ORIGINAL ARTICLE

Multidimensional Firefly Algorithm for Solving Day-Ahead Scheduling Optimization in Microgrid

YuDe Yang[1,2] · JinLian Qiu[1,2] · ZhiJun Qin[1]

Abstract

In this paper, an improved metaheuristic optimization algorithm based on the firefly algorithm, called multidimensional firefly algorithm (MDFA), is presented for solving day-ahead scheduling optimization in a microgrid. The proposed algorithm takes the output of power generations among a quantity of distributed energy resources during 24 h together rather than a single hour as a firefly separately. The proposed algorithm is combined with strategy of solving equality constraint replacing the use of the penalty-function technique. It is also enhanced by using a novel method in parameters self-adaption instead of applying fixed values, resulting in avoiding tuning frequently the algorithm parameters during the process of optimization. The MDFA is utilized for optimization of energy production cost in a microgrid. The superiority of the MDFA is demonstrated by using the classic test power system proved in the previous literature. The solutions obtained by MDFA are compared with the results found by five famous optimization algorithms. The high performance of MDFA is established by the quality with the minimum total cost, the reliability of gained solutions, the speed of convergence, and the ability to satisfy various constraints.

1 Introduction

The emergence of the microgrid integrates smart infrastructure, various technologies, advanced management [1]. Compared with traditional power grids, the microgrid has a variety of advantages including more resilient, sustainable, and reliable [2–4]. Distributed energy resources (DERs) are primary alternatives for supplying the incremental demand of the electrical market in the microgrid. The load is preferentially supplied by the DERs in the microgrid, improving

energy efficiency and reducing power transmission losses as DERs are located near the load.

Nowadays, economic dispatch (ED) [5, 6] problem of the microgrid as an optimization problem [7, 8] is eager to be solved by appropriately scheduling the output of each power generator so as not to waste clean power resources and minimize the operation cost at the same time [9]. The ED problem in the microgrid has attracted more extensive attention from plenty of scholars in recent years [10–16]. Literature reviews show that different technologies have been proposed for solving the optimization problems. There are many metaheuristic optimization methods including particle swarm optimization (PSO) [17], differential evolution (DE) [18], and other algorithms that have successfully solved the economic dispatch problem. They have a powerful ability to search for global or near-global solutions of nonconvex and complex optimization problems. Reference [19] uses multi-objective particle swarm optimization (MOPSO) to find the optimal configuration of off-grid hybrid microgrid system and find their optimal component size in Swedish. In [20], a memory-based genetic algorithm (MGA) has been used for minimizing the energy production cost of the

✉ JinLian Qiu
651231855@qq.com

YuDe Yang
yangyude@gxu.edu.cn

ZhiJun Qin
zjqin@gxu.edu.cn

1 School of Electrical Engineering, Guangxi University, No.100, Daxue East Road, Nanning City, Guangxi, China

2 Guangxi Key Laboratory of Power System Optimization and Energy Technology, Guangxi University, No.100, Daxue East Road, Nanning City, Guangxi, China

day-ahead scheduling problem. In [21], the additive increase multiplicative decrease (AIMD) algorithm is developed to schedule the task of power generation in an optimized way among generators. In [22], four methods including PSO, lambda logic (LL), lambda iteration (LI) and direct search method (DSM) in solving economic dispatch problem are proposed to minimize the fuel cost of distributed generations considering operational constraints. Reference [23] has used HOMER and GAMS (CPLEX solver) to solve the optimization of mixed-integer linear programming problem. Authors in [24] have proposed a combination of differential evolution (DE) and harmony search (HS) to solve the optimization problem of an optimal day-ahead scheduling model. An advanced real-time energy management system is proposed to optimize microgrid performance in [25]. The authors capitalize the binary particle swarm optimization (BPSO) algorithm to maximize the power of the available renewable energy resources while reducing the energy cost, carbon dioxide and pollutant emissions at the same time.

However, there exists a dissatisfied limitation in many of the optimization algorithms reported above. Before we put these algorithms into application, we have to adjust the algorithm parameters on the particular problem. It is an intricate problem that the algorithms need to adapt their parameters due to changes in system load or unit constraints. To solve this defect, the algorithm has to add the technique for adaptively adjusting parameter values. But this method reduces computational efficiency since additional computational effort must be required. Thus, adding self-adjusting parameter will be highly efficient [26]. The firefly algorithm (FA) regarded as an efficiency optimization algorithm has been proposed in [27]. Compared with many optimization algorithms, Yang confirmed that the FA has especial superiority in different aspects including numerical stability, calculation accuracy, and convergence rate [28]. The application of FA for solving ED problems in small or medium power systems is quite successful [29]. In general, current algorithms related to FA have unique superiorities over other recently developed algorithms in solving various optimization problems with a single moment [30, 31]. Most studies for solving ED problems related to FA regard one feasible solution whose vector dimension is $1 \times D$ or $D \times 1$ as a firefly (D is a positive integer). They are efficient for solving the ED problem of a single moment. However, when solving the ED problem of day-ahead scheduling for multiple moments, we must separate numerous moments into suitable single moments if we want to use the above algorithms, whose superiorities will lose because of low efficiency. Therefore, a novel algorithm called multidimensional firefly algorithm (MDFA) is presented to deal with the ED problem. In MDFA, the vector dimension of a firefly is $D \times E$ (E is a positive integer) which is essentially different from FA, and its superiority will be proved in this paper.

In this paper, MDFA is applied to solve the day-ahead scheduling optimization of the microgrid among DERs. We propose two modifications to improve significantly the MDFA efficiency which is convenient to obtain the minimum solutions. One of the proposed modifications is adding a novel strategy in parameter self-adaption selection rather than using a fixed value. Another is exploring a new attractiveness calculation strategy instead of the former attractiveness equation. Besides, in most optimization methods, the penalty-function technique is used to handle equality constraints. However, suppose we use penalty-function to deal with equality constraint in MDFA. In that case, it is hard for MDFA to generate feasible solutions, and we are confused about choosing appropriate penalty factors during the process of optimization, resulting in generating undesirable solutions. Thus, a strategy is proposed for solving equality constraints in MDFA without adding any additional factors to the objective function. This strategy not only helps MDFA handle constraints but also help MDFA keep away from result deviation. In this way, the strategy guarantees the fireflies always be a feasible solution. Indeed, the proposed methods have a reliable and positive effect on the convergence of the algorithm and the quality of the solutions. Results obtained by the MDFA for solving ED problems in the microgrid are compared with the results by other algorithms in previous literature. Simulation results accentuate the superiority of the proposed MDFA technique over the other methods.

2 Mathematical Formulation of the ED Problem

2.1 Generation Cost

The microgrid faces a variety of methods for portioning the requested power among DERs. The best way is to achieve the minimum operating cost of microgrid through the reasonable allocation of the distributed generations. It is generally accepted that the generation cost function for DERs can be modeled as a quadratic function, which can be written as follows [32, 33]:

$$C_i(t) = U_i \times (a_i \times (P_i(t))^2 + b_i \times P_i(t) + c_i) \tag{1}$$

where $C_i(t)$ is the hourly cost (\$), and $P_i(t)$ (MW) denotes output power of the i-th generation at period t, respectively. U_i is a binary variable. When $P_i(t)$ is equal to 0, U_i is 0, otherwise U_i is 1. And a_i, b_i, and c_i are the cost coefficients of i-th DER.

2.2 Objective Function and Constraints

The microgrid aggregates various local production or consumption units that act in the electricity market as a single entity. In this paper, there are photovoltaic panels (PV), combined heat and power (CHP), and wind plants (WP) as DERs. In order to minimize the generation cost of the required power among DERs, an objective function (OF) that defined based on a quadratic cost function can be written as:

$$Min\ OF \sum_{t=0}^{T} \sum_{i=1}^{N_{DERs}} C_i(t) = \sum_{t=0}^{T} \sum_{i=1}^{N_{DERs}} U_i \times (a_i \times (P_i(t))^2 + b_i \times P_i(t) + c_i)$$

(2)

Subject to:

$$\sum_{i=1}^{N_{DERs}} P_i(t) = P_L(t)$$

(3)

$$P_{CHP}^{min}(t) \le P_{CHP}(t) \le P_{CHP}^{max}(t)$$

(4)

$$P_{PV}^{min}(t) \le P_{PV}(t) \le P_{PV}^{max}(t)$$

(5)

$$P_{WP}^{min}(t) \le P_{WP}(t) \le P_{WP}^{max}(t)$$

(6)

where T stands for total hours. In this paper, T is equal to 23. N_{DERs} is the total number of DERs participating in the electricity market of microgrid, and its value is 6 in this paper. $P_L(t)$ means the load demand in the microgrid at time t. $P_{DER}^{min}(t)$ and $P_{DER}^{max}(t)$ are the minimum and the maximum generation power of DER at time t.

In conventional optimization theory, the measure to handle the equality constraint is using penalty-function. From this perspective, an optimization problem with equality constraint can be converted to an optimization problem without equality constraint. By adding penalty-function, the problem defined by (2) is modified as follows:

$$Min\ OF = \sum_{t=0}^{T} \sum_{i=1}^{N_{DERs}} C_i(t) = \sum_{t=0}^{T} \sum_{i=1}^{N_{DERs}} U_i \times (a_i \times (P_i(t))^2$$
$$+ b_i \times P_i(t) + c_i) + P_f \times \left| \sum_{i=1}^{N_{DERs}} P_i(t) - P_L(t) \right|$$

(7)

where P_f is the penalty factor which plays a vital role in solving equality constraint in penalty-function.

2.3 Strategy for Solving Equality Constraint

Penalty-function is widely used in most documents to deal with an equality constraint. Owing to the traditional statement, an optimization problem without equality constraint can replace an optimization problem with equality constraint. Penalty-function has a great effect on dealing with equality constraint on many occasions, but it is not mature enough in the application of MDFA. In MDFA, using penalty-function may cause deviation during the period of searching for optimal solution. The penalty factor is the key to find global solutions, and it is determined by the specific data in penalty-function. Hence, it is hard to choose the most suitable penalty factor which greatly increases the computational time during the process of finding the best solution. Furthermore, due to the application of penalty-function, the ability of exploiting and exploring is weakened significantly in MDFA. In this case, the solution found by penalty-function can't be guaranteed to be optimal because of premature convergence, especially in disposing of complex nonlinear problems. Thus, penalty-function won't be applied in equality constraint during the process of MDFA in this paper. It's simply assumed that the available DERs have enough power to supply the requested load in this paper. A strategy is applied in the step of initialization and position update in MDFA which will be introduced in Sect. 3 to solve the equality constraint. There are two scenarios including the sum of DERs' output is higher than load at time t and the sum of DERs' output is lower than load at time t. It's assumed that the sum of output is higher than the load, and the strategy can be shown as follows:

(1) Initialization: randomly initialize the value of DERs' output and ensure that they are within the output limits.

(2) Calculation: calculate the value of unbalance between the sum of the DERs' output and the load and define it as Delta.

(3) Check whether Delta is less than the precision K. Where K is the value of acceptable unbalance between the sum of the DERs' output and the load. If yes, end this process. Otherwise, randomly choose a DER to increase its output by Delta. Note that the chosen DER can't be chosen again in the late progress.

(4) Check whether the chosen DER's output is within the output range. If positive, end this process. Otherwise, adjust its output to its upper limit and return to step 2.

Since the available DERs have enough power to supply the requested load, a feasible solution must be generated by the above strategy. Figure 1 shows the flowchart of the strategy for solving equality constraints. If the sum of DERs' output is lower than load, and vice versa.
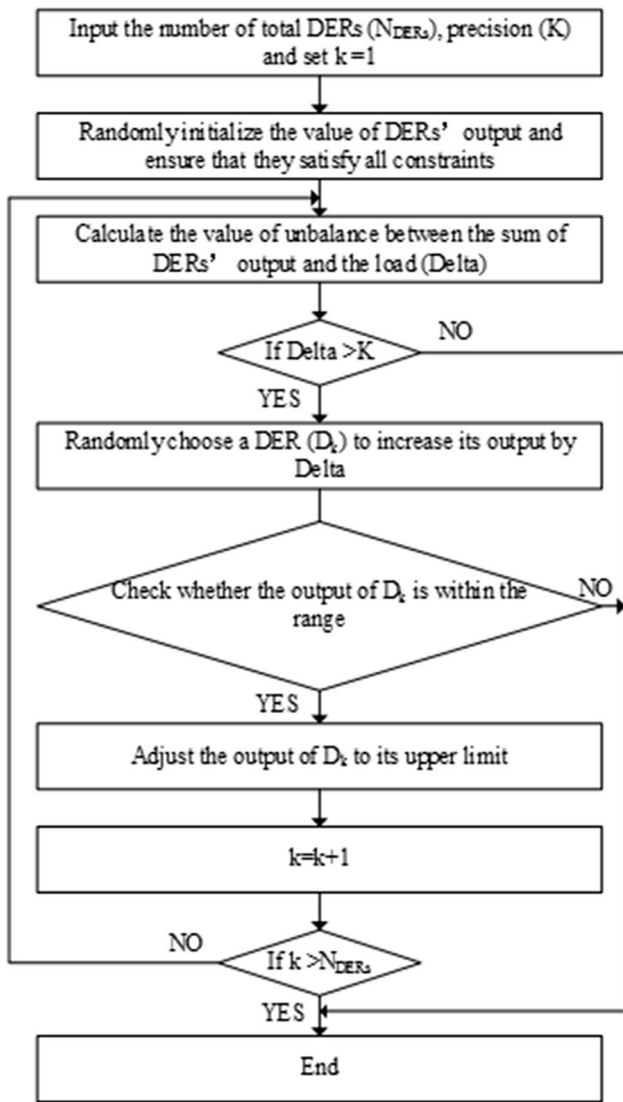
Fig. 1 The flowchart of the strategy for solving equality constraints

3 Proposed Methodology

3.1 Firefly Algorithm

Nowadays, FA becomes one of the most popular population-based algorithms [27]. The algorithm simulates the social behavior of the flashing features of fireflies that the fireflies will move to brighter fireflies and eventually find the optimal global position. In FA, each firefly represents a potentially feasible solution to the optimal problem. The following steps should be employed for implementing FA:

Initialization: first, N initial fireflies are randomly generated in the search space. Each firefly represents a vector of D dimension where the size of D is determined by the number

of decision variables. In the previous paper, the structure of each firefly can be defined as the following:

$$X_i = [X_{i,1}, X_{i,1}, \ldots, X_{i,D}]; i = 1, 2, \ldots N \qquad (8)$$

Apparently, the vector dimension of each firefly is $1 \times D$ in original FA.

Fitness evaluation: calculate the value of the objective function of fitness for each firefly using (2).

Position update-The ith firefly updates its position by moving closer to the jth firefly, which is brighter than it. The way of the fireflies update their position can be written as [34]:

$$X_{i,firefly}^{k+1} = \begin{cases} X_{i,firefly}^{k} + \beta^k(X_{j,firefly}^{k} - X_{i,firefly}^{k}) + \alpha^k(rand(.)_{1 \times D} \\ -0.5) \qquad if \ OF(X_{i,firefly}^{k}) > OF(X_{j,firefly}^{k}) \\ X_{i,firefly}^{k} \qquad otherwise \end{cases} \qquad (9)$$

where α is the random search factor, and its value is in the range of 0–1. And *rand* represents a random number generated from a uniformly distributed selected in the range [0, 1]. β is the attractiveness of a firefly which can be described as the following:

$$\beta = \beta_{min} + (\beta_{max} - \beta_{min}) \cdot e^{-\gamma r^2} \qquad (10)$$

where β_{min} and β_{max} are set to 0.2 and 1, respectively. γ is the light absorption coefficient, and it is a fixed parameter. It will affect the brightness of fireflies and commonly set to 1. r is the Cartesian distance between the mth and nth firefly which can be defined as:

$$r_{m,n}^{k} = \sqrt{\sum_{i=1}^{N} (X_{i,m}^{k} - X_{i,n}^{k})^2}; m, n = 1, 2, \cdots N \qquad (11)$$

Fitness evaluation of new firefly: the objective function value of every new firefly should be calculated by using (2).

Offspring generation: if the brightness of the new fireflies generated by the position update step is higher than that of the current firefly, then it replaces the current firefly and becomes a new member of the population. The way of offspring generation can be described as:

$$X_{i,firefly}^{k+1} = \begin{cases} X_{i,firefly}^{k+1} \qquad if \ OF(X_{i,firefly}^{k}) \geq OF(X_{i,firefly}^{k+1}) \\ X_{i,firefly}^{k} \qquad otherwise \end{cases} \qquad (12)$$

Termination criterion-Check if the conditions for stopping the iteration have been met. In this paper, once the number of iterations reaches the maximum setting, the program stops and outputs the best solution. If not, turn to the step of fitness evaluation and repeat.

3.2 Multidimensional Firefly Algorithm

The FA is widely applied in various aspects of the power system due to its small number of parameters need for tuning and simple concept. In FA, its population diversity makes a great contribution to improving computational efficiency and convergence rate. FA is proved to its superiority compared with other algorithms in solving the problem of economic dispatch in a single hour. In this way, the dimension of a firefly is $1 \times D$ or $D \times 1$. However, it may convergence to an improper position or be caught in premature convergence in complex nonlinear problems, especially in solving ED problems of day-ahead scheduling. Therefore, MDFA is proposed to solve optimal day-ahead scheduling problems whose dimension is $D \times E$ while satisfying various constraints. In the following few subsections, the structure of MDFA and the control parameters are discussed.

3.2.1 The Structure of MDFA

In MDFA, each firefly with D rows and E columns represents feasible solutions in the search space. We apply it to solve the ED problems of day-ahead scheduling in the power system. In a firefly, the data of each column represent the feasible solution of a single hour. The columns represent the total hours. The rows are determined by the number of participant decision variables. Thus, the vector corresponding to the i-th firefly can be expressed as following:

$$X_i = \begin{bmatrix} X_1^1 & X_1^2 & \cdots & X_1^E \\ \vdots & \vdots & \ddots & \vdots \\ X_j^1 & X_j^2 & \cdots & X_j^E \\ \vdots & \vdots & \ddots & \vdots \\ X_D^1 & X_D^2 & \cdots & X_D^E \end{bmatrix}; i = 1, 2 \ldots N \qquad (13)$$

3.2.2 Dynamic Adjustment of α and β

Replacing random sequences with chaotic sequences has been demonstrated to improve the performance of the metaheuristic algorithm [35]. In [36], variants of the basic firefly algorithm with chaotic maps have been presented. The random movement factor α is the key to balance between exploitation and exploration of FA. Finding a specific strategy to determine the value of α can produce different effects on algorithms related to FA. Therefore, the manner to decide α has a significant impact on the performance of MDFA. In this paper, the variation of the random movement factor α is found to provide better performance particularly in MDFA, and it can be defined as:

$$\alpha^{k+1} = \alpha^k * (0.2/iterMax)^{\wedge}(1/iterMax) \qquad (14)$$
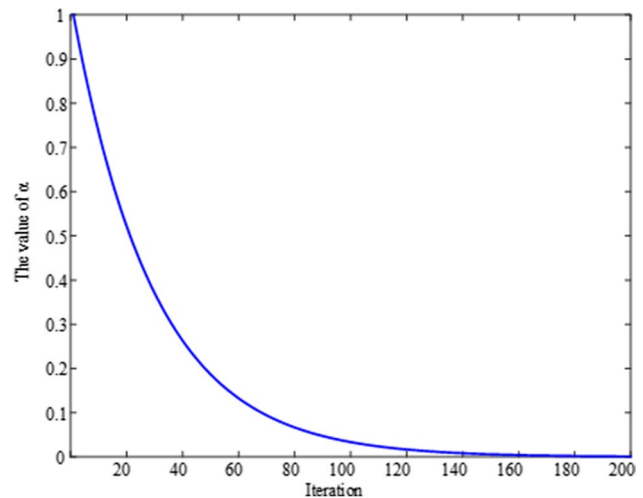
where *iterMax* is the maximum number of iterations. Fig-

Fig. 2 The trajectories of α in MDFA

ure 2 shows the trajectories of α in MDFA.

And for the attractiveness β, it plays a vital role in the performance of MDFA. As the Cartesian distance r increases, the value of β stays too long at the beginning of β_{min}, and after reaching a certain number of iterations it quickly increases to β_{max}. This will have a negative impact on optimization. An improved strategy is proposed to solve it [26]. In [37], the formulation of the attractiveness has been proposed to improve the partially attracted firefly algorithm (PaFA). However, when we solve the high dimensional and broad search boundaries optimization problems, the value of the Cartesian distance between the fireflies is too large to avoid making the attractiveness stays unchanged at β_{min}. Furthermore, the value of attractiveness will increase to 1 very quickly when the difference between fireflies gradually close. In this case, the improved strategy in [26] is not applicable to MDFA, and the diversity of different locations can't be adequately reflected so that the firefly can't

produce different attractiveness. Thus, a novel strategy for β is urgently need to be raised so that the superiority of the

Algorithm 1: The proposed MDFA
1. Generate an initial population using (13) with the proposed strategy handle equality constraint and set k=0.
2. Define initial value of α and γ.
3. Evaluate the fitness values $OF(X_i)$ of all N initial fireflies by using (2), i=1,2, …N.
4. **while** k < iterMax **do**
5.   **for** i = 1 to N **do**
6.   **for** j = 1 to N **do**
7.   **if** $OF(X_j^k) < OF(X_i^k)$ **then**
8.   Update position of X_i^k using the formula in (9)
9.   Using the improved strategy for MDFA
10.   Evaluate the fitness values of $X_i^{move,k}$
11.   **if** $OF(X_i^{move,k}) < OF(X_i^k)$ **then**
12. $X_i^{k+1} = X_i^{move,k}$
13.   **Else**
14. $X_i^{k+1} = X_i^k$
15.   **end if**
16.   **end if**
17.   **end for**
18.   **end for**
19. Rank the fireflies and find the current best X_{best}
20. Update α and β by using (14) and (15)
21. k=k+1
22. **end while**
23. Output the Optimum solution X_{best}

population with its search experience and fast convergence speed can be utilized. In this paper, an improved calculation
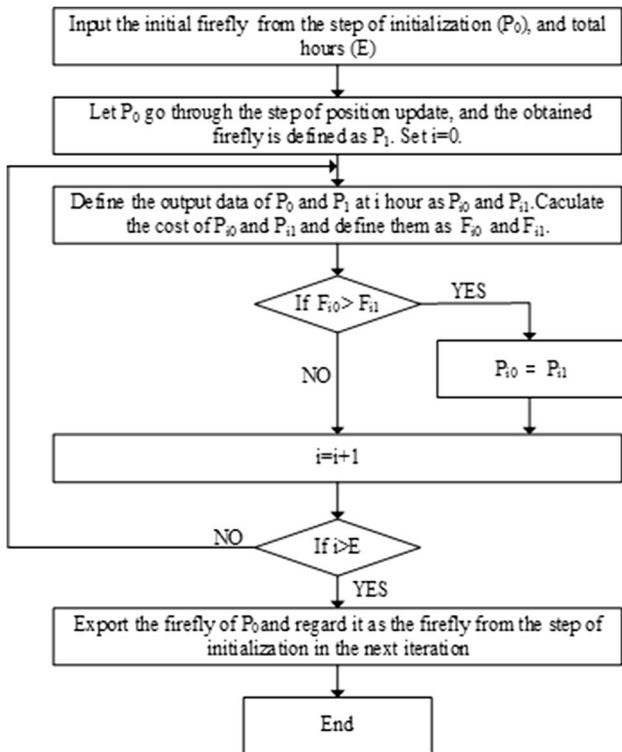
strategy of the attractiveness is proposed to improve MDFA which can be described as:

$$\beta = \beta_{min} + \frac{\beta_{max}}{2 + 2r} + \frac{(\beta_{max} - \beta_{min}) \cdot e^{-\gamma r^2}}{2} \tag{15}$$

where r is the Cartesian distance which has been mentioned.

According to the above formulation, the attractiveness β is equal to β_{max} at $r = 0$ and the attractiveness β decreases with the distance r increases. From this perspective, the proposed strategy is consistent with the characteristic of FA.
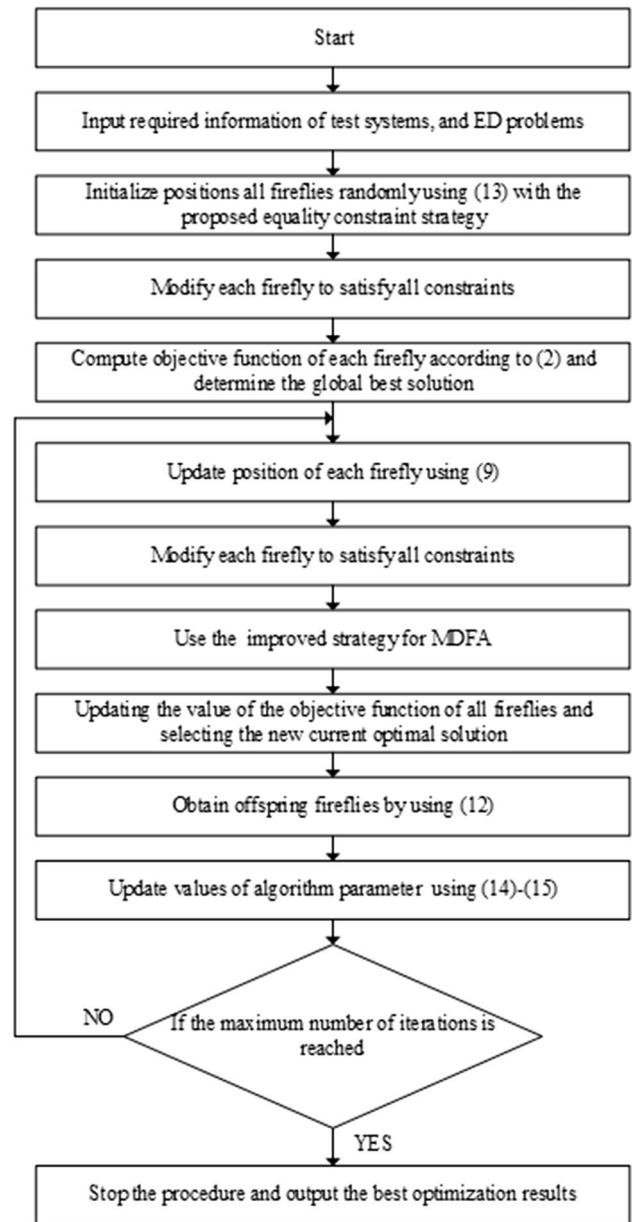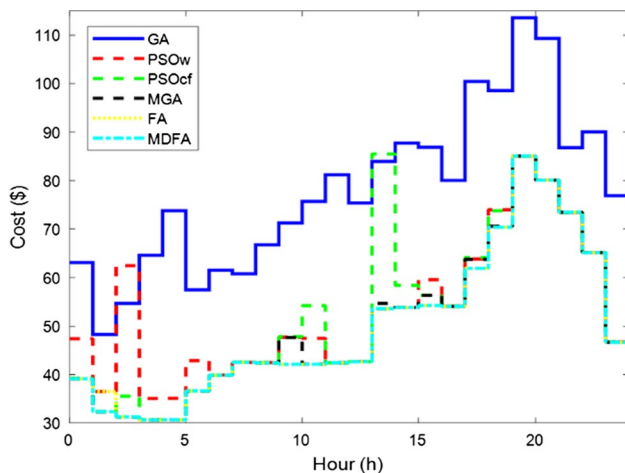

Fig. 3 The strategy for MDFA

Fig. 4 The flow chart of the MDFA algorithm

Table 1 Results obtained by MDFA

| Optimal power scheduling of DERs provided by MDFA (kW)					
Hour	WP1	WP2	WP3	PV1	PV2	CHP
0	357	685	429	0	0	0
1	617	708	0	0	0	0
2	566	697	0	0	0	0
3	532	697	0	0	0	0
4	661	568	0	0	0	0
5	482	671	168	0	0	0
6	486	671	352	0	0	0
7	442	689	532	0	0	0
8	432	727	498	0	0	0
9	406	733	504	0	0	0
10	456	679	508	0	0	0
11	627	659	366	0	0	0
12	658	635	373	0	0	0
13	678	557	196	140	68	0
14	696	653	74	134	85	0
15	707	658	23	121	131	0
16	692	656	138	96	94	0
17	714	660	381	0	0	165
18	721	658	613	0	0	222
19	645	660	652	0	0	425
20	675	662	706	0	0	339
21	680	661	744	0	0	242
22	690	643	695	0	0	146
23	518	674	711	0	0	0

Fig. 5 The value of the generation cost at each hour of the day obtained by the investigated algorithms

Table 2 Comparison of results by mentioned methods

Method	Total generation cost in 24 h ($)	Total computation time (s)
GA	1868.5	48.72
PSO$_w$	1273.8	29.76
PSO$_{cf}$	1259.6	31.2
MGA	1203.8	49.44
FA	1197.3	37.92
MDFA	1193.1	9.89

3.2.3 The Improved Strategy for MDFA

From the previous statement, each firefly has E solutions during total E hours in MDFA. It's clear that there are plenty of changes for each solution in a firefly after the step of position update. In this way, the cost of each solution may be better or worse, and the total cost of solutions in a firefly is difficult to control, which limits immensely the efficiency of MDFA. Thus, a control strategy is proposed to modify
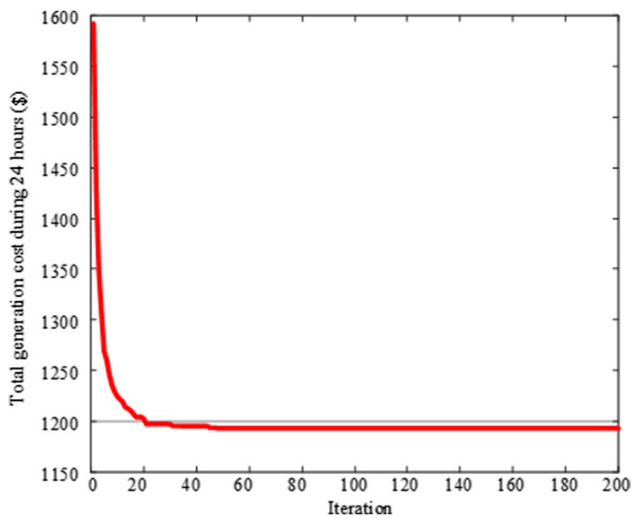
Fig. 6 Convergence process of MDFA for solving the scheduling problem

the firefly gained by the step of position update to avoid the defect during the process of evolution for the firefly as following:

(1) The firefly is obtained from the step of initialization and it is defined as P_0. Let this firefly go through the step of position update, and the obtained firefly is defined as P_1. They both have the output data of the generator in 24 h. Set i = 0, and set its maximum value to 23 in this paper.

(2) The output data of P_0 and P_1 at i-th hour is defined as P_{i0} and P_{i1}. Calculate the cost of P_{i0} and P_{i1}, and their costs are defined as F_{i0} and F_{i1}.

(3) Check whether F_{i0} is larger than F_{i1}. If yes, modify P_{i0} to be consistent with P_{i1}. Otherwise, go to next step.

(4) Increase i by one, and check whether i reaches the maximum value. If yes, output the firefly of P_0 as the firefly.

obtained from the step of position update. And it's also regarded as the firefly from the step of initialization in the next iteration in MDFA. Otherwise, repeat steps 3 and 4.

Figure 3 shows the strategy for a firefly in MDFA. The above strategy is executed to obtain new solutions which represented by a firefly. There are N fireflies are implemented by the strategy.

The proposed control strategy will help MDFA escape from the local optimum and increase the convergence speed, which will be confirmed in Sect. 4.

So far, a new improved version of FA called MDFA, which is shown in Algorithm 1, has been proposed for enlarging exploitation area and accelerating convergence speed. The MDFA has following advantages. First, it ensures

that information is shared among the population, whatever for large-scale or high dimension optimization problem. So that the search experience of the population can be efficiently used. Second, the global search ability of the algorithm is enhanced. It significantly slows down the increased speed of the value of attractiveness, which improves the ability to escape from the local optimum. Third, it enlarges the exploitation area and accelerates convergence speed. Figure 4 shows the flowchart of the MDFA method.

4 Simulation Results and Discussion

In this section, in order to show the performance of the proposed algorithm, the MDFA is applied to solve the formulated problem in the microgrid. All data has been adopted from [20]. In this system, we consider total six decision variables, namely wind plant 1 (WP1), wind plant 2 (WP2), wind plant 3 (WP3), photovoltaic panel 1 (PV1), photovoltaic panel 2 (PV2) and a CHP in our test system. In addition, the 24-h load and the maximum power output for the WPs, PVs and CHP are shown in 3 in Appendix. The minimum power output of the WP, PV and.

CHP during 24 h are zero in this paper. We also assume that the rated capacity of wind plants, PV panels and CHP are 750 kW, 200 kW and 1000 kW, respectively. The cost coefficients of the DERs are also presented in Table 4 in Appendix. Based on the assumption that the available DERs have enough ability to supply the requested power at each hour, the proposed algorithm tries to deliver the required power during 24 h at minimum cost.

In order to effectively test that the proposed algorithm has an excellent performance in dealing with optimization problems, the optimization solutions obtained from the MDFA will be compared with the optimization results provided by five algorithms including genetic algorithm (GA), FA, MGA, particle swarm optimization with constriction factor (PSO$_{cf}$), and particle swarm optimization with inertia weight (PSO$_w$), which have been proven to be superior in solving optimization problems. In this paper, the settings of the algorithm parameters mentioned above except MDFA and FA are in [20]. The settings parameters of MDFA and FA are as following:

FA: N = 50, *iterMax* = 200, α = 1 and γ = 1. It's noted that FA is separately simulated at each hour, and the number of FA needs to be calculated is 24 in this paper, respectively.

MDFA: N = 50, *iterMax* = 200, α = 1 and γ = 1.

The simulations are carried out on MATLAB (R2019a) environment using a desktop machine, which CPU is Intel Core (TM) i7 processor with 3.6 G-Hz clock frequency and 8 GB of RAM.

Table 1 shows the day-ahead scheduling of sources obtained by MDFA. It can be seen more intuitively that the

power balance constraint is fully satisfied owing to the application of the proposed strategy to deal with equality constraints in MDFA. The comparison of the scheduling power obtained by GA, MGA, PSO_{cf}, PSO_w, FA and MDFA at each hour of the day is shown in Fig. 5. From Fig. 5, the cost of results provided by FA and MDFA is same or less than the cost of solutions gained by other mentioned algorithms at most hours in a day, and both FA and MDFA produce more accurate solutions than other investigated algorithms. But FA is trapped into local convergence at hour 1. The results of day-ahead scheduling obtained by FA, GA, MGA, PSO_{cf} and PSO_w are shown in Tables 5, 6, 7, 8, 9 in Appendix. In this paper, in order to ensure fairness for comparing the performance of all algorithms, the computing time for the mentioned algorithms is the total time to obtain the optimal results of six generators during 24 h. From [20], the average computing time required to solve scheduling problems at each hour for GA, MGA, PSO_{cf} and PSO_w are 2.03, 2.06, 1.30, and 1.24 s, respectively. And the average computing time for FA at each hour is 1.58 s. The computing time for the mentioned algorithms has to been converted into 24 h, and the total computational time of GA, MGA, PSO_{cf}, PSO_w, and FA is 48.72, 49.44, 31.2, 29.76 and 37.92 s.

Since the MDFA only needs to be calculated once to obtain the output results of six generators at 24 h, the total computing time for MDFA is 9.89 s during 24 h, which is less than other methods. The comparison of results quality gained by MDFA with other investigated algorithms is and listed in Table 2. From Table 2, if the summation of the costs (24 h) is calculated, MDFA, FA, MGA, PSO_{cf}, PSO_w and GA reaches to 1193.1, 1197.3, 1203.8, 1259.6, 1273.8, and 1868.5 \$.

As the mentioned algorithms except MDFA is hourly calculated, we don't show their convergence characteristics anymore. The convergence characteristic of the MDFA that providing the best solutions in this paper is shown in Fig. 6. It can be clearly seen that the convergence curve of the MDFA appears to be smooth. And the MDFA converges fast to the optimal results whose number of iterations is 47. It indicates that the proposed improved algorithm enhances the global search capability of the original algorithm and is able to explore previously unexplored areas to provide better solutions. In combination with Table 2, it is evident that the generation cost provided by the MDFA are not only better than other methods, but also have higher stability than other methods. From the point of view that the MDFA regards the output of generations during 24 h as a whole, a firefly in MDFA is always a feasible solution meeting all constrains when we solve the solving the scheduling problem considering constraints such as ramp rate limits and other constraints. It shows that MDFA is fast, stable, and has ability to provide optimum solution and satisfy all constraints.

5 Conclusion

In this paper, an improved algorithm called MDFA is presented and its superiority is demonstrated by optimally distributing the required power among DERs in a microgrid. A diminishing strategy was brought into MDFA for the trajectory of the random movement factor (α). And the novel attractiveness formula is introduced instead of initial formula for β, which enhances search area of the MDFA. Besides, the strategy of solving equality constraint, and the strategy to improve efficiency of MDFA are applied in this paper. Through these strategies, individuals are better than that in traditional FA and their search information can be fully utilized which ensure information sharing among the groups. Simulated results reveal that MDFA is more powerful than FA, GA, MGA, PSO_{cf} and PSO_w. MDFA not only finds the solutions with minimal generation cost, but also solves the scheduling problem at quite quick convergence speed. The proposed algorithm can be much more desirable than conventional algorithms in solving complex and nonlinear optimization problems related to day-ahead scheduling of power system.

Appendix

The details data of DERs, load, and results obtained by other investigated algorithms.

See Tables 3, 4, 5, 6, 7, 8, 9

Table 3 The details data of DERs and demand

Hour	Maximum available power of DERs at each hour (kW)						Demand (kw)
	WP1	WP2	WP3	PV1	PV2	CHP	
0	661	685	429	0	0	1000	1471
1	698	708	442	0	0	1000	1325
2	740	697	220	0	0	1000	1263
3	721	697	39	0	0	1000	1229
4	666	568	22	0	0	1000	1229
5	557	671	168	0	0	1000	1321
6	668	671	352	0	15	1000	1509
7	666	689	532	13	71	1000	1663
8	718	727	498	83	101	1000	1657
9	710	733	504	111	124	1000	1643
10	718	679	508	131	144	1000	1643
11	714	659	366	144	160	1000	1652
12	707	635	373	147	165	1000	1666
13	678	557	196	140	160	1000	1639
14	696	653	74	134	143	1000	1642
15	707	658	23	121	131	1000	1640
16	692	656	138	96	108	1000	1676
17	714	660	381	64	87	1000	1920
18	724	658	613	18	49	1000	2214
19	645	660	652	0	0	1000	2382
20	675	662	706	0	0	1000	2382
21	680	661	744	0	0	1000	2327
22	690	643	695	0	0	1000	2174
23	693	674	711	0	0	1000	1903

Table 4 The cost coefficients of DERs

Plant	a	b	c
WP1	0.0027	17.83	4.46
WP2	0.0028	17.54	4.45
WP3	0.0026	17.23	4.44
PV1	0.0055	29.3	4.45
PV2	0.0055	29.58	4.46
CHP	0.0083	75.73	5.21

Table 5 Optimal results obtained by FA

Optimal power scheduling of DERs gained by FA (kW)					
Hour	WP1	WP2	WP3	PV1	PV2	CHP
0	357	685	429	0	0	0
1	175	708	442	0	0	0
2	566	697	0	0	0	0
3	532	697	0	0	0	0
4	661	568	0	0	0	0
5	482	671	168	0	0	0
6	486	671	352	0	0	0
7	442	689	532	0	0	0
8	432	727	498	0	0	0
9	406	733	504	0	0	0
10	456	679	508	0	0	0
11	627	659	366	0	0	0
12	658	635	373	0	0	0
13	678	557	196	140	68	0
14	696	653	74	134	85	0
15	707	658	23	121	131	0
16	692	656	138	96	94	0
17	714	660	381	0	0	165
18	721	658	613	0	0	222
19	645	660	652	0	0	425
20	675	662	706	0	0	339
21	680	661	744	0	0	242
22	690	643	695	0	0	146
23	518	674	711	0	0	0

Table 6 Optimal results obtained by GA

Optimal power scheduling of DERs obtained by GA (kW)					
Hour	WP1	WP2	WP3	PV1	PV2	CHP
0	357	685	429	0	0	0
1	617	708	0	0	0	0
2	566	697	0	0	0	0
3	532	697	0	0	0	0
4	661	568	0	0	0	0
5	482	671	168	0	0	0
6	486	671	352	0	0	0
7	442	689	532	0	0	0
8	432	727	498	0	0	0
9	406	733	504	0	0	0
10	456	679	508	0	0	0
11	627	659	366	0	0	0
12	658	635	373	0	0	0
13	678	557	196	140	68	0
14	696	653	74	134	85	0
15	707	658	23	121	131	0
16	692	656	138	96	94	0
17	714	660	381	0	0	165
18	721	658	613	0	0	222
19	645	660	652	0	0	425
20	675	662	706	0	0	339
21	680	661	744	0	0	242
22	690	643	695	0	0	146
23	518	674	711	0	0	0

Table 7 Optimal results obtained by MGA

Hour	WP1	WP2	WP3	PV1	PV2	CHP
0	357	685	429	0	0	0
1	617	708	0	0	0	0
2	566	697	0	0	0	0
3	532	697	0	0	0	0
4	661	568	0	0	0	0
5	482	671	168	0	0	0
6	486	671	352	0	0	0
7	442	689	532	0	0	0
8	432	727	498	0	0	0
9	312	733	504	94	0	0
10	474	661	508	0	0	0
11	714	659	279	0	0	0
12	658	635	373	0	0	0
13	586	557	196	140	160	0
14	696	653	74	134	85	0
15	707	658	0	121	131	23
16	692	656	138	96	94	0
17	710	660	381	61	0	108
18	717	658	613	0	0	226
19	645	660	652	0	0	425
20	675	662	706	0	0	339
21	680	661	744	0	0	242
22	690	643	695	0	0	146
23	518	674	711	0	0	0

Optimal power scheduling of DERs obtained by MGA (kW)

Table 8 Optimal results obtained by PSO$_{cf}$

Optimal power scheduling of DERs obtained by PSO$_{cf}$(kW)

Hour	WP1	WP2	WP3	PV1	PV2	CHP
0	655	387	429	0	0	0
1	617	708	0	0	0	0
2	346	697	220	0	0	0
3	719	510	0	0	0	0
4	661	568	0	0	0	0
5	557	596	168	0	0	0
6	486	671	352	0	0	0
7	652	479	532	0	0	0
8	708	451	498	0	0	0
9	299	733	504	107	0	0
10	202	662	508	131	140	0
11	707	579	366	0	0	0
12	658	635	373	0	0	0
13	678	0	196	140	0	625
14	696	653	0	134	0	159
15	707	658	0	121	131	23
16	692	656	138	82	108	0
17	711	660	381	64	84	20
18	724	658	613	18	0	201
19	645	660	652	0	0	425
20	675	662	706	0	0	339
21	680	661	744	0	0	242
22	690	643	695	0	0	146
23	673	654	576	0	0	0

Table 9 Optimal results obtained by PSO$_w$

	Optimal power scheduling of DERs obtained by PSO$_w$(kW)					
Hour	WP1	WP2	WP3	PV1	PV2	CHP
0	661	685	0	0	0	125
1	175	708	442	0	0	0
2	740	0	0	0	0	523
3	493	697	39	0	0	0
4	657	550	22	0	0	0
5	557	671	0	0	0	93
6	659	498	352	0	0	0
7	442	689	532	0	0	0
8	707	452	498	0	0	0
9	710	340	504	89	0	0
10	703	363	508	69	0	0
11	706	659	287	0	0	0
12	701	592	373	0	0	0
13	678	557	196	140	68	0
14	696	653	74	134	85	0
15	707	658	23	120	129	3
16	692	656	138	95	95	0
17	714	660	381	64	87	14
18	721	658	613	18	0	204
19	645	660	652	0	0	425
20	675	662	706	0	0	339
21	680	661	744	0	0	242
22	690	643	695	0	0	146
23	693	499	711	0	0	0

References

1. Wang A, Liu W (2020) Distributed incremental cost consensus-based optimization algorithms for economic dispatch in a microgrid. IEEE Access 8:12933–12941
2. Siddiqui IF, Lee SU, Abbas A, Bashir AK (2017) Optimizing lifespan and energy consumption by smart meters in green-cloud-based smart grids. IEEE Access 5:20934–20945
3. Muhanji SO, Muzhikyan A, Farid AM (2018) Distributed control for distributed energy resources: long-term challenges and lessons learned. IEEE Access 6:32737–32753
4. Liu W (2019) Modeling ransomware spreading by a dynamic node-level method. IEEE Access 7:142224–142232
5. Li P, Hu J (2018) An ADMM based distributed finite-time algorithm for economic dispatch problems. IEEE Access 6:30969–30976
6. Modiri-Delshad M, Aghay Kaboli SH, Taslimi-Renani E, Rahim NA (2016) Backtracking search algorithm for solving economic dispatch problems with valve-point effects and multiple fuel options. Energy 116:637–649
7. Chen J, Yang X, Zhu L, Zhang M, Li Z (2013) Microgrid multi-objective economic dispatch optimization. Proc CSEE 33(19):57–66
8. Kaboli SHA, Alqallaf AK (2019) Solving non-convex economic load dispatch problem via artificial cooperative search algorithm. Expert Syst Appl 128:14–27
9. Liu X-K, Wang Y-W, Yan H, Wang X, Hu X (2017) Hybrid consensus-based algorithm for distributed economic dispatch problem. IFAC PapersOnLine 50(1):177–182
10. Khan I, Xu Y, Sun H, Bhattacharjee V (2018) Distributed optimal reactive power control of power systems. IEEE Access 6:7100–7111
11. Cherukuri A, Cortes J (2016) Initialization-free distributed coordination for economic dispatch under varying loads and generator commitment. Automatica 74:183–193
12. Zhao C, He J, Cheng P, Chen J (2017) Consensus-based energy management in smart grid with transmission losses and directed communication. IEEE Trans Smart Grid 8(5):2049–2061
13. Guo F, Wen C, Mao J, Song Y-D (2016) Distributed economic dispatch for smart grids with random wind power. IEEE Trans Smart Grid 7(3):1572–1583
14. Loukarakis E, Dent CJ, Bialek JW (2016) Decentralized multi-period economic dispatch for real-time flexible demand management. IEEE Trans Power Syst 31(1):672–684
15. Li C, Yu X, Yu W, Huang T, Liu Z-W (2016) Distributed event-triggered scheme for economic dispatch in smart grids. IEEE Trans Ind Inf 12(5):1775–1785
16. Liu J, Li J (2015) A bi-level energy-saving dispatch in smart grid considering interaction between generation and load. IEEE Trans Smart Grid 6(3):1443–1452

17. Gaing Z-L (2003) Particle swarm optimization to solving the economic dispatch considering the generator constraints. IEEE Trans Power Syst 18(3):1187–1195

18. Noman N, Iba H (2008) Differential evolution for economic load dispatch problems. Electr Power Syst Res 78(8):1322–1331

19. Azaza M, Wallin F (2017) Multi objective particle swarm optimization of hybrid micro-grid system: a case study in Sweden. Energy 123:108–118

20. Askarzadeh A (2018) A Memory-based genetic algorithm for optimization of power generation in a microgrid. IEEE Trans Sustain Energy 9(3):1081–1089

21. Crisostomi E et al (2014) Plug-and-play distributed algorithms for optimized power generation in a microgrid. IEEE Trans Smart Grid 5(4):2145–2154

22. Maulik A, Das D (2017) Optimal operation of microgrid using four different optimization techniques. Sustain Energy Technol Assess 21:100–120

23. Amrollahi MH, Bathaee SMT (2017) Techno-economic optimization of hybrid photovoltaic/wind generation together with energy storage system in a stand-alone micro-grid subjected to demand response. Appl Energy 202:66–77

24. Zhang J, Wu Y, Guo Y, Wang B, Wang H, Liu H (2016) A hybrid harmony search algorithm with differential evolution for day-ahead scheduling problem of a microgrid with consideration of power flow constraints. Appl Energy 183:791–804

25. Elsied M, Oukaour A, Gualous H, Lo Brutto OA (2016) Optimal economic and environment operation of micro-grid power systems. Energy Convers Manag 122:182–194

26. Yang Y, Wei B, Liu H, Zhang Y, Zhao J, Manla E (2018) Chaos firefly algorithm with self-adaptation mutation mechanism for solving large-scale economic dispatch with valve-point effects and multiple fuel options. IEEE Access 6:45907–45922

27. Yang X-S (2008) Nature-inspired metaheuristic algorithms. Luniver press

28. Yang XS (2010) Firefly algorithm, stochastic test functions and design optimisation. Int J Bio Inspir Comput 2(2):78–84

29. Yang X-S, Hosseini SSS, Gandomi AH (2012) Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect. Appl Soft Comput 12(3):1180–1186

30. dos Santos Coelho L, Mariani VC (2013) Improved firefly algorithm approach applied to chiller loading for energy conservation. Energy Build 59:273–278

31. Niknam T, Azizipanah-Abarghooee R, Roosta A (2012) Reserve constrained dynamic economic dispatch: a new fast self-adaptive modified firefly algorithm. IEEE Syst J 6(4):635–646

32. Parisio A, Glielmo L (2011) A mixed integer linear formulation for microgrid economic scheduling. In: The Proceeding of the second IEEE international conference on smart grid communications, Brussels, pp 505–510

33. Caldon R, Patria AR, Turri R (2004) Optimal control of a distribution system with a virtual power plant. In: Proceedings on bulk power system dynamics and control conference, Cortina d'Ampezzo, pp 278–284

34. Baykasoğlu A, Ozsoydan FB (2015) Adaptive firefly algorithm with chaos for mechanical design optimization problems. Appl Soft Comput 36:152–164

35. Caponetto R, Fortuna L, Fazzino S, Xibilia MG (2003) Chaotic sequences to improve the performance of evolutionary algorithms. IEEE Trans Evol Comput 7(3):289–304

36. Gandomi AH et al (2013) Firefly algorithm with chaos. Commun Nonlinear Sci Numer Simul 18(1):89–98

37. Zhou L et al (2019) An accurate partially attracted firefly algorithm. Computing 101(5):477–493

YuDe Yang was born in Guangxi Province, China. He received the M.S. and Ph.D. degrees in electrical engineering from the School of Electrical Engineering, Guangxi University, China in 2004 and 2007, respectively. He is currently an Associate Professor with the Department of Electrical Engineering. His current research interests major in the transient stability of power system and power system optimal operation and control.

JinLian Qiu was born in Fujian Province, China. He is currently pursuing the M.S. degree in electrical engineering with the School of Electrical Engineering, Guangxi University, China. His current research interests include intelligent computation and their applications.

ZhiJun Qin was born in Guangxi Province, China. He received the B.E. and M.S. degrees (Hons.) from the Huazhong University of Science and Technology, Wuhan, China, in 2000 and 2003, respectively. And the Ph.D. degree from The University of Hong Kong, Hong Kong, in 2015, all in electrical engineering. From 2015 to 2016, he was a PostDoctoral Research Fellow with The University of Hong Kong. He was the major algorithm developer and among the key contributors of a decision support software tool entitled System Restoration Navigator, which was developed under the support of Electric Power Research Institute, USA, from 2010 to 2016. He is currently an Associate Professor with Guangxi University. His research interests include power system resilience, renewable energy integration, cyber security of cyber-physical system, optimal power flow, and power system dynamics.