Energy*, vol. 183, pp. 926–935, 2019.

[32] K. Mason, J. Duggan, and E. Howley, "Multi-objective dynamic economic emission dispatch using particle swarm optimisation variants," *Neurocomputing*, vol. 270, pp. 188–197, 2017.

[33] S. Velamuri and S. Sreejith, "Reserve constrained economic dispatch incorporating solar farm using particle swarm optimization," *International Journal of Energy Research*, vol. 6, pp. 150–156, 2016.

[34] T. Niknam and F. Golestaneh, "Enhanced adaptive particle swarm optimisation algorithm for dynamic economic dispatch of units considering valve-point effects and ramp rates," *IET Generation, Transmission & Distribution*, vol. 6, no. 5, pp. 424–435, 2012.

[35] S. Hemamalini and S. P. Simon, "Dynamic economic dispatch using artificial bee colony algorithm for units with valve-point effect," *European Transactions on Electrical Power*, vol. 21, no. 1, pp. 70–81, 2011.

[36] S. Hemamalini and S. P. Simon, "Dynamic economic dispatch using artificial immune system for units with valve-point effect," *International Journal of Electrical Power & Energy Systems*, vol. 33, no. 4, pp. 868–874, 2011.

[37] M. Basu, "Artificial immune system for dynamic economic dispatch," *International Journal of Electrical Power & Energy Systems*, vol. 33, no. 1, pp. 131–136, 2011.

[38] M. Basu, "Fast convergence evolutionary programming for economic dispatch problems," *IET Generation, Transmission & Distribution*, vol. 11, no. 16, pp. 4009–4017, 2017.

[39] M. Basu, "Fast convergence evolutionary programming for multi-area economic dispatch," *Electric Power Components and Systems*, vol. 45, no. 15, pp. 1629–1637, 2017.

[40] C. Christober and A. Rajan, "A solution to the economic dispatch using EP based SA algorithm on large scale power system," *International Journal of Electrical Power & Energy Systems*, vol. 32, no. 6, pp. 583–591, 2010.

[41] D. McLarty, N. Panossian, F. Jabbari, and A. Traverso, "Dynamic economic dispatch using complementary quadratic programming," *Energy*, vol. 166, pp. 755–764, 2019.

[42] U. Krishnasamy and D. Nanjundappan, "Hybrid weighted probabilistic neural network and biogeography based optimization for dynamic economic dispatch of integrated multiple-fuel and wind power plants," *International Journal of Electrical Power & Energy Systems*, vol. 77, pp. 385–394, 2016.

[43] G. Xiong and D. Shi, "Hybrid biogeography-based optimization with brain storm optimization for non-convex dynamic economic dispatch with valve-point effects," *Energy*, vol. 157, pp. 424–435, 2018.

[44] E. B. Elanchezhian, S. Subramanian, and S. Ganesan, "Economic power dispatch with cubic cost models using teaching learning algorithm," *IET Generation, Transmission & Distribution*, vol. 8, no. 7, pp. 1187–1202, 2014.

[45] X. He, Y. Rao, and J. Huang, "A novel algorithm for economic load dispatch of power systems," *Neuro Computing*, vol. 171, pp. 14540–14561, 2016.

[46] P. Zakian and A. Kaveh, "Economic dispatch of power systems using an adaptive charged system search algorithm," *Applied Soft Computing*, vol. 73, pp. 607–622, 2018.

[47] A. Y. Abdelaziz, E. S. Ali, and S. M. Abd Elazim, "Implementation of flower pollination algorithm for solving economic load dispatch and combined economic emission dispatch problems in power systems," *Energy*, vol. 101, pp. 506–518, 2016.

[48] Y. Labbi, D. B. Attous, H. A. Gabbar, B. Mahdad, and A. Zidan, "A new rooted tree optimization algorithm for economic dispatch with valve-point effect," *International Journal of Electrical Power & Energy Systems*, vol. 79, pp. 298–311, 2016.

[49] N. J. Singh, J. S. Dhillon, and D. P. Kothari, "Synergic predator-prey optimization for economic thermal power dispatch problem," *Applied Soft Computing*, vol. 43, pp. 298–311, 2016.

[50] M. Basu, "Kinetic gas molecule optimization for nonconvex economic dispatch problem," *International Journal of Electrical Power & Energy Systems*, vol. 80, pp. 325–332, 2016.

[51] D. Singh and J. S. Dhillon, "Ameliorated grey wolf optimization for economic load dispatch problem," *Energy*, vol. 169, pp. 398–419, 2019.

[52] F. Mohammadi and H. Abdi, "A modified crow search algorithm (MCSA) for solving economic load dispatch problem," *Applied Soft Computing*, vol. 71, pp. 51–65, 2018.

[53] S. Hemamalini and S. P. Simon, "Dynamic economic dispatch using Maclaurin series based Lagrangian method," *Energy Conversion and Management*, vol. 51, no. 11, pp. 2212–2219, 2010.

[54] J. S. Alsumait, M. Qasem, J. K. Sykulski, and A. K. Al-Othman, "An improved Pattern Search based algorithm to solve the Dynamic Economic Dispatch problem with valve-point effect," *Energy Conversion and Management*, vol. 51, no. 10, pp. 2062–2067, 2010.

[55] V. Ravikumar Pandi and B. K. Panigrahi, "Dynamic economic load dispatch using hybrid swarm intelligence based harmony search algorithm," *Expert Systems with Applications*, vol. 38, no. 7, pp. 8509–8514, 2011.

[56] P. S. Manoharan, P. S. Kannan, S. Baskar, M. Willjuice Iruthayarajan, and V. Dhananjeyan, "Covariance matrix adapted evolution strategy algorithm-based solution to dynamic economic dispatch problems," *Engineering Optimization*, vol. 41, no. 7, pp. 635–657, 2009.

[57] N. Pandit, A. Tripathi, S. Tapaswi, and M. Pandit, "An improved bacterial foraging algorithm for combined static/dynamic environmental economic dispatch," *Applied Soft Computing*, vol. 12, no. 11, pp. 3500–3513, 2012.

[58] E. S. Ali and S. M. Abd Elazim, "Mine blast algorithm for environmental economic load dispatch with valve loading effect," *Neural Computing and Applications*, vol. 30, no. 1, pp. 261–270, 2018.

[59] P. N. Snganthan, "Particle swarm optimizer with neighborhood operator," in *Proceedings of the Congress on*

Evolutionary Computation, pp. 1958–1962, Washington, DC, USA, July 1999.

[60] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection: convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, no. 6, pp. 317–325, 2003.

[61] G. Venter and J. Sobieszczanski-Sobieski, "Multidisciplinary optimization of a transport aircraft wing using particle swarm optimization," *Structural and Multidisciplinary Optimization*, vol. 26, no. 1-2, pp. 121–131, 2004.

[62] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.

[63] M. Kheshti, L. Ding, S. Ma, and B. Zhao, "Double weighted particle swarm optimization to non-convex wind penetrated emission/economic dispatch and multiple fuel option systems," *Renewable Energy*, vol. 125, pp. 1021–1037, 2018.

[64] D. Zou, S. Li, Z. Li, and X. Kong, "A new global particle swarm optimization for the economic emission dispatch with or without transmission losses," *Energy Conversion and Management*, vol. 139, pp. 45–70, 2017.

[65] W. T. Elsayed, Y. G. Hegazy, M. S. El-bages, and F. M. Bendary, "Improved random drift particle swarm optimization with self-adaptive mechanism for solving the power economic dispatch problem," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1017–1026, 2017.

[66] F. Yao, Z. Y. Dong, K. Meng, Z. Xu, H. H.-C. Iu, and K. P. Wong, "Quantum-Inspired particle swarm optimization for power system operations considering wind power uncertainty and carbon tax in Australia," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 4, pp. 880–888, 2012.

[67] A. I. Selvakumar and K. Thanushkodi, "Anti-predatory particle swarm optimization: solution to nonconvex economic dispatch problems: solution to nonconvex economic dispatch problems," *Electric Power Systems Research*, vol. 78, no. 1, pp. 2–10, 2008.

[68] Y. H. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Proceedings of the 7th International Conference on Evolutionary Programming VII*, San Diago, CA, USA, March 1998.

[69] M. Basu, "Dynamic economic emission dispatch using nondominated sorting genetic algorithm-II," *International Journal of Electrical Power & Energy Systems*, vol. 30, no. 2, pp. 140–149, 2008.

[70] Y. S. Zhu, J. Wang, and B. Y. Qu, "Multi-objective dynamic economic emission dispatching of power grid containing wind farms," *Power Systems Technology*, vol. 39, pp. 1315–1322, 2015.

[71] G. Liu, Y. L. Zhu, and W. Jiang, "Wind-thermal dynamic economic emission dispatch with a hybrid multi-objective algorithm based on wind speed statistical analysis," *IET Generation, Transmission & Distribution*, vol. 12, no. 17, pp. 3972–3984, 2018.