

Research Article

Three-Partition Multistrategy Adaptive Fruit Fly Optimization Algorithm for Microgrid Droop Control

Xinkun Tao,¹ Li Zhang ,¹ Fuzhong Wang,¹ Guangqiang Tian,² and Hongwei Zhang¹

¹School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo, Henan, China

²School of Intelligent Engineering, Huanghe Jiaotong University, Zhengzhou, Henan, China

Correspondence should be addressed to Li Zhang; dqzhangli@hpu.edu.cn

Received 22 October 2021; Accepted 20 December 2021; Published 31 January 2022

Academic Editor: Pawan Sharma

Copyright © 2022 Xinkun Tao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Summary. When the microgrid topology changes, the power output of the inverter cannot be adaptively adjusted by traditional droop control, and the dynamic performance and steady-state accuracy of the inverter are affected. To solve this problem, a three-partition multistrategy adaptive fruit fly optimization algorithm (MSAD-FOA) is proposed, which performs a real-time optimization of the PI parameters to realize microgrid droop control. The fruit fly population is divided into three regions according to the ranking of the fitness values of the algorithm. Next, the multistrategy model is automatically updated according to the difference in the fruit fly performance in each region. The local fine search in zone I ensures that the population does not degenerate. Zone II pertains to the adaptive adjustment to ensure the diversity and convergence of the algorithm. Zone III guides the fruit flies to accelerate convergence. The effectiveness of the algorithm and feasibility of the proposed control strategy are verified through a theoretical simulation and microgrid droop control simulation. The comparison with other algorithms demonstrates the superiority of the development and exploration ability of the proposed algorithm. The response speed of the inverter is 40 times higher when the proposed control strategy is used, and the steady-state error is reduced by 4.3%.

1. Introduction

Microgrid droop control refers to a double closed-loop control system, which is composed of multiple PI controllers in series and parallel [1]. The control effect of the system varies in cases involving different PI parameters. Because the microgrid is a dynamic system, when the system topology changes, the conventional PI controller cannot adapt to the changes in the system parameters, thereby reducing the response speed of the inverter. The output power, frequency, and voltage of the inverter tend to be out of limit and oscillatory, and the dynamic performance and steady-state accuracy of droop control are reduced. Therefore, it is necessary to adjust the PI parameters online in real time.

In recent years, the research on microgrid has been continuous. Under the premise of considering voltage stability, a microgrid dispatching model for economy and environmental pollution was established by Reddy et al. [2], using multiobjective particle swarm optimization for

optimization, fuzzy logic selection of compromise solutions, thereby improving the economy and stability of the system. The thermoelectric and wind energy integrated energy-saving system was built by Reddy et al. [3], the adaptive differential evolution algorithm was used to optimize the solution, and it makes the operating cost of the microgrid and the emission of pollutants be controlled in the best condition. This kind of literature is devoted to the research of microgrid optimization dispatching, by establishing the economic and environmental pollution model of the microgrid, using intelligent algorithm optimization that reduces operating costs and environmental pollution. However, in addition to considering the economics of microgrid operation, the stability of the microgrid should also be considered. The research and education microgrid control model was established in Momoh et al. [4], which solved the control problems of voltage, reactive power, and frequency. At present, droop control is a popular control strategy of microgrid multimachine parallel operation in the

island mode. The principle is to realize the load power distribution according to the capacity by adjusting the characteristic curve [5]. Many scholars have conducted considerable research on droop control, for example, to address the fluctuation and out of limit tendencies of the frequency and voltage in droop control. Ling et al. [5] attempted to prevent the overlimit frequency resulting from the switching of heavy load or large-capacity DGs by using an inverted-S droop control strategy. The microsource could promptly adjust the frequency in case of large load fluctuations to prevent the values from exceeding the limit. However, the problem of the voltage being out of limit was not considered. A droop control strategy based on self-recovery was proposed by Chai et al. [6]. When the load changed suddenly, the proposed approach could prevent the dynamic response oscillation of the inverter output voltage. However, the response speed of the inverter was low. For complex nonlinear systems, it is difficult to simplify the model and deduce the formula [7]. Therefore, certain researchers introduced an intelligent optimization algorithm in droop control. The method of fuzzy sliding mode droop control was studied by Zhu et al. [8]. The fuzzy control strategy adjusts the parameters for droop control in an online manner. When the load fluctuates, this method can reduce the voltage error and increase the accuracy of the power distribution. Shivam et al. [9] and Taghizadeh et al. [10] used different algorithms to optimize the droop control parameters. The robustness of droop control was enhanced, and the fluctuation of the voltage and frequency was suppressed. Although many scholars have proposed improvement methods, the adaptive ability of droop control strategies in events including microgrid topology changes has not been considered. In practice, a microgrid is a dynamic system, and a microgrid performs not only load switching but also microsource switching. Therefore, wind and light abandonment caused by the uncertainty in the renewable power output and load demand often occurs [11]. Therefore, when the microgrid topology changes, it is necessary to maintain the stability of droop control.

The traditional droop control strategy includes many PI controls. When the microgrid topology changes, the PI parameter must be reset. If the PI parameters are not set, the stable operation of the microgrid may be affected. With the emergence of intelligent algorithms, many scholars have introduced intelligent algorithms to adjust the PI parameters of microgrids. A control algorithm based on GOA optimized PI was proposed by Jumani et al. [12]. The minimum fitness value was used as the optimization objective, the PI parameters were determined, and the problems of voltage and frequency overshoot and total harmonic distortion were solved. A PI automatic gain controller based on the genetic algorithm was proposed by Ismayil et al. [13]. The optimization objectives were the steady-state error, response time, and maximum overshoot/undershoot of the system response; the robustness of PI control was enhanced. A PI inverter controller scheme based on the particle swarm optimization algorithm was proposed by Roslan et al. [14]. The inverter could effectively reduce the harmonics and stabilize the frequency. Moreover, the online PI parameter

optimization strategy of the VSI inverter based on the fruit fly algorithm was designed by Liu et al. [15]. Four PI controls were simultaneously optimized, and the output performance of the VSI inverter was enhanced. Among the abovementioned algorithms, the GOA formula is highly complex and difficult to program, the genetic algorithm is complex and easily falls into the local optima, and the computational stability of the particle swarm optimization algorithm is low [16]. In comparison, the fruit fly algorithm exhibits a high optimization speed, simple structure, low amount of calculation, adjustable parameters, outstanding global search ability [16, 17], and strong ability to solve nonlinear problems. Therefore, many scholars have applied the fruit fly algorithm for the parameter optimization of microgrids.

The fruit fly algorithm exhibits an outstanding global optimization ability in the early stage and can rapidly find the optimal value. However, the search ability in the later stage is inferior, the algorithm easily falls into the local optima, and it exhibits a low optimization accuracy in solving complex problems [18]. These deficiencies were highlighted by Xiong et al. [19]. In terms of population zoning, the sentinel mechanism and the multigroup mechanism were introduced by Chen et al. [20]. The sentinel mechanism consists of greedy selection and Gaussian variation to enhance the convergence speed of the algorithm. The multigroup mechanism divides the fruit fly population into several subgroups to enhance the exploration ability of fruit flies. A double-driven fruit fly algorithm was proposed by Feng et al. [21]. In this framework, fruit fly flight is guided by the concept of multiple repellents and attractants. Adaptive determination of the search radius is performed. The algorithm is used to optimize and enhance the performance of the PID framework. To introduce a novel search mechanism, a fruit fly algorithm for adaptive cloud escape search was proposed by Zhang et al. [22]. The search method considered the number of iterations as the guiding factor to adjust the global and local search. Moreover, a new evolutionary direction intelligent selection mechanism was proposed by Wu et al. [23]. This selection mechanism provides the correct direction for fruit flies, and the convergence speed of the algorithm is increased. An improved fruit fly optimization algorithm (IFOA) was proposed by Yuan et al. [24]. By introducing the inertial weight function to the search step, the search ability of the algorithm was improved. In the abovementioned techniques, the introduction of a new search mechanism can compensate for the defects of the fruit fly algorithm; however, the improved FOA algorithm relies on a single strategy to update the positions of the fruit flies and thus cannot distinguish fruit flies with different performances. In addition, real-time detection of the optimization effect of the fruit fly algorithm is not implemented in the improved methods. Therefore, in the context of the online optimization of droop control PI parameters, the output of the droop control inverter cannot be fed back in time, and the convergence and diversity of the algorithm cannot be adjusted in time, causing the algorithm to perform many invalid calculations.

To enhance the adaptive ability of the inverter of a microgrid droop control system, this paper analyzes the

basic principle of droop control and limitations of the standard fruit fly algorithm. A three-partition multistrategy adaptive fruit fly algorithm for microgrid droop control is proposed. The algorithm can detect and feedback the optimization effect and adjust the PI parameters in real time. When the topology and load change, a rapid response and an accurate output of the inverter are ensured during the operation of the microgrid. Consequently, the response speed of the inverter can be increased, the error can be decreased, and a more stable output can be obtained.

The main contributions are as follows. A novel three-partition multistrategy adaptive fruit fly optimization algorithm is proposed in this paper. It has made great progress in promoting the convergence and diversity of the algorithm. The algorithm is applied to droop control strategy, providing online optimization strategy and theoretical framework. In terms of algorithm: (1) Three-partition and multistrategy modes are proposed, the optimization ability of each fruit fly is fully exploited, and invalid calculations are avoided. (2) An evaluation index for the average individual increment in the fruit fly population is proposed. The effect of the algorithm optimization can be fed back in real time, and the convergence and diversity of the adaptive adjustment algorithm can be ensured. In terms of droop control optimization, (1) the absolute value integral term of the PI error derivation is introduced in the objective function of microgrid online optimization, and the oscillation and deviation of the inverter output power are effectively suppressed. (2) In noninitial optimization, the use of adaptive and optimal delivery strategies shortens the online optimization time.

2. Preliminary Study on FOA

In recent years, optimization technologies have continuously developed, and intelligent optimization algorithms, represented by particle swarm optimization algorithms and genetic algorithms, have provided a strong technical support to solve complex nonlinear problems. However, for the particle swarm optimization and genetic algorithms, it is necessary to update a variety of attributes, the calculation is complex, and the program is difficult to implement. In comparison, the update strategy of the fruit fly algorithm is simple, easy to implement, and requires a small amount of calculation, rendering this algorithm more suitable for online optimization.

The fruit fly algorithm is a new algorithm that imitates the search of food by fruit flies [25]. The flowchart is shown in Figure 1, and the updated formula is as follows:

$$\begin{aligned} x_{i,j}(k+1) &= x_{i,j}(k) + c \cdot L \cdot \text{rand}(), \\ Y_{i,j}(k+1) &= Y_{i,j}(k) + c \cdot L \cdot \text{rand}(), \end{aligned} \quad (1)$$

where L is the optimization step size, c is a nonnegative acceleration constant, rand is a random number generated between 0 and 1, i is the number of fruit flies, j is the dimension, $x_{i,j}(k)$, $y_{i,j}(k)$ represent the global optimal position, and $x_{i,j}(k)$ and $Xgbest_{i,j}^t$, described in the following text, are values with the same meaning.

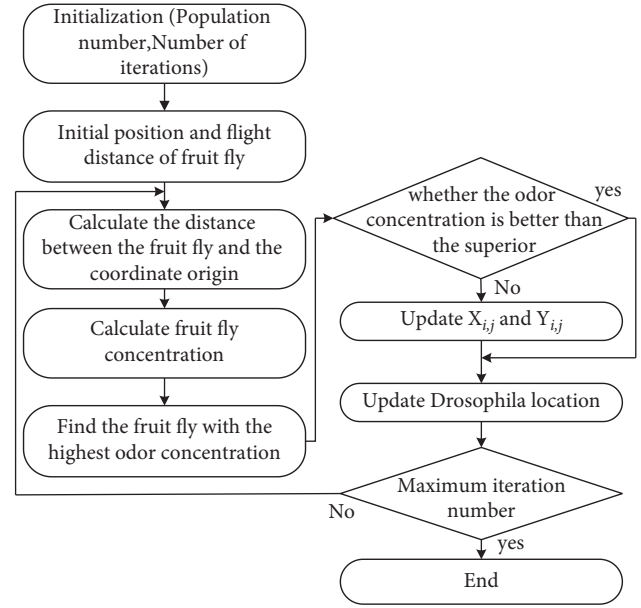


FIGURE 1: Flowchart of standard fruit fly algorithm.

The fruit fly algorithm is applied to the PI parameter optimization of droop control to ensure the stability of droop control (minimum error) and obtain the optimal droop control parameters. Parameter k_p is a proportional adjustment term to ensure the speed of adjustment. An increase in k_p can increase the regulation speed of the inverter, but an extremely large k_p may lead to the instability of the inverter output. Parameter k_i pertains to integral regulation, aimed at eliminating the steady-state error. An increase in k_i can shorten the time required for the inverter to reach steady state; however, if k_i is extremely large, the overshoot may increase. The goal of the microgrid is to ensure that in events involving topology changes, the droop control promptly responds without any overshoot. After the microgrid enters the steady state, the output power, voltage, and frequency do not fluctuate, and the error is small. Four PI controllers are included in droop control. The interactions among the PI controllers [26] form a complex control system. The dynamic performance and the steady-state accuracy of the microgrid system are affected by the value of each PI parameter; therefore, the accuracy of optimization should be ensured. Online optimization should ensure a higher optimization speed.

In the preliminary study, the standard FOA is applied to the online optimization of droop control after many experiments: (1) The FOA leads to a low response speed of the inverter and a large error in the steady state (Figure 2). (2) When many microsources exist, the FOA cannot ensure the normal operation of each microsource, and the output power of the inverter considerably fluctuates. (3) The optimization results of the FOA cannot be improved by adjusting the algorithm step size and number of iterations (Figures 2 and 3). Notably, the accuracy of FOA optimization is not high, the algorithm falls into a local optimum, and the optimal PI parameters cannot be identified to ensure the microgrid performance. (4) The convergence curve of

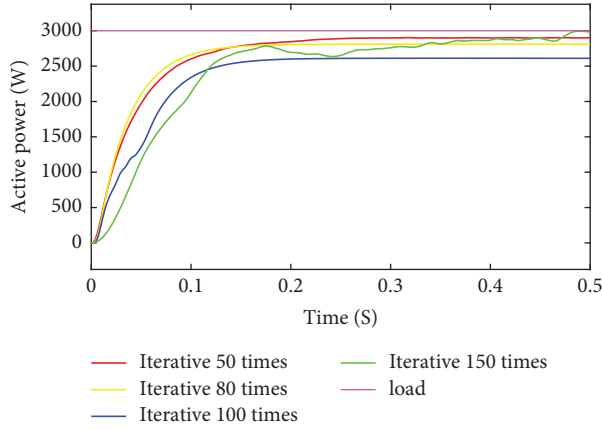


FIGURE 2: Results by changing the number of iterations.

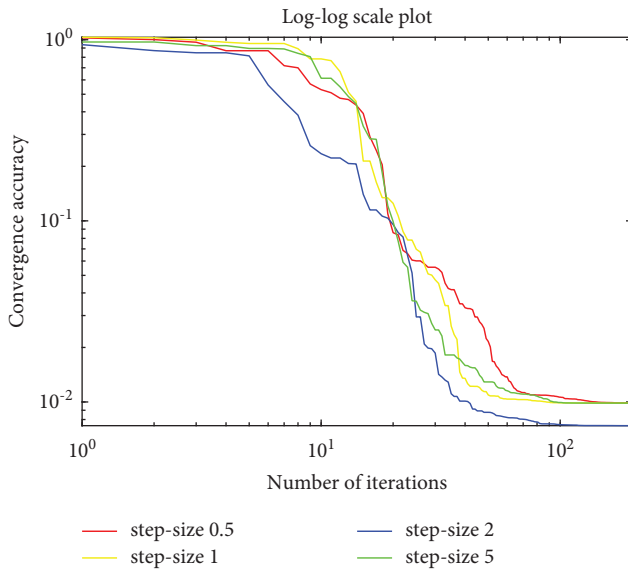


FIGURE 3: Results of changing step size.

the FOA indicates that the algorithm needs at least 80 iterations to ensure a low convergence accuracy. When the number of microsourses increases, the algorithm dimension increases, and cliff convergence occurs in the optimization of the FOA (Figure 4); thus, convergence in each iteration cannot be ensured. The convergence of the FOA cannot be changed by increasing the number of populations because the algorithm performs many invalid calculations, which adversely influence the convergence.

According to this analysis, the standard FOA can be enhanced in terms of the following aspects: (1) Increase the convergence accuracy and speed of the fruit fly algorithm as follows: (i) adopt the updating method of the variable step size; (ii) introduce the flight experience of the optimal individual of the fruit fly population in the updated formula. (2) Enhance the development and exploration ability of the fruit fly algorithm as follows: (i) adopt a new idea of population zoning and use different update strategies for fruit flies with different performances to fully exploit the optimization ability of each fruit fly; (3) the algorithm can

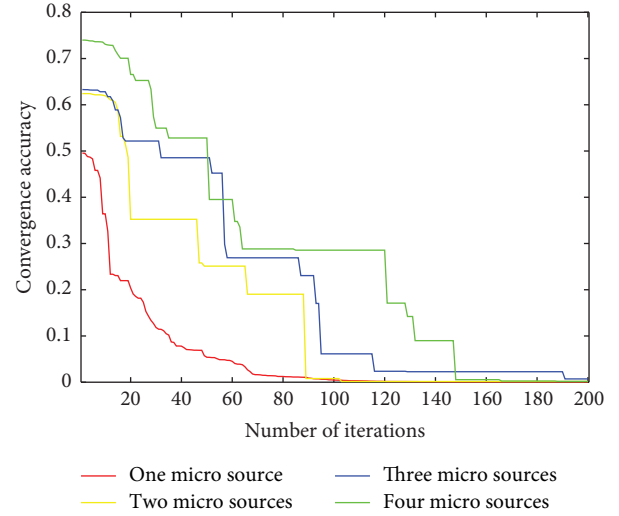


FIGURE 4: Optimization results of adding microsource.

perform detection and feedback optimization in real time, closed-loop control can be formulated, and negative feedback adjustments can be performed in the algorithm in time, thereby avoiding invalid calculations.

The population includes fruit flies with high, average, and low performances. Fruit flies with different performances exhibit different abilities, and it is easy to distinguish superior and inferior fruit flies. However, the existing approaches cannot yield an accurate evaluation of fruit flies with average performances because of the lack of a clear evaluation standard. In practice, fruit flies with average performance account for the majority of the flies, and these fruit flies contain both slightly superior fruit flies and slightly inferior fruit flies. In the process of optimization, these fruit flies serve as a “pillar rock in midstream.” Therefore, in the optimization process, it is necessary to effectively partition the fruit fly population and consider their characteristics. Division schemes for the fruit fly population were proposed by Wang et al. [27] and Wang et al. [28]. Subgroups 1 and 2 adopted the methods of local search and global search, respectively, to enhance the algorithm stability. The experiment was conducted using dichotomy concepts. It was noted that the convergence speed of the algorithm was not increased. This phenomenon occurred because the method considered only an extremely small number of optimal fruit flies. Fruit flies with average and low performances were not distinguished, the phenomenon of fuzzy partition occurred, and the performance of each fruit fly could not be exploited. The strategy of dividing the population into multiple subgroups was proposed by Yang et al. [29] and Zhang et al. [30]. Each subgroup performed a parallel search, and the convergence accuracy was increased. The experiment was conducted using the multipartition method, and it was noted that with the increase in zoning, the number of fruit flies was required to be increased to ensure that a large number of fruit flies were present in each subgroup, leading to a significant increase in the amount of calculation of the algorithm. Therefore, while the partition strategy should ensure that fruit flies with different performances can be

distinguished and avoid the phenomenon of partition ambiguity, the increase in the amount of calculation due to an increase in the number of partitions must also be considered.

Considering the advantages and limitations of zoning in the literature, in this paper, fruit flies with an average performance are placed in the same area. Three partitions are formed based on two partitions. The convergence and diversity of the algorithm are adaptively adjusted by these fruit flies. According to the unique search mechanism of the fruit fly algorithm and problems associated with its optimization, a closed-loop control algorithm that can detect the optimization effect in real time is proposed. The algorithm adaptively adjusts the fruit fly population in the target search and explores and develops the region in an optimal manner. Increasing the population diversity can accelerate population convergence, considering the diversity and convergence of the fruit fly population. This algorithm is named the three-partition multistrategy adaptive fruit fly optimization algorithm (MSAD-FOA).

3. Implementation of MSAD-FOA Algorithm

3.1. Three Divisions of the Fruit Fly Population. This paper considers the optimization minimum as an example. After 5 iterations of the algorithm, three partitions of the population are schematically illustrated in Figure 5. The fruit fly population is sorted according to the individual evaluation index of the fruit flies. A three-partition formula is as follows:

$$\begin{aligned} X_{i,j}^{t+1} &= X_{i,j}^t + \varphi \cdot L \cdot \left(\text{rand}_1(Xpbest_{i,j}^t - X_{i,j}^t) + \text{rand}_2(Xgbest_{i,j}^t - X_{i,j}^t) \right), \\ Y_{i,j}^{t+1} &= Y_{i,j}^t + \varphi \cdot L \cdot \left(\text{rand}_1(Ypbest_{i,j}^t - Y_{i,j}^t) + \text{rand}_2(Ygbest_{i,j}^t - Y_{i,j}^t) \right), \\ \varphi &= \frac{1}{2} \times \left(1 + \cos \left(\pi \times \sqrt{\frac{t-1}{t_{\max}}} \right) \right). \end{aligned} \quad (3)$$

In equation (3), φ is the weighting factor, which decreases with the number of iterations. $Xgbest_{i,j}^t, Ygbest_{i,j}^t$ represent the global fruit fly optimal positions at time t , and $Xpbest_{i,j}^t, Ypbest_{i,j}^t$ represent the optimal positions of the individual fruit flies at time t . This equation shows that for the fruit fly population of zone I, the update direction of the fruit fly position at the next moment is always toward the optimal position.

The fruit flies in zone II exhibit an average performance but reach the fruit fly optimal target after a relatively small number of iterations. In zone II, the superior or inferior fruit

$$\text{region}(f(t)) \begin{cases} I, & \text{rank}(f(t)) < \alpha \cdot n, \\ II, & \alpha \cdot n \leq \text{rank}(f(t)) < \beta \cdot n, \\ III, & \text{rank}(f(t)) \geq \beta \cdot n. \end{cases} \quad (2)$$

In equation (2), α and β are positive numbers between 0 and 1, and n is the number of fruit flies. The fruit fly population is divided into three regions based on the values of α and β , and the number of fruit flies in each region is determined. The values of α and β can be specified only after determining the multistrategy mode and adaptive strategy. Section 3.4 discusses the impact of the values of α and β on the algorithm.

3.2. Determination of Multistrategy Mode. Different renewal strategies should be formulated according to the different performances of fruit flies to enhance the exploration and development ability of the fruit fly populations.

The fruit flies in zone I exhibit the highest performance, suggesting that the fruit fly population in zone I is closest to the optimal target. The elite fruit fly population should be maintained; no population degradation occurs, and the local search capability increases. The concept of the historical individual optimal is introduced in the fruit fly algorithm by referring to the particle swarm optimization formula [31]. The next position of a fruit fly is updated through a combination of the fruit fly global optimum and the fruit fly individual optimum. The updated formula for zone I is as follows:

flies are not distinguished, and thus, the fruit fly population in zone II takes into account multiple styles and convergence. The updated formula is as follows:

$$\begin{aligned} X_{i,j}^{t+1} &= \begin{cases} Xgbest_{i,j}^t + \varphi \cdot L \cdot \text{rand}, & \text{rand}_{i,j} > G, \\ X_{i,j}^t + \varphi \cdot L \cdot \text{rand} \cdot (Xgbest_{i,j}^t - X_{i,j}^t), & \text{rand}_{i,j} \leq G, \end{cases} \\ Y_{i,j}^{t+1} &= \begin{cases} Ygbest_{i,j}^t + \varphi \cdot L \cdot \text{rand}, & \text{rand}_{i,j} > G, \\ Y_{i,j}^t + \varphi \cdot L \cdot \text{rand} \cdot (Ygbest_{i,j}^t - Y_{i,j}^t), & \text{rand}_{i,j} \leq G, \end{cases} \end{aligned} \quad (4)$$

where G is the evaluation index of the population convergence and diversity.

The fruit flies in zone III exhibit a low performance, with no notable contribution to the fruit fly population, and are far from the optimal target. The convergence of fruit flies in zone III should be accelerated. The update formula is as follows:

$$\begin{aligned} X_{i,j}^{t+1} &= X_{i,j}^t + \varphi \cdot L \cdot \text{rand} \cdot (Xg\text{best}_{i,j}^t - X_{i,j}^t), \\ Y_{i,j}^{t+1} &= Y_{i,j}^t + \varphi \cdot L \cdot \text{rand} \cdot (Yg\text{best}_{i,j}^t - Y_{i,j}^t). \end{aligned} \quad (5)$$

3.3. Implementation of the Adaptive Mode. In algorithm, the convergence and diversity contradict each other to a certain extent. However, adaptively adjusting the convergence and diversity according to the environment of the algorithm can enhance the comprehensive performance of the algorithm. Certain fruit flies are randomly selected from zone II to

perform diverse location updating. The remaining fruit flies perform convergent position updating. The selection method is randomized. The first update formula is selected when the random number is larger than G . This update formula is the standard update formula for the fruit fly optimization algorithm, which has a global search capability that can ensure population diversity. The second update formula is selected when the random number is less than G . This update formula is based on the position in the previous instance; the updated orientation ensures that the individual fruit flies fly along the global optimal direction, and the flight distance decreases with the number of iterations. This update method promotes the accelerated convergence of fruit fly individuals. Thus, the adaptive G -value can automatically adjust the convergence and diversity of the algorithm. An evaluation index of the average individual increment (AII) of the fruit fly population is proposed as the basis for adaptive adjustment:

$$\begin{aligned} DI(t) &= \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^{\text{dim}} ((X(t)_i^j - Xp\text{best}_i^j) + (Y(t)_i^j - Yp\text{best}_i^j)), \\ DI(t-1) &= \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^{\text{dim}} ((Y(t-1)_i^j - Yp\text{best}_i^j) + (Y(t-1)_i^j - Yp\text{best}_i^j)), \\ AII &= DI(t) - DI(t-1), \\ G &= \eta \cdot AII. \end{aligned} \quad (6)$$

In equation (6), $DI(t-1)$ is the dimensional difference between the fruit fly position at the previous moment and the historical optimal position, and $DI(t)$ is the dimensional difference between the fruit fly position at the current moment and the historical optimal position. AII is the increment of the difference in the current moment and the previous moment. η is the weighting factor. G is the adaptive adjustment parameter. When the fruit fly updates its position to be near the optimal position, the DI decreases. In contrast, a larger DI . A smaller AII (G) means that the fruit fly population is closer to the optimal position of the fruit fly at the current moment compared with that in the previous moment. In this case, the diversity of the fruit fly population should be increased, thereby enhancing the exploitation ability of the fruit fly population. Similarly, when AII (G) is large, the convergence of the fruit fly population should be accelerated, thereby enhancing the exploratory capabilities of the fruit fly population.

The dimensional difference between the positions at the same moment is considered. The advantages of this method are as follows: the same fruit fly may exhibit low and high performances on different dimensions. However, the comprehensive evaluation of the individual fruit flies is enhanced, and a direct summation can offset the deviation of the dimension extremes to a certain extent, thereby preventing the bias from affecting the overall evaluation of the individual fruit flies.

3.4. Partition Parameters α and β . Zone II can adaptively adjust the convergence and diversity according to the environment of the algorithm; therefore, determining partition parameters α and β is expected to affect the comprehensive performance of the algorithm. Figure 6 shows the variation curve of the AII evaluation index when solving F3 and F5. The value schemes are as follows: Case 1: $\alpha = 0.1$ N, $\beta = 0.9$ N; Case 2: $\alpha = 0.2$ N, $\beta = 0.8$ N; and Case 3: $\alpha = 0.3$ N, $\beta = 0.7$ N.

According to Figure 2, the convergence rate of the algorithm gradually decreases with the increasing population size in zone II. Reducing the number of populations in zone II accelerates the convergence of the algorithm. When the number of fruit flies in zone II increases, the variation range of the AII evaluation index increases and fluctuates more significantly, indicating a higher diversity of the algorithm. In contrast, less diversity is observed. When testing function F3, the number of fruit flies in zone II is small in Case 3, and the AII evaluation indicators are less volatile and exhibit a steep descent, which shows that the algorithm lacks the development capability. Consequently, the algorithm likely falls into a local optimum. In Case 2, an increase in the number of fruit flies leads to a larger range of variation in the AII evaluation metrics, and thus, the performance of the algorithm is improved. Moreover, according to the test for

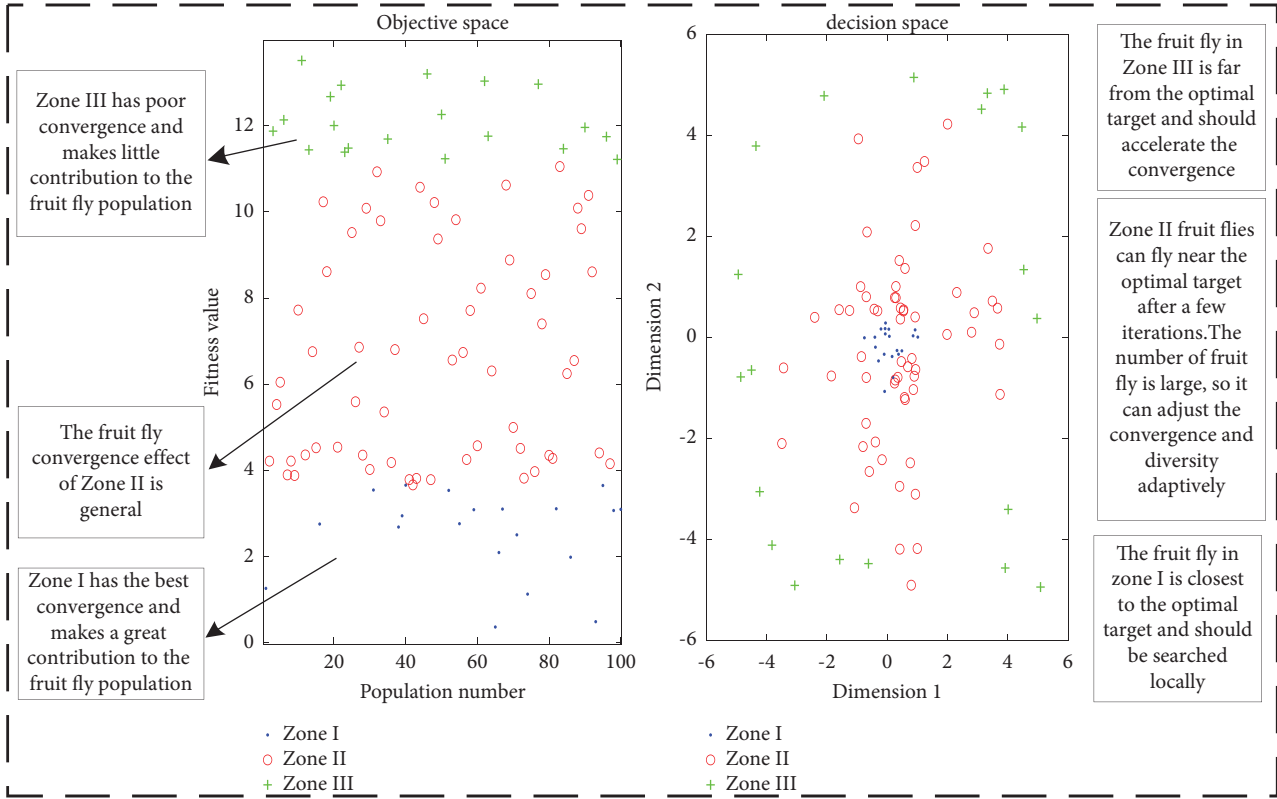


FIGURE 5: Schematic diagram of population zoning.

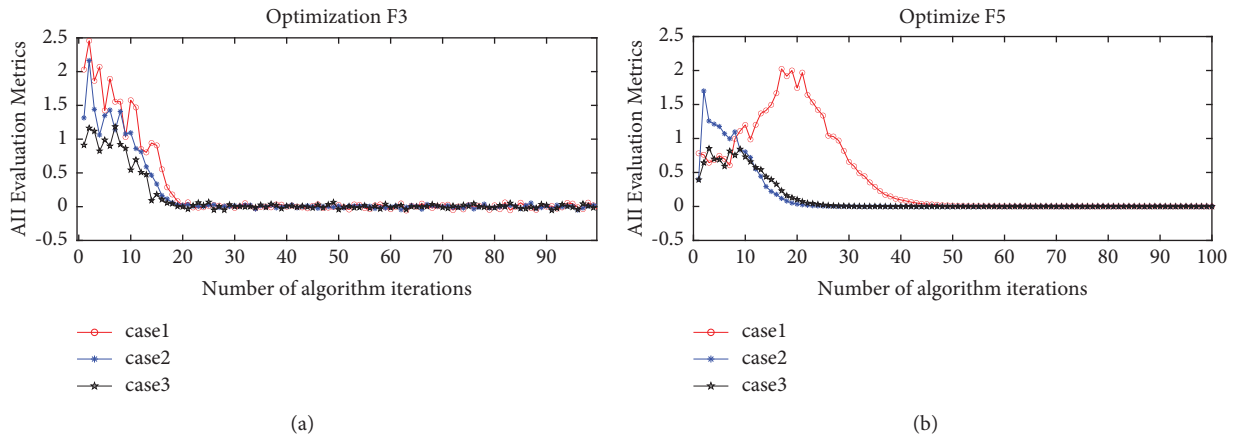


FIGURE 6: Variation of All indicators under different zoning parameters taking schemes. (a) Optimization F3. (b) Optimize F5.

function F5, when the number of fruit flies in zone II decreases to a certain value, the convergence speed of the algorithm does not significantly change. However, the addition of a large number of flies leads to slower convergence of the algorithm. To ensure the performance of algorithm exploration and development, in this study, the partition parameters α and β are set as 0.2 and 0.8, respectively.

3.5. *Steps of the MASD-FOA.* The flowchart of the MASD-FOA is shown in Figure 7, and the implementation process is performed through the following steps.

Step 1. Initialize the maximum number of iterations, partition parameters, population size, and other relevant parameters.

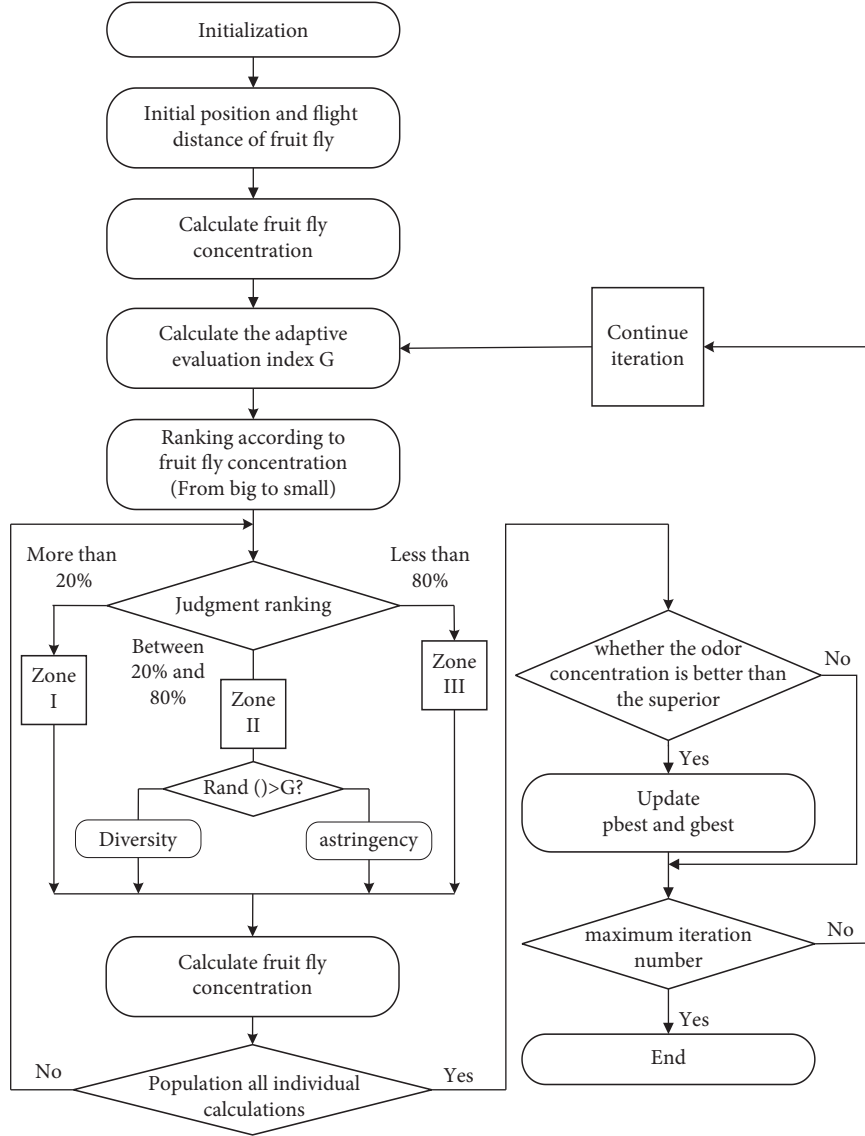


FIGURE 7: MASD-FOA optimization flowchart.

Step 2. Calculate the fitness value of each fruit fly and update the global optimal position $Xgbest_{i,j}^t, Ygbest_{i,j}^t$ and the historical individual optimal position $Xpbest_{i,j}^t, Ypbest_{i,j}^t$ according to

$$\begin{aligned}
 & \text{IF } \text{bestSmell}_t < \text{smellbest} \begin{cases} Xgbest = X_t(\text{bestIndex}), \\ Ygbest = Y_t(\text{bestIndex}), \end{cases} \\
 & \text{IF } pbest_t^i < \text{smell}^i \begin{cases} Xpbest_t^i = X_t^i, \\ Ypbest_t^i = Y_t^i. \end{cases} \quad (7)
 \end{aligned}$$

Step 3. Calculate the convergence evaluation index AII of the average individual increment of the fruit fly population

and rank the values based on the fitness value and zoning population according to equation (2).

Step 4. Update the position of the fruit fly according to the different update strategies for different zones, as indicated in equations (3)–(5).

Step 5. Determine if the algorithm has reached the maximum number of iterations. If the maximum number of iterations has been attained, output the optimal result; otherwise, repeat steps two through four.

In equation (7), smellbest is the global optimal odor concentration, bestSmell_t is the current optimal odor

concentration, $pbest_t^i$ is the current individual odor concentration, $smell^i$ is the historical individual optimal odor concentration, and $bestIndex$ is the returned position coordinate.

3.6. Complexity of Algorithm Calculation. Assuming that the population size is N , the dimension to solve the problem is D , and the number of iterations is T , we can easily get the computational complexity of each part. The computational complexity of fitness value is $O(N)$, the computational complexity of fitness value sorting is $O(N\log N)$ (assuming that the quick sort method is used), the complexity of population partition is $O(N)$, and the position of fruit flies is updated. The complexity is $O(ND)$, and the calculation complexity of the adaptive evaluation index is $O(ND)$. The computational complexity of this algorithm is $O(NT(2D + \log N))$.

4. Verification of the Algorithm Performance

4.1. Test Problems. To verify the performance of the proposed algorithm, the MASD-FOA is compared with four algorithms pertaining to the FOA, PSO, and NSGA in the experiments. To ensure a fair comparison, the number of iterations and the number of populations are set as 100 for all algorithms. Three typical unimodal functions (F1–F3) and three typical multimodal functions (F4–F5) are used in the experiment, as shown in Table 1. The six typical test functions have different characteristics, and the optimization performance of the algorithms for different types of problems can be investigated. Dimension D of the six test functions is set as 2, 5, 8, and 10, as indicated in Table 1. Fifty experiments are conducted for each dimension, and the average value (mean) of the convergence accuracy and the standard deviation (Std) of the optimal value are calculated.

4.2. Algorithm Comparison Results and Analysis. The algorithms are implemented in MATLAB r2020a. The experimental results are shown in Tables 2 and 3. In 24 experiments, the MASD-FOA obtains the optimal average value 21 times and the optimal standard deviation 16 times, and the optimization effect is superior to those of the other algorithms. The NSGA achieves the optimal results in the 2-dimensional test of F3, and PSO achieves the optimal results in the 8-dimensional and 10-dimensional tests of F2. However, the MASD-FOA exhibits a reasonable performance even in these tests. This phenomenon occurs because when the local extreme value is far from the optimal value point, in a certain iteration, the MASD-FOA falls into the local optimum, resulting in partition blur. However, the adaptive strategy increases the diversity of the algorithm, enabling it to rapidly jump out of the local optimum. Therefore, the MASD-FOA exhibits a high performance (the convergence accuracy and optimal result are only slightly different). When the other algorithms fall into the local optimum, because there is no adaptive strategy, the probability of jumping out of the local optimum only by relying on the algorithm's random update strategy is very small. It is

worth noting that the farther the difference between the global optimal and the local optimal is, the smaller the probability of the algorithm jumping out of the local area, which makes the algorithm's convergence effect worse. Therefore, an adaptive update strategy is necessary.

4.3. Comparison of Convergence Curves. Griewank (F5) is a multimodal function that exhibits fluctuations and jumps in the peak shape, and a higher dimension can more accurately reflect the convergence of the algorithm. Therefore, an experiment with 10 dimensions is performed for test function F5 to compare the convergence speed and convergence accuracy of the algorithm. The results are shown in Figure 8. The 2-D line plot shows that the convergence speed of the MASD-FOA is the highest. The log-log scale plot shows that the convergence accuracy of the MASD-FOA is the highest. Figure 8 shows that in the whole convergence process, the MASD-FOA evenly converges. This phenomenon occurs because the MASD-FOA can adaptively adjust the convergence and diversity according to the environment of the algorithm. The multistrategy update mode enhances the exploration and development ability of the algorithm. However, the PSO algorithm drops steeply in the early stage, does not converge in the late stage, and falls into the local optimal solution. A single update strategy will make the algorithm weaker in development and exploration and cannot guarantee the uniform convergence of the algorithm. Therefore, the multistrategy update mode is necessary to improve the algorithm development and exploration capabilities.

4.4. Adaptive Index Change Curve. Figure 9 shows the adaptive G -change curve with 6 test functions and 10 dimensions. To prevent excessive divergence and convergence of the MASD-FOA, the range of variation of the adaptive G is set between 0.2 and 0.8. The change curve is determined according to the proposed evaluation index of the average individual increment of the fruit fly population.

When $G > 0.5$, the diversity of the algorithms is maintained. When $G < 0.5$, the convergence of the algorithm is maintained. When F4 and F6 are optimized, G continuously fluctuates because the test function has more local extremes; however, overall, the value exhibits the same trend as the other four test functions. The fluctuation range is mostly less than 0.5 in the late iteration, conforming to the algorithm convergence variation.

5. Application of MASD-FOA in Microgrid Droop Control

5.1. Overall Structure of the Droop Control System Based on MASD-FOA. The droop control strategy is similar to the primary frequency regulation of the grid and regulation of the grid voltage and frequency by changes in the inverter output power. Since the inverter output impedance is highly inductive, its resistance can be ignored. The droop control equation is defined as

TABLE 1: Six test functions.

Function	Formula representation	Extreme values
F1 (Sphere)	$f(x) = \sum_{i=1}^D x_i^2$	0
F2 (Rosenbrock)	$f(x) = \sum_{i=1}^D [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	0
F3 (Ackley)	$f(x) = -20 \exp(-0.2 \sqrt{1/n \sum_{i=1}^n x(i)^2}) - \exp(1/n \sum_{i=1}^n \cos(2\pi x(i))) + 20 + e$	0
F4 (Rastrigin)	$f(x) = 10 \cdot D + \sum_{i=1}^D (x(i)^2 - 10 \cos(2\pi \cdot x(i)))$	0
F5 (Griewank)	$f(x) = \sum_{i=1}^D x_i^2 / 4000 - \prod_{i=1}^D \cos(x_i / \sqrt{i}) + 1$	0
F6 (Schwefel)	$f(x) = 418.9829 \cdot n - \sum_{i=1}^n x(i) \sin(\sqrt{ x(i) })$	837.9658

TABLE 2: Comparison of convergence accuracy and standard deviation of optimal values of different algorithms.

		2 dim				5 dim			
		MASD-FOA	FOA	PSO	NSGA	MASD-FOA	FOA	PSO	NSGA
F1	Mean	7.03e-21	1.87e-4	1.06e-4	1.22e-6	3.38e-20	0.001	2.21e-9	3.41e-10
	Std	4.15e-20	1.01e-5	1.49e-11	2.18e-7	2.0e-19	4.24e-5	6.8e-9	1.51e-15
F2	Mean	7.03e-21	1.87e-4	1.06e-4	1.22e-6	3.38e-20	0.001	2.21e-9	3.41e-10
	Std	9.41e-4	0.0011	6.56e-09	0.2651	0.1934	0.1708	1.4519	0.1645
F3	Mean	9.56e-11	0.043	2.35e-6	8.88e-16	1.21e-10	0.067	7.57e-5	2.24e-9
	Std	1.99e-10	0.2555	5.48e-06	3.88e-09	3.30e-10	0.0018	0.1646	3.06e-08
F4	Mean	7.93e-15	0.0379	2.93e-9	1.85e-10	7.11e-15	0.7218	3.7809	5.03e-12
	Std	0	0.3926	0.00995	0	7.11e-14	1.1565	1.7898	3.99e-9
F5	Mean	1.11e-16	6.31e-5	2.00e-8	1.26e-13	1.11e-16	2.13e-4	0.0155	0.0031
	Std	0	0.0027	0.0033	0	0	0.0016	0.0115	8.93e-4
F6	Mean	837.96	748.05	830.6269	1.9e-3	1.9e+3	2.04e+3	2.08e+3	2.08e+3
	Std	0.0221	52.2848	4.91e-12	0.295	18.4367	72.787	1.5e-9	1.7428

Bold numbers are the best optimization results.

TABLE 3: Comparison of convergence accuracy and standard deviation of optimal values of different algorithms.

		8 dim				10 dim			
		MASD-FOA	FOA	PSO	NSGA	MASD-FOA	FOA	PSO	NSGA
F1	Mean	1.69e-20	0.0024	2.3e-7	4.21e-8	3.31e-19	0.0036	3.12e-6	7.78e-6
	Std	1.10e-19	1.33e-4	2.28e-7	8.25e-8	5.79e-19	1.86e-6	3.36e-6	4.17e-5
F2	Mean	6.8565	5.4518	3.3014	6.8612	8.2214	7.694	7.2325	8.8979
	Std	0.0141	0.2007	2.0217	0.1682	0.0835	0.2841	11.1522	0.2525
F3	Mean	1.65e-10	0.0852	0.2667	4.93e-4	6.08e-10	0.00947	0.601	0.0025
	Std	2.53e-10	0.0024	0.6674	2.74e-4	4.33e-10	0.0028	0.8054	0.0036
F4	Mean	1.42e-14	1.4299	8.4009	1.49e-4	1.42e-5	2.3033	13.2594	0.0201
	Std	0.0381	2.37	4.4361	0.0087	0.6042	2.6712	5.8735	0.1046
F5	Mean	4.28e-8	3.58e-4	0.0317	1.3e-3	1.78e-14	4.64e-4	0.0309	0.00201
	Std	0	1.57e-5	0.0309	0.00192	0.001	0.059	0.0234	0.0039
F6	Mean	3.15e+3	3.28e+3	3.33e+3	3.33e+3	4.16e+3	4.09+3	4.15e+3	4.16e+3
	Std	27.2146	134.3186	1.33e-7	2.4722	39.6465	47.0489	0.526	2.7432

Bold numbers are the best optimization results.

$$\begin{cases} f = f_0 - m(P_0 - P), \\ E = E_0 - n(Q_0 - Q), \end{cases} \quad (8)$$

where P_0 is the reference active power; Q_0 is the reference reactive power; f_0 is the reference frequency, generally set as 50 Hz; E_0 is the reference voltage; m and n are droop control coefficients of the active and reactive power, respectively; E is the actual voltage; and f is the actual frequency.

The overall block diagram of the MASD-FOA applied to droop control is shown in Figure 10. The droop control strategy includes voltage and current dual loop closed control, which refers to the decoupling control of the active

and reactive power. In the event of a sudden change in the load, the dual closed-loop voltage and current control schemes regulate the inverter to track the load power, control voltage, and current stability. DC voltage source is used to provide electric energy for the load.

The voltage outer loop refers to the difference in the given reactive power and actual reactive power after n regulations. The value is compared with the given voltage to obtain the difference and further compared with the value after the outer loop controller to obtain the reference value of the current inner loop control to ensure a stable output voltage. The regulation formula is as follows:

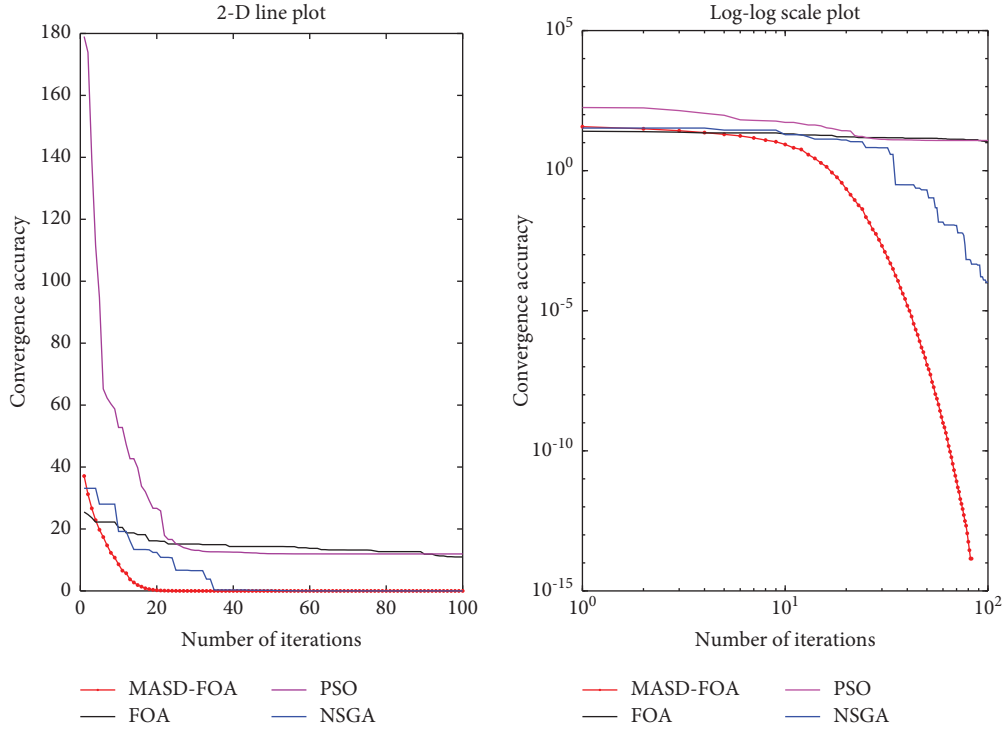


FIGURE 8: Comparison of convergence speed and accuracy of optimized F5.

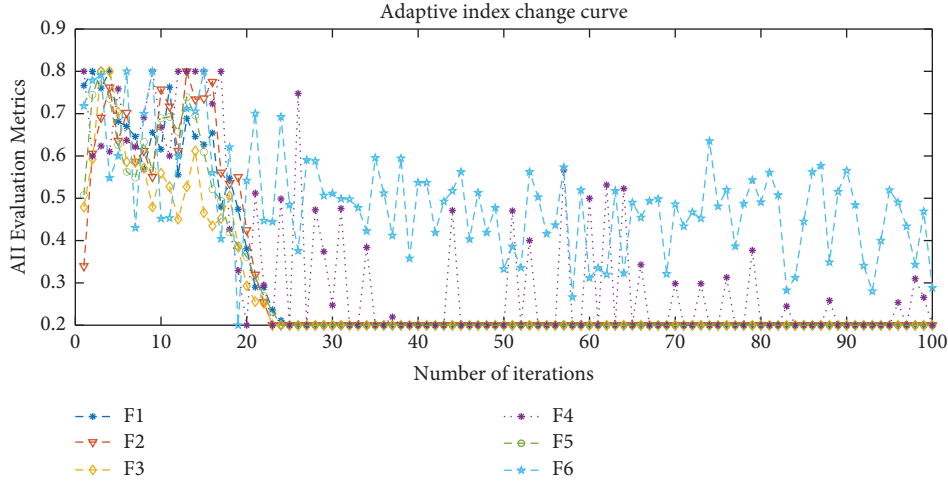


FIGURE 9: Adaptive index change curve.

$$\begin{cases} i_{dref} = C_f \left(k_p + \frac{k_i}{s} \right) (u_{dref} - u_d) - \omega C_f u_d, \\ i_{qref} = C_f \left(k_p + \frac{k_i}{s} \right) (u_{qref} - u_q) + \omega C_f u_q, \end{cases} \quad (9)$$

where u_{dref} and u_{qref} are the reference voltages and u_d and u_q are the instantaneous voltages. k_p and k_i are the proportional and integral parameters of the PI controller, respectively. ω is the grid angular frequency.

The current inner loop refers to the difference between the reference value of the current inner loop and feedback

current and current internal loop regulation via the internal loop PI controller. The adjustment formula is as follows:

$$\begin{cases} u_{sd} = \left(k_p + \frac{k_i}{s} \right) (i_{dref} - i_d) - \omega L i_d + u_d, \\ u_{sq} = \left(k_p + \frac{k_i}{s} \right) (i_{qref} - i_q) - \omega L i_q + u_q. \end{cases} \quad (10)$$

The basic principle of the online optimization of the dual closed-loop PI parameters for a microgrid drooped control system is to optimize the parameters of the four PI

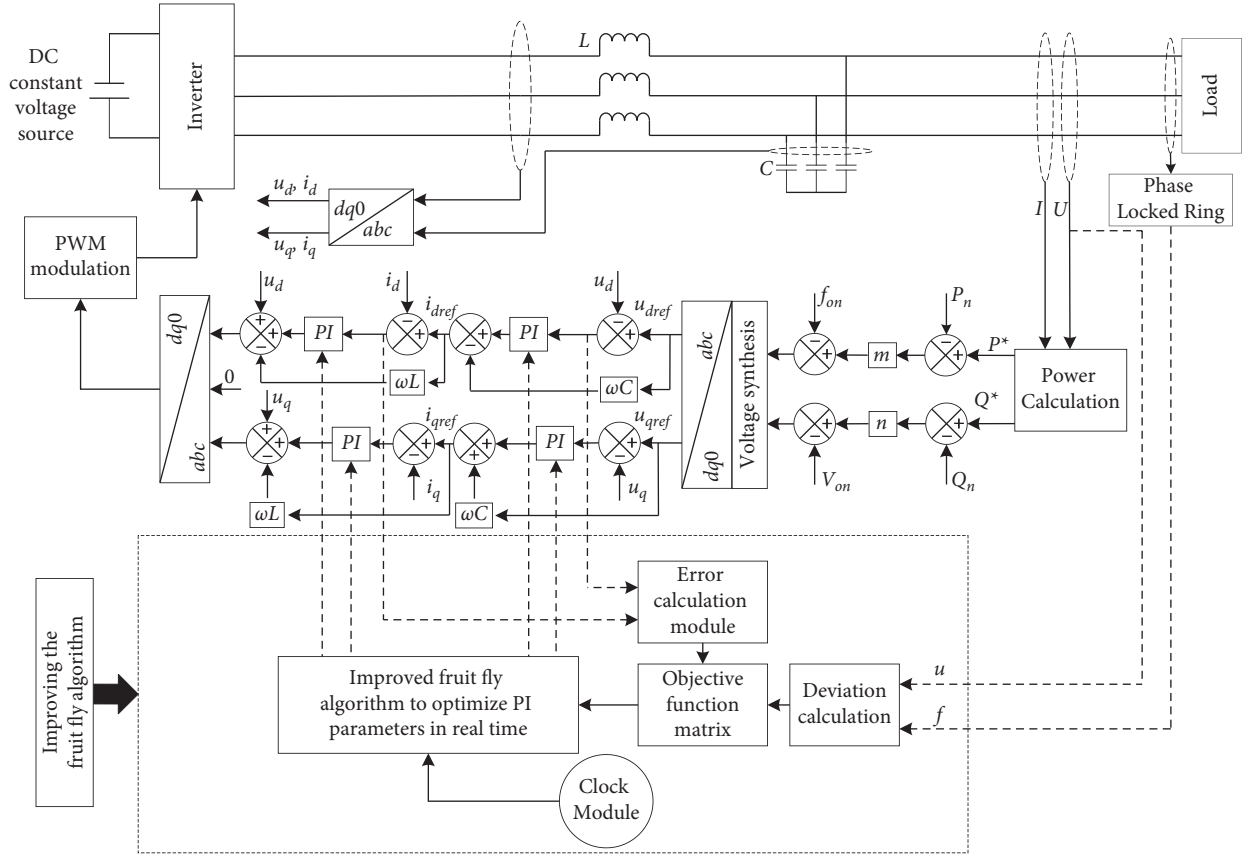


FIGURE 10: Overall structure of the improved fruit fly optimization algorithm for optimal droop control.

controllers defined in equations (9) and (10) by using the MASD-FOA algorithm: $k_{p,1}, k_{i,1}, k_{p,2}, k_{i,2}, k_{p,3}, k_{i,3}, k_{p,4}, k_{i,4}$.

5.2. Determination of the Objective Function. In traditional PI controller optimization, the PI error evaluation performance index $I_{TAE} = \int_0^{\infty} t|e(t)|dt$ is often used as the objective function. Considering the droop control, the change in the output power is realized by adjusting the frequency and voltage, and the stability of the frequency and voltage can ensure the stability of the active and reactive power. Therefore, the frequency deviation Δf and the voltage deviation ΔU are introduced in the objective function. Specifically, $\Delta f = f - f_{ref}$ and $\Delta U = U - U_{ref}$, where Δf and ΔU are the system reference values. The objective function matrix is defined as

$$E_1(t) = \left[\int |\text{error}|dt, \int |f - f_{ref}|dt, \int |U - U_{ref}|dt \right], \quad (11)$$

$$\theta_1 = [1, 1, 1]^T,$$

where error is the error of the PI controller and θ_1 is the weight matrix of the droop control.

When the microgrid is optimized using equation (11) as the objective function, the microgrid output is expected to be unstable. Figure 11 shows that in the multiple simulation experiments, when the objective function is extremely small, the output power of the microgrid exhibits four

characteristics: startup oscillation, small amplitude oscillation, small amplitude rise, and continuous oscillation. These four phenomena are not conducive to the stable operation of microgrids. By changing the number of iterations of the algorithm, resetting the initial parameters and adjusting the step size, these phenomena cannot be effectively suppressed. A several-cause analysis indicates that these four cases correspond to the PI error curve. For example, when the PI error output oscillates, the inverter output power also oscillates. Therefore, while the stability of the PI error is guaranteed, the stable output of the inverter can be ensured. To solve this problem, the absolute value integral term of the PI error derivation is introduced in the objective function in this paper. When the PI error oscillates or deviates, the value of the objective function continuously increases, and the unstable output of the inverter can be avoided in the optimization iteration. The new objective function is shown as follows:

$$E(t) = \left[\int |\text{error}|dt, \int \left| \frac{d\text{error}}{dt} \right| dt, \int |f - f_{ref}|dt, \int |U - U_{ref}|dt \right],$$

$$\theta = [1, 1, 1, 1]^T. \quad (12)$$

5.3. Optimization Ideas and Processes. The core concept of the optimization is to solve for the nonnegative minimum of the objective function. Online optimization is transformed to a mathematical solution problem. The proposed

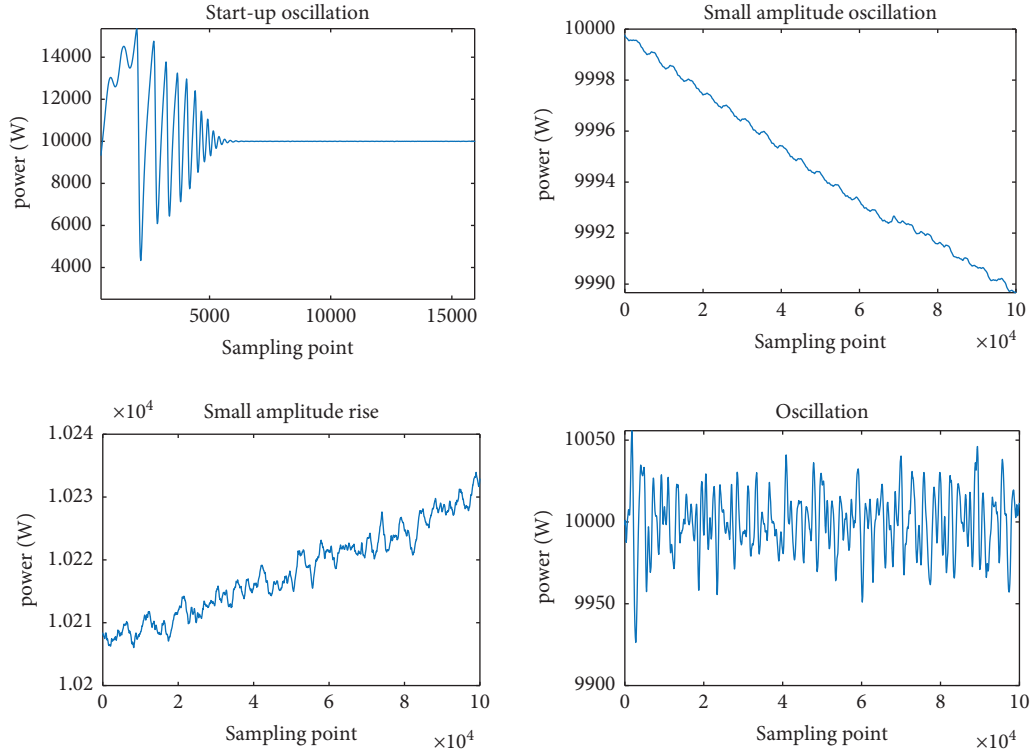


FIGURE 11: Power oscillations and deviation.

algorithm is applied to microgrid-drooped control, and the optimization process of the algorithm is shown in Figure 12. During microgrid operation, when the microgrid topology changes, the control module sends the deviation information to the MASD-FOA module. The process can be summarized as follows: (1) Perform weighted calculation of the deviation information. (2) Calculate the target fitness function and result of the fruit fly concentration. (3) In this paper, the fruit fly population is divided into three regions according to Formula (2), and then the multistrategy model is updated according to the performance of the fruit fly and the environment in which the algorithm is located, and the calculation is iteratively calculated until the end. The MASD-FOA module interacts with the control module in real time, transmitting the parameters to the PI controller of the control module and dual closed-loop controls to regulate the voltage and current, and a sine wave pulse width modulation (SPWM) signal is generated.

5.4. Microgrid Model. To verify the effectiveness of the proposed control strategy, build Simulink simulation according to the basic principle of Section 5.1; set the DC voltage source as 800 V according to the load capacity in the simulation [32]. Other simulation parameters are shown in Table 4. The line impedance is negligible, and off grid operation is considered. Figure 13 shows a frame diagram of multiple microsources in parallel.

Experiment 1. Optimization results of different objective functions.

Experiment 1 verifies that the introduction of the PI error derivative absolute value integral term into the objective function can effectively suppress the oscillation and deviation of the inverter power output. Only microsource 1 runs for 0.2 s (to more clearly display the output waveform at the startup time of the microgrid). MASD-FOA is used to optimize microsource 1; load: 3000 W, 1000 Var. In situations 1 and 2, Formulas (12) and (11) are used as the objective functions for the optimization, respectively. In this case: STS1 and STS3 are closed, and STS1 and STS3 are disconnected.

Experiment 2. Control effects of different strategies.

Experiment 2 verifies the performance of the MASD-FOA online optimized PI control strategy. When the topology of the microgrid changes, the MASD-FOA can optimize the PI parameters of the microgrid in time and ensure the stable output of the inverter. At 0 s, microsource DG1 works independently. In this case, STS1 and STS3 are closed, and STS1 and STS3 are disconnected. load: 3000 W, 1000 Var. After 0.5 s, microsources DG1 and DG2 simultaneously operate with loads of 8000 W and 3000 Var. In this case, STS1 to STS4 are all closed. Equation (12) is used as the objective function.

In the experiment, MASD-FOA is compared with FOA, PSO, NSGA, and traditional methods. Among them, the traditional method 1: do not adjust the PI parameters; the traditional method 2: manually adjust the PI parameters;

5.5. Comparison and Analysis of Experiments

5.5.1. Comparison and Analysis of Experiment 1. As shown in Figures 14 and 15, in situation 1, the output power of the

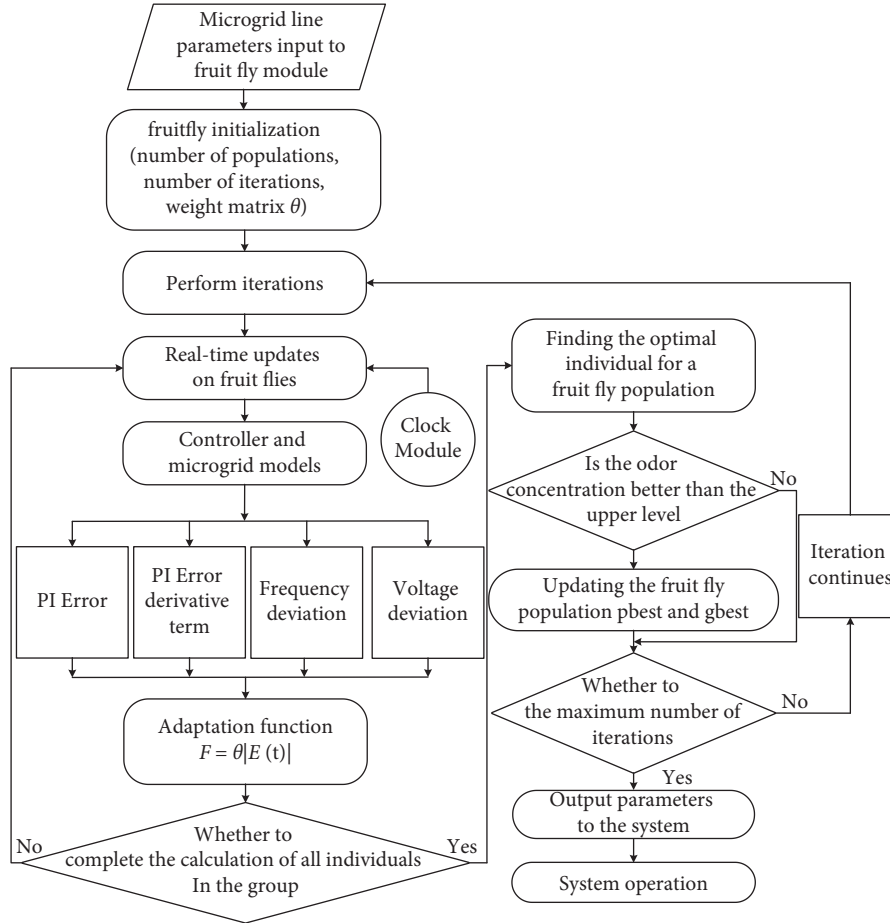


FIGURE 12: Flowchart of inverter droop control algorithm for microgrid based on improved fruit fly optimization algorithm optimization.

TABLE 4: Simulation parameters.

Parameter name	Numerical value
DC side voltage (V)	800
Filter inductors (H)	1.35×10^{-3}
Filter capacitor (F)	50×10^{-6}
Rated frequency (Hz)	50
Rated voltage (V)	311
Active power droop factor	$1e^{-4}$
Reactive power droop factor	$3e^{-4}$
Reference active power (W)	3000
Reference reactive power (Var)	0

inverter slowly fluctuates in a small range, and the range of fluctuation is constant. After optimization based on Formula (12), when the system starts, the output power of the inverter considerably fluctuates. In the steady state, the output power continues to oscillate at a high frequency, and the oscillation range is large. With time, the output power exhibits a small deviation. Therefore, the absolute value integral term of the PI error derivation can suppress the oscillation and deviation of the output power.

5.5.2. Comparison and Analysis of Experiment 2. Figures 16 and 17 show the frequency and voltage output by the inverter, respectively. The MASD-FOA is adopted

to ensure prompt responses of the inverter. In terms of the frequency, the output is stable, and no frequency overshoot occurs. At 0.5 seconds, both the topology and the load have changed (the topology is changed after STS2 is closed, and the load is changed after STS4 is closed), so it is necessary to reoptimize the droop control PI parameters. The increase of the load leads to the increase of the output power because the increase of the output power causes the frequency to decrease slightly. The frequency stability complies with the national standard GB/T15945-1995 [33]: the frequency of the power system is controlled within the range of 50 ± 0.2 Hz. The other four algorithms exhibit frequency oscillations; for example, after NSGA optimization, the frequency of the microgrid greatly

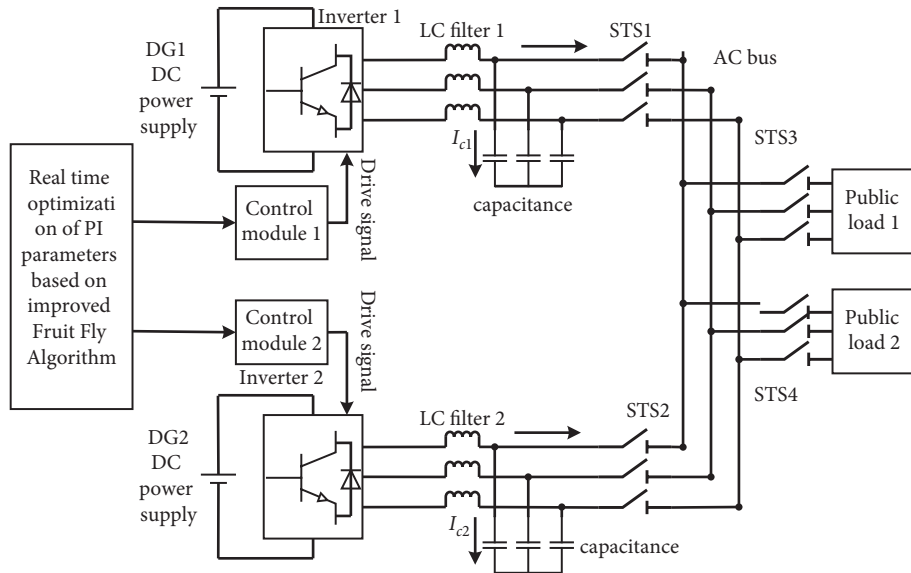


FIGURE 13: Online optimization framework of microgrid based on MASD-FOA.

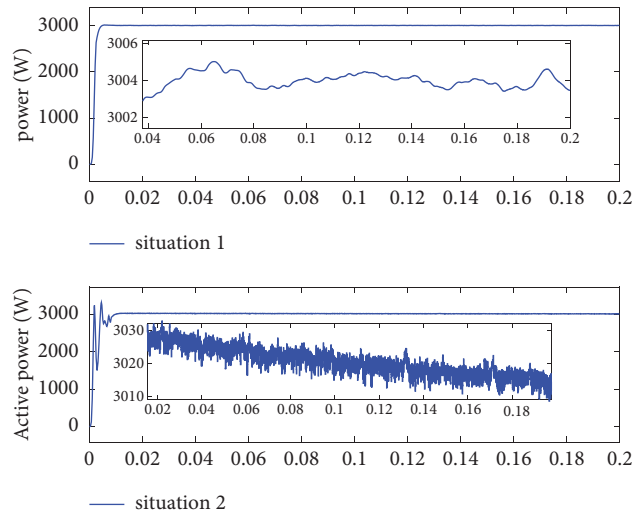


FIGURE 14: Output of active power.

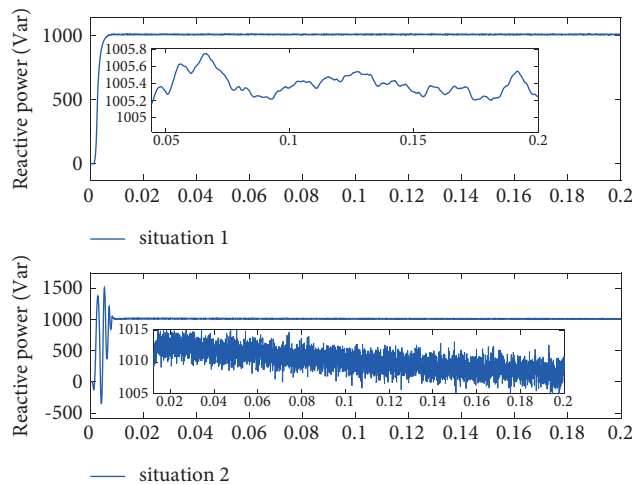


FIGURE 15: Output of reactive power.

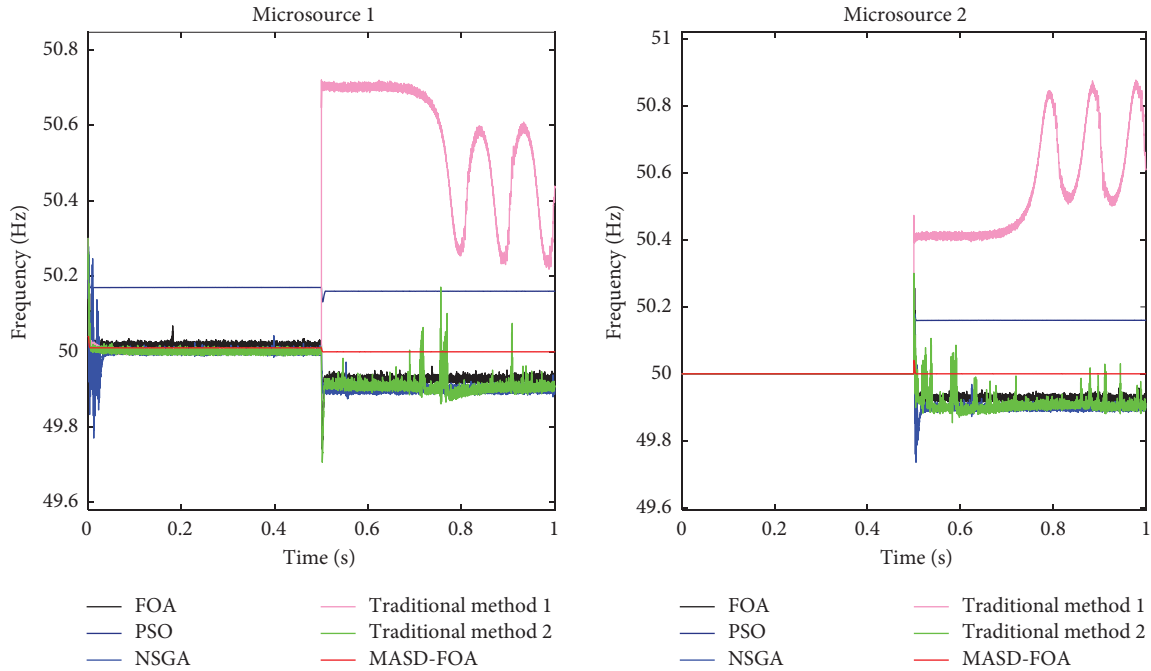


FIGURE 16: Comparison chart of inverter output frequency.

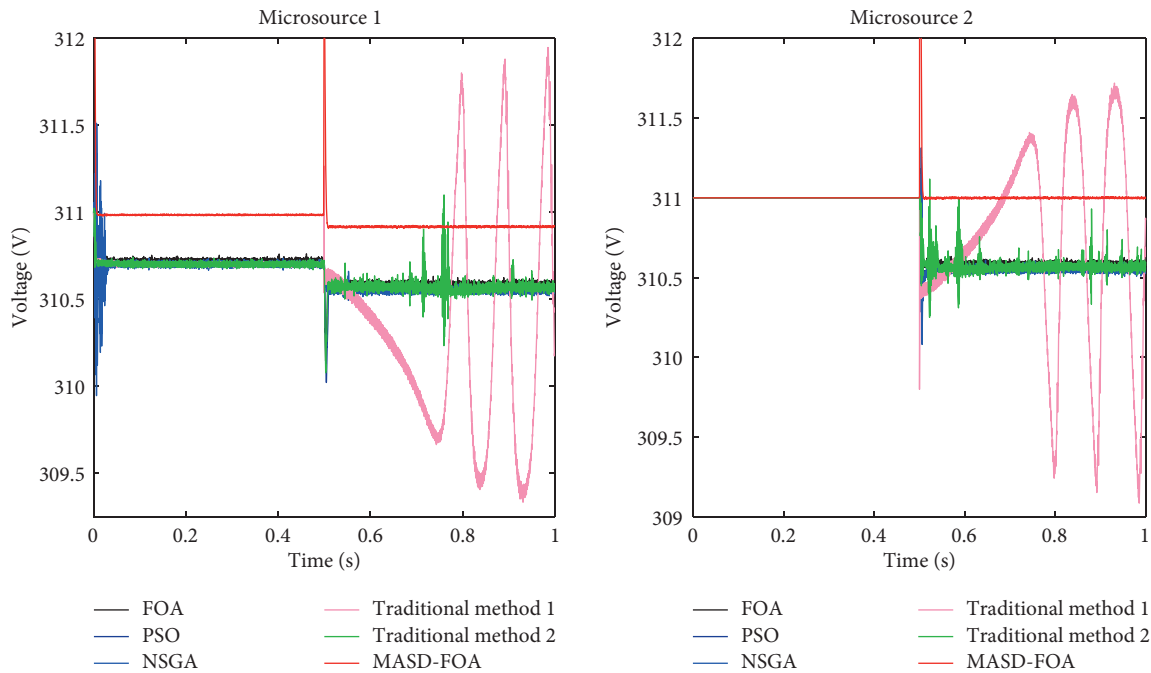


FIGURE 17: Comparison chart of inverter output voltage.

fluctuates at the moment of startup, and the overshoot reaches 0.3 Hz. Although PSO stabilizes the frequency output, the difference in the output frequency and reference frequency is large. It can be observed from Figure 16 that the error of inverter output frequency can be reduced by the MASD-FOA algorithm, which is caused by the introduction of the frequency error term in the objective function. In the algorithm optimization, the steady-state frequency with smaller error can be obtained.

Both exploitation and exploration capabilities of the proposed algorithm are greatly improved by population partitioning, multistrategy, and adaptive scheme, respectively. Facing the complex nonlinear systems, according to the evolutionary environment where the algorithm is located, the convergence and diversity of the algorithm are adaptively adjusted by the MASD-FOA algorithm, and the best PI parameters can be found. With the increase of dimension of the optimization problem,

the remarkable performance of the MASD-FOA algorithm can be still remained, and the small error of the inverter output is guaranteed. In terms of the voltage stability, the MASD-FOA can render the inverter voltage output highly stable, and no fluctuation occurs in a small range, corresponding to the minimum error. In the traditional method 1, because there is no online optimization of PI parameters, causing huge fluctuations in microgrid frequency and voltage, microgrid cannot work properly. In the traditional method 2, the PI parameters are properly adjusted according to human experience. It can be seen that the artificial adjustment accuracy is not high, especially with the increase of microsources, the adjustment difficulty increases, resulting in high-frequency oscillation of inverter output voltage and frequency, and the oscillation amplitude is large.

Figure 18 shows that the MASD-FOA ensures the rapid response of the inverter when the microsource is started or the topology changes. Almost no overshoot occurs, the inverter can promptly follow the load change, and a precise power output can be guaranteed in the steady state. In contrast, the NSGA algorithm decreases the response speed of the inverter. The standard FOA increases the steady-state error of the inverter and decreases the response speed. The MASD-FOA is compared with the PSO algorithm: the locally magnified view indicates that in the case of the MASD-FOA, the response of the inverter is a smooth rising curve; however, PSO causes the output power of the inverter to oscillate at startup. At 0.5 s, in the case of the MASD-FOA, the steady-state response time of the inverter is extremely small, and no overshoot occurs. In contrast, the steady-state response time of the inverter is extremely large in the case of the PSO. In the traditional method 1, the output of the inverter will oscillate because the PI parameters do not adjust in time when topology changes. The original PI parameters do not enable stable operation of the microgrid. In the traditional method 2, due to the insufficient regulation accuracy, the output power of the inverter fluctuates after 0.5 seconds, and the steady-state error is large.

In order to measure the deviation between the output power of inverter and the required power of load, Formulas (13) and (14) are used to calculate the relative error of the power.

$$P_{\text{error}} = \frac{P_i - P_{\text{load}}}{P_{\text{load}}} \% \quad (13)$$

$$Q_{\text{error}} = \frac{Q_i - Q_{\text{load}}}{Q_{\text{load}}} \% \quad (14)$$

where P_i is the effective value of active power, Q_i is the effective value of reactive power, P_{load} is the required active power of the load, and Q_{load} is the required active reactive power of the load.

Table 5 shows the comparison of the results under different algorithm optimizations. Microsources 1 and 2 correspond to the statistics of operation for 1 s and 0.5 s, respectively. The overshoot amount is the maximum overshoot that occurs during the entire operation of the

microsource. The rise time is the time required for the inverter output to reach the first extreme value at system startup. The steady-state response time is the time at which the system enters the steady state. The performance of the MASD-FOA is summarized in Table 6. The MASD-FOA is slightly inferior to the other algorithms in terms of the overshoot; however, it outperforms the other algorithms in terms of the error and response speed. The performance in terms of the amount of overshoot is not optimal, but the value of overshoot is extremely small and in accordance with microgrid operation requirements. In terms of the stabilization time, the stable state of the system can be induced in 0.005 and 0.008 s after MASD-FOA optimization, almost 40 times faster than other algorithms.

5.5.3. Comparison of Running Time. It is difficult to design a reasonable experiment to explain the computational complexity or computational time of the algorithms. Many methods have been proposed [34–36]. A general and simple one is adopted in this paper, that is, the running time of the algorithms under the same environmental platform and the same maximum number of iterations. On the basis of comparing the convergence and diversity, the running time of each algorithm is measured simultaneously to fairly demonstrate the computational complexity of the proposed MASD-FOA algorithm.

In this article's optimization strategy, all algorithms set the maximum number of iterations to 60. In the first optimization, each algorithm will give the same initial value range. The initial value range of the next optimization is determined by the optimal value of the previous optimization, and this strategy is called optimal delivery strategy (ODS). The sampling time of the algorithm is 0.001 seconds. The adaptive strategy proposed in this paper can detect the environment of the algorithm, and the algorithm stops optimization when the detection environment indicators converge. For an algorithm without an adaptive strategy, the objective function F is used to determine whether the convergence is completed, the optimization is stopped when the convergence is completed, and the optimization is stopped when the maximum number of iterations is reached if the convergence is not completed. The comparison of different algorithms for the computational time is presented in Table 6.

It can be seen from Table 6 that the optimization strategy proposed in this paper is much shorter than other strategies in terms of optimization time because the strategy proposed in this paper greatly reduces the number of iterations of the algorithm and makes the algorithm converge quickly. In particular, the method of determining the algorithm initial value of this time based on the optimal value of the previous time can significantly reduce the number of iterations. This is because the use of ODS can reduce the range of optimization to a certain extent, and adaptive adjustment will also intensify the convergence of the algorithm, converging rapidly in the optimal direction. Therefore, adaptive regulation strategy plays an important role in online optimization.

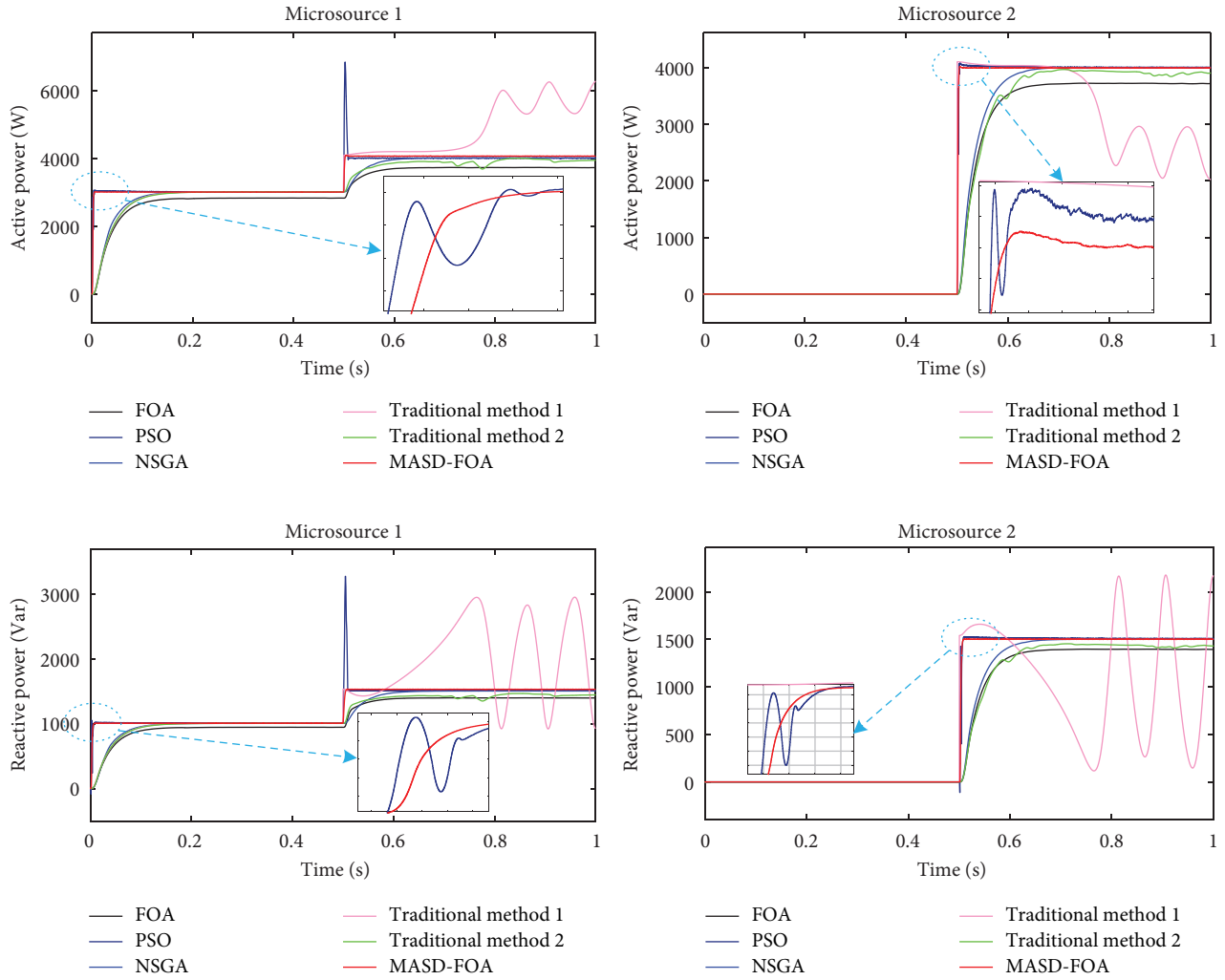


FIGURE 18: Comparison chart of inverter output power.

TABLE 5: Comparison of algorithm stability margins.

Comparison algorithm	Microsource	Relative error of active power	Relative error of reactive power	Frequency deviation	Voltage deviation	Pi1 error	Pi2 error	Overshoot	Rise time	Steady-state response time
MASD-FOA	Microsource 1	1.061	1.428	0.005352	0.1063	0.4554	0.6871	104	0.005	0.005
	Microsource 2	0.217	0.3463	$8.681e^{-5}$	0.01558	0.424	0.3649	33	0.008	0.008
FOA	Microsource 1	10.62	10.89	0.04618	0.3513	10.85	6.532	0	0.235	0.235
	Microsource 2	7.067	7.254	0.03628	0.2077	6.02	3.169	0	0.25	0.25
PSO	Microsource 1	1.182	2.798	0.1648	0.3784	0.6387	0.8562	3690	0.005	0.22
	Microsource 2	0.3488	0.6297	0.08	0.2251	0.5632	0.5424	95	0.005	0.2
NSGA	Microsource 1	4.673	5.191	0.05449	0.3764	5.205	3.64	0	0.2	0.2
	Microsource 2	3.736	3.848	0.05112	0.2247	2.241	1.421	0	0.2	0.2

TABLE 5: Continued.

Comparison algorithm	Microsource	Relative error of active power	Relative error of reactive power	Frequency deviation	Voltage deviation	Pi1 error	Pi2 error	Overshoot	Rise time	Steady-state response time
Traditional method 1	Microsource 1	312.3	641.1	1.55	5.454	28.31	35.05	3028	-	-
	Microsource 2	9.658	9.79	0.02511	0.1958	10.32	1.835	952	-	-
Traditional method 2	Microsource 1	7.256	7.9132	0.1519	0.581	7.842	7.385	0	0.3	0.3
	Microsource 2	5.618	5.869	0.1256	0.431	6.89	6.235	0	0.3	0.3

Bold numbers are the best optimization results.

TABLE 6: Comparison of different algorithms for the computational time (all results were averaged on 30 independent runs).

	Time (s)	MSAD-FOA	FOA	PSO	GA
Number of iterations	0	43	60	60	60
	0.5	27	60	49	54
Optimization time (s)	0	2.15	3.00	3.00	3.00
	0.5	1.35	3.00	2.45	2.70

6. Conclusion

The response speed, output power oscillation, and voltage and frequency stability of inverters in microgrid droop control are studied, and the following conclusions are derived: (1) when the microgrid topology changes, the proposed three-partition multistrategy adaptive fruit fly optimization algorithm can optimize the PI parameters of the microgrid droop control in real time and enhance the robustness of droop control. Consequently, the inverter exhibits a faster response, a smaller error, and more stable output. (2) Microgrid experiments with different objective functions verify that the PI error derivation absolute value integral term can effectively suppress the oscillation and deviation of the inverter output power. (3) Six test functions are used. Through a comparison with the FOA, PSO, and NSGA algorithms, the effectiveness of the three-partition, multistrategy, and adaptive update mode is verified. This framework can help exploit the development and exploration ability of each fruit fly when solving different problems. Consequently, the MASD-FOA exhibits a high convergence speed and convergence accuracy. (4) In noninitial optimization, the combination of adaptive strategy and ODS greatly reduces the number of iterations of the algorithm, thereby reducing the time for online optimization.

The three-partition multistrategy adaptive fruit fly optimization algorithm proposed in this paper can not only solve the problem of optimizing droop control parameters, but the algorithm can be applied to more scenarios in the future. Compared with other algorithms, the advantages of this algorithm are as follows: the algorithm can detect the algorithm environment, adaptively adjust the convergence and diversity of the algorithm, quickly jump out of the local optimum in the development phase, and improve the convergence accuracy during the exploration phase. The

multistrategy update mode can maximize the development and exploration capabilities of fruit flies, thereby avoiding invalid calculations.

Data Availability

Research data are provided in the manuscript; if additional data are needed, the author of the article can also provide it appropriately.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. U1804143), the Technology Planning Project of Henan Province of China Grant (no. 202102210295), the Fundamental Research Funds for the Universities of Henan Province Grant (no. NSFRF210424), the Doctoral Scientific Research Foundation of Henan Polytechnic University Grant (no. B2017-20), and The Young Key Teacher Program of Henan Polytechnic University Grant (no. 2019XQG-17).

References

- [1] W. Li, X. Wen, and J. Zhang, "Current sharing strategy of AC/DC rectifiers based on new droop control," *Electric Machines and Control*, vol. 24, no. 07, pp. 62–72, 2020.
- [2] S. S. Reddy and B. K. Panigrahi, "Fuzzified multi-objective particle swarm optimisation for the solution of economic and emission dispatch problem," *International Journal of Power and Energy Conversion*, vol. 8, no. 3, Article ID 276, 2017.
- [3] S. S. Reddy and J. A. Momoh, "Minimum Emission Dispatch in an Integrated Thermal and Wind Energy Conservation System Using Self-Adaptive Differential Evolution," in *Proceedings of the 2016 IEEE PES PowerAfrica*, Livingstone, Zambia, July 2016.
- [4] J. A. Momoh and S. S. Reddy, "Value of Hardware-In-Loop for Experimenting Microgrid Performance System Studies," in *Proceedings of the 2016 IEEE PES PowerAfrica*, pp. 199–203, Livingstone, Zambia, July 2016.

- [5] H. Liang, C. Zheng, and Y. Gao, "Research on improved droop control strategy for microgrid," *Proceedings of the CSEE*, vol. 37, no. 17, pp. 4901–4910+5209, 2017.
- [6] X. Chai, C. Zhang, and X. Zhao, "Self-recovery droop based active synchronization control and parameter optimization of distributed generator [J]," *Automation of Electric Power Systems*, vol. 44, no. 19, pp. 128–133, 2020.
- [7] F. Gao, X. Ling, and W. Wang, "PI parameters joint tuning FOR individual pitch controller of large wind turbine," *Acta Energiæ Solaris Sinica*, vol. 39, no. 02, pp. 307–314, 2018.
- [8] Y. Zhu, Z. Wang, Y. Liu, H. Wang, N. Tai, and J. Jiang, "Optimal control of microgrid operation based on fuzzy sliding mode droop control," *Energies*, vol. 12, no. 19, Article ID 3600, 2019.
- [9] Shivam and R. Dahiya, "Distributed control for DC microgrid based on optimized droop parameters," *IETE Journal of Research*, vol. 66, no. 2, pp. 192–203, 2020.
- [10] M. Taghizadeh, M. Mardaneh, and M. Sha Sadeghi, "A new method of voltage and frequency control in isolated micro-grids using enhanced droop controller optimized by frog algorithm," *Journal of Renewable and Sustainable Energy*, vol. 6, no. 1, pp. 1017–1022, 2014.
- [11] P. Xie, Z. Cai, and P. Liu, "Cooperative optimization of energy storage capacity for renewable and storage involved micro-grids considering multi time scale uncertainty coupling influence," *Proceedings of the CSEE*, vol. 39, no. 24, pp. 7126–7136+7486, 2019.
- [12] T. A. Jumani, M. W. Mustafa, M. M. Rasid, N. H. Mirjat, Z. H. Leghari, and M. S. Saeed, "Optimal voltage and frequency control of an islanded microgrid using grasshopper optimization algorithm," *Energies*, vol. 11, no. 11, Article ID 3191, 2018.
- [13] I. Chathoth, R. Kumar, S. Sindhu, and T. Krishnan, "Optimal fractional order PID controller for automatic generation control of two-area power systems," *International Transactions on Electrical Energy Systems*, vol. 25, no. 12, pp. 3329–3348, 2016.
- [14] M. F. Roslan, A. Q. Al-Shetwi, M. A. Hannan, P. J. Ker, and A. W. M. Zuhdi, "Particle swarm optimization algorithm-based PI inverter controller for a grid-connected PV system," *PloS One*, vol. 15, no. 12, Article ID e0243581, 2020.
- [15] R. Dong, S. Liu, and G. Liang, "Research on control parameters for voltage source inverter output controllers of micro-grids based on the fruit fly optimization algorithm," *Applied Sciences*, vol. 9, no. 7, Article ID 1327, 2019.
- [16] L. Wang, C. Lv, and Y. Zeng, "Literature survey of fruit fly optimization algorithm," *Control and Decision*, vol. 32, no. 07, pp. 1153–1162, 2017.
- [17] G. Ding, Y. Qiao, W. Yi, W. Fang, and L. Du, "Fruit fly optimization algorithm based on a novel fluctuation model and its application in band selection for hyperspectral image," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 1517–1539, 2021.
- [18] X. Yang, W. Li, L. Su, Y. Wang, and A. Yang, "An improved evolution fruit fly optimization algorithm and its application," *Neural Computing & Applications*, vol. 32, no. 14, pp. 9897–9914, 2020.
- [19] C. Xiong and S. Lian, "Structural damage identification based on improved fruit fly optimization algorithm KSCE," *Journal of Civil Engineering*, vol. 25, no. 3, pp. 985–1007, 2021.
- [20] H. Chen, S. Li, A. Asghar Heidari et al., "Efficient multi-population outpost fruit fly-driven optimizers: framework and advances in support vector machines," *Expert Systems with Applications*, vol. 142, no. 15, Article ID 112999, 2020.
- [21] L. Feng, Y. Wang, and L. Han, "Double drive fruit fly optimization algorithm and its application in PID controller," *Control and Decision*, vol. 36, no. 09, pp. 2225–2233, 2021.
- [22] S. Zhang and Y. Chen, "Fruit Fly optimization algorithm for adaptive search cloud escape," *Journal of Huazhong University of Science and Technology (Nature Science Edition)*, vol. 46, no. 09, pp. 40–44+51, 2018.
- [23] W. Lei, X. Wensheng, Z. Liang, L. Qi, and W. Jingli, "An improved fruit fly optimization algorithm based on selecting evolutionary direction intelligently," *International Journal of Computational Intelligence Systems*, vol. 9, no. 1, p. 80, 2016.
- [24] X. Yuan, X. Dai, J. Zhao, and Q. He, "On a novel multi-swarm fruit fly optimization algorithm and its application," *Applied Mathematics and Computation*, vol. 233, no. 3, pp. 260–271, 2014.
- [25] W. Pan and T. Tsao, "A new Fruit Fly Optimization Algorithm: taking the financial distress model as an example," *Knowledge-Based Systems*, vol. 26, no. 2, pp. 69–74, 2012.
- [26] Q. Sun, Q. Sun, and D. Qin, "Adaptive fuzzy droop control for optimized power sharing in an islanded microgrid," *Energies*, vol. 12, no. 1, p. 45, 2018.
- [27] Y. Wang and L. Feng, "Novel double subgroups and partition sampling based fruit fly optimization algorithm," *Journal of Zhejiang University (Engineering Science)*, vol. 51, no. 11, pp. 2292–2298, 2017.
- [28] Y. Wang, J. Zhu, and L. Feng, "Improved fruit fly optimization algorithm based on population density and its application in anomaly detection," *Advanced Engineering Sciences*, vol. 49, no. 05, pp. 127–134, 2017.
- [29] Y. Yang, T. Zhang, W. Yi et al., "Deployment of multistatic radar system using multi-objective particle swarm optimization," *IET Radar, Sonar & Navigation*, vol. 12, no. 5, pp. 485–493, 2018.
- [30] W. Zhang, G. Li, W. Zhang, J. Liang, and G. G. Yen, "A cluster based PSO with leader updating mechanism and ring-topology for multimodal multi-objective optimization," *Swarm and Evolutionary Computation*, vol. 50, Article ID 100569, 2019.
- [31] Q. Feng, Q. Li, and W. Quan, "Overview of multiobjective particle swarm optimization algorithm," *Chinese Journal of Engineering*, vol. 43, no. 06, pp. 745–753, 2021.
- [32] L. Song, North China Electric Power University, China, Beijing, *The Principle and Analysis of Stability of DroopControl in Micro-grid*, vol. 6, pp. 1–60, 2016.
- [33] IEEE 1547.2-2008, 2018 IEEE 1547, IEEE Standard for Interconnection and Interoperability of Distributed Energy Resources with Associated Electric Power Systems Interfaces, vol. 1, pp. 1–138, Article ID 8332112, 2018.
- [34] J. Qiao, F. Li, S. Yang, C. Yang, W. Li, and K. Gu, "An adaptive hybrid evolutionary immune multi-objective algorithm based on uniform distribution selection," *Information Sciences*, vol. 512, pp. 446–470, 2020.
- [35] H. Wang Hu and G. G. Yen, "Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 1, pp. 1–18, 2015.
- [36] R. Kumari, D. Kumar, and V. Chahar, "Impact of controlling parameters on the performance of MOPSO algorithm," *Procedia Computer Science*, vol. 167, pp. 2132–2139, 2020.