Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1500646, 16 pages
https://doi.org/10.1155/2022/1500646

Research Article

Mutigroup-Based Phasmatodea Population Evolution Algorithm with Mutistrategy for IoT Electric Bus Scheduling

Yunxiang Zhu,[1] Fengting Yan,[1] Jeng-Shyang Pan,[2] Lei Yu,[1] Yuanfei Bai,[1] Weigang Wang,[3] Chunxia He,[4] and Zhicai Shi[1,5]

[1]*Shanghai University of Engineering Science, Shanghai 201620, China*
[2]*Shandong University of Science and Technology, Qingdao, Shandong 266590, China*
[3]*Shanghai Municipal Engineering Design and Research Institute (Group) Co., Ltd., Shanghai 200092, China*
[4]*Shanghai Urban Construction Vocational College, Shanghai 201999, China*
[5]*Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, Shanghai 200240, China*

Correspondence should be addressed to Fengting Yan; yanfengting2008@163.com

Received 29 October 2021; Revised 7 December 2021; Accepted 4 January 2022; Published 31 January 2022

Academic Editor: Feng-Jang Hwang

The Phasmatodea population evolution algorithm (PPE) is a novel metaheuristic algorithm proposed in recent years, which simulates the evolutionary trend of stick insect population. In this article, a multigroup-based Phasmatodea population evolution algorithm with mutistrategy (MPPE) is proposed to further improve the overall performance of PPE. During the initialization period, the stick insect population is divided into multiple groups, and the step factor of the flower pollen algorithm is introduced into the population growth model of no more than half of the groups. This makes the population evolution trend diversified and prevents the algorithm from falling into the local optimal solution to a certain extent. In terms of intergroup communication, two communication strategies are adopted to mutate and replace the inferior particles, respectively, which improves the convergence speed and search ability of the algorithm. In the MPPE performance test, we compared it with PPE, five standard algorithms, and other parallel algorithms in CEC 2013 Benchmark Suite. Finally, this algorithm is applied to the IoT based electric bus scheduling for urban waterlogging situation, and the excellent performance of MPPE is verified comprehensively.

1. Introduction

Metaheuristic algorithms are inspired by relevant experiences, behaviors, rules, and mechanisms in the fields of physics, chemistry, biology, society, and art [1]. It is a method based on computational intelligence to solve complex optimization problems with the best or satisfactory solution [2–5]. Because it does not depend on specific problems, metaheuristic algorithms can be widely used in industrial production, economic life, military, and other fields [6–9].

Metaheuristic algorithms can be roughly divided into three categories according to their sources. The first category is derived from human theory. In recent years, the gaining-sharing knowledge based (GSK) algorithm [10] and the volleyball premier league (VPL) algorithm [11] have been proposed, which simulate the development and activities of human life. The second category comes from biological behavior. Among the novel algorithms are island artificial bee colony (iABC) algorithm [12], monarch butterfly optimization (MBO) [13], moth search algorithm (MSA) [14], Harris hawks optimization (HHO) [15], and so on. The third category is inspired by physical phenomena. Such novel algorithms include thermal exchange optimization (TEO) [16], electrosearch (ES) algorithm [17], and lightning attachment procedure optimization (LAPO) [18]. Since people have obtained a lot of inspiration from the infinite nature, more and more nature-inspired algorithms have been proposed.

The Phasmatodea population evolution (PPE) algorithm is a new metaheuristic algorithm proposed in 2020 [19]. Song et al. [20] simplified Phasmatodea population evolution algorithm and applied it to engineering optimization problems. This algorithm is inspired by the evolution trend of stick insect population in the natural environment. The characteristic of stick insect population growth is that it can adapt to a dynamic environment and change autonomously. In the initial stage, there is a favourable growth environment for the population, so it grows exponentially. When the population grows to a certain extent, due to the interference of environmental factors, its growth rate tends to slow down and eventually stops growing.

The advantage of the PPE algorithm is that it has relatively good exploratory ability and stability. In terms of application, it is suitable for various engineering optimization problems. So, this algorithm also has a good generalization ability. But it still has some disadvantages. Its evolutionary mechanism will lead to the generation of some poor particles during the early period, and it converges slowly. When solving some problems, its solution tends to be locally optimal rather than globally optimal.

To address the shortcomings of the PPE algorithm, we attempt to use parallel method to improve [21, 22]. Predecessors have done a lot of work in parallel optimization algorithm research over the years, such as parallel particle swarm optimization (PPSO) [23], parallel genetic algorithm (PGA) [24], parallel ant colony algorithm (PACO) [25–27], and parallel differential evolution algorithm (PDE) [28]. In this study, we introduce multiple strategies into the parallel approach. First, the stick insect population is divided into several groups, all of which evolve simultaneously. Secondly, the step factor of the flower pollen algorithm is introduced into the population growth model of some groups, and its evolution trend is changed to some extent. The diversity of population evolution mechanism makes the algorithm more capable of global exploration. Finally, we adopt two strategies for intergroup communication to improve the overall performance of PPE. The first strategy is to mutate the particles with poor fitness during the early stage to speed up the convergence of PPE. The second strategy is to randomly replace the best particle with other random particles in all the groups during the late period. It is intended to improve the global exploration accuracy of the algorithm. We made the above improvements to PPE and named it MPPE. In the comparative experiment of CEC 2013 Benchmark Suite, we verify the excellent performance of MPPE algorithm.

Public traffic optimization is a hot issue in recent years [29–31]. With the rapid development of new energy technology, electric buses have gradually replaced fuel buses in short-distance transportation and have been popularized in many cities. This change promotes the wide application of intelligent transportation combined on Internet of Things (IoT) technology [32], in which the electric bus scheduling problem has become the focus of many scholars [33–35]. Bus scheduling is a multiobjective combination optimization problem, which has a recognized NP-hard problem. The purpose of this problem is to find an optimal solution in a complex objective function, that is, the best scheduling scheme. The metaheuristic algorithm can effectively calculate the optimal solution in a limited time and space and is not restricted by the problem itself. So, it is suitable for solving this kind of large-scale optimization problem. In recent years, more and more researchers have applied metaheuristic algorithms to bus scheduling [36–38], which proves their effectiveness in solving such problems.

However, traditional metaheuristic algorithms often have some limitations for the solution results of the bus scheduling optimization. Novel metaheuristic algorithms are often more suitable for solving such complex optimization problems. In order to verify the effectiveness of the improved PPE algorithm in solving practical applications, we apply the MPPE algorithm to this hot problem for the first time. While solving this problem, we also successfully tried a new method, not limited to classic algorithms. This is to open up new ideas for scholars doing this kind of research. In this article, in addition to comparing MPPE with several original algorithms, we also compare with the recently proposed algorithms that have been used to solve the bus scheduling problem. The final experimental results prove that MPPE we proposed can achieve good results in solving this problem.

The main contributions of this article are as follows:

(1) A parallel method is proposed with a two-stage communication strategy

(2) The step size factor of the flower pollen algorithm was introduced into the growth of the stick insect population to achieve the diversity of evolutionary trends

(3) MPPE algorithm has been successfully implemented for the optimization of the IoT electric bus scheduling on rainy days in city, and the result is satisfactory

The remaining sections of the article are organized as follows. Section 2 introduces the relevant formulas of the IoT electric bus scheduling problem and basic principle of PPE. In Section 3, a mutigroup-based Phasmatodea population evolution algorithm with mutistrategy (MPPE) is proposed. In Section 4, we compared MPPE with other algorithms on the benchmark suite and analyzed the performance of this algorithm. In Section 5, MPPE algorithm is applied to the IoT electric bus scheduling. Section 6 concludes the study.

2. Related Work

2.1. IoT Electric Bus Scheduling Problem. The Internet of Things based electric bus under the intelligent bus system is equipped with multisensor, high precision positioning, and wireless communication technology. This multitechnical fusion can detect the passenger flow and the consumption in the course of driving. In addition, passengers at the station can receive real-time bus scheduling information. This includes the estimated arrival time of next bus and the departure time interval for a certain period of time. When the same type of bus operates on the same section of the road, its total mileage and number of stations are fixed, the fare is uniform throughout the journey, and the time to reach each station during the journey is roughly equal. For these basic conditions, we can do the following analysis of the bus scheduling problem in this case.

When only one-way operation of the electric bus is considered, the departure time of the first and last bus is fixed. Then, the day is divided into five time periods; each time period of passenger flow is different. Passenger flow is usually calculated by GPS positioning and IC card records. After the first bus leaves, the bus company should consider the departure interval of the next bus, which is different for each time period. The motivation of the IoT electric bus scheduling is to balance the interests of the company and passengers [39], which makes the setting of the departure interval fully consider the company's profit and the patience of passengers. This is a multiobjective minimization problem. Therefore, the usual approach is to apply a weight to the company benefit and passenger benefit coefficient, respectively. In this case, decision-makers can balance the relationship between these two weights according to the actual situation. The objective function model can be represented by the following equation:

$$F\left(\Delta t_k\right) = w_1 Q_1 D \sum_{k=1}^{K} \frac{T_k}{\Delta t_k} + w_2 Q_2 \sum_{k=1}^{K} \sum_{j=1}^{J} m_k \left(\frac{u_{kj} \Delta t_k^2}{2}\right), \quad (1)$$

where the first item of the equation represents the cost consumption of the company and the second is the loss factor of passengers. w_1 and w_2 represent two weights, and the sum of these two values is equal to 1. Q_1 is the cost of each bus per kilometer distance, including the electricity consumed, the loss of vehicle equipment, and the salary of bus drivers. Q_2 is the patience loss of a unit of time for each passenger waiting at the station. D is the total distance of one-way bus operation, divided into K time periods. k indicates the current time period, T_k is the length of the current time period, Δt_k is the departure interval of the kth time period, J is the total number of sites, j is the number of the current site, m_k is the total number of departures in the kth time period, and u_{kj} is the arrival rate of passengers at station j in the kth time period.

In solving a practical optimization problem, in addition to the objective function, the corresponding constraints should also be considered. The constraint of this bus scheduling problem is that the bus company must be in a profitable state. In other words, the company's operating income should be greater than the cost, to ensure that the company does not lose money. The equation for its constraint is

$$n \times \frac{\sum_{k=1}^{K} \sum_{j=1}^{J} u_{kJ}}{\sum_{k=1}^{K} \left(T_k / \Delta T_k\right) > Q_1 D}. \quad (2)$$

Using the above equation, if the turnover of bus company is below cost, an additional time is added to the bus departure interval as set out in the current system until the turnover is not less than the cost.

According to the above introduction, the complexity of this problem mainly depends on the number of sites and time periods. With a series of data such as cost and passenger flow, this nonlinear problem is dynamic and difficult to solve by conventional methods.

2.2. Phasmatodea Population Evolution Algorithm.

The basic principle of the Phasmatodea population evolution algorithm is that the movement of the solution in multidimensional space simulates the evolutionary trend of stick insect population. In a dynamic environment, each stick insect population is self-determining and is assigned two properties: growth rate and size. The growth trend of population is similar to the logistic regression model. During the initialization phase, solutions x of N_p populations are randomly generated, and the size of each x is computed as follows:

$$p_i = \frac{1}{N_p}. \quad (3)$$

In addition, h historically optimal solutions were selected to lead the search of the other solutions, and the number of h is

$$h = \left[\log\left(N_p\right)\right] + 1. \quad (4)$$

These historically optimal solutions are stored in ho, $ho = [x_{k1}, \ldots, x_{ki}, \ldots, x_{kh}]$. The initialized generated N_p solutions select h local optimal values through ho and store their fitness values. These local optimal solutions will direct other solutions throughout the population.

During the iterative update phase, the evolutionary trend of the population is represented by evt, the position of the t-th evolution of the population is x^t, and then the position obtained by the next evolution is x^{t+1}. The position update equation is

$$x^{t+1} = x^t + evt. \quad (5)$$

After the position is updated, the global optimal values Gbest and h optimal solutions ho of the population are calculated. Evolutionary trend evt is divided into three parts. The first two parts use convergent evolutionary and path-dependent population growth models. If the quality of the updated solution is better than before, the first part of the update is adopted, and the update equation is

$$evt^{t+1} = \left(1 - p^{t+1}\right)A + p^{t+1}\left(evt^t + m\right), \quad (6)$$

where p^{t+1} represents the population size of the next iteration, and its update formula is

$$p^{t+1} = a^{t+1} p^t \left(1 - p^t\right), \quad (7)$$

where a represents growth rate, which is set to 1.1 at the beginning, and A indicates the degree of closest to the optimal value, calculated as follows:

$$A = \left(s\left(ho, x^t\right) - x^t\right) \cdot c, \quad (8)$$

where s (ho, x^t) is to find the closest historically optimal solution x^t in ho. c is a coefficient of the extent to which the most recent optimal solution affects the current population, ranging from 0 to 1. m represents the population mutation in a partial dimension.

If the quality of the solution after the position update is worse, the update method of Part 2 will be used instead of

Part 1. The second part is called convergence evolution, which tends to the most recent local optimal solution. The update equation is

$$evt^{t+1} = \text{rand} \cdot A + st \cdot N, \tag{9}$$

where N is the n-dimensional random factor generated by the normal distribution. The value of st is set to $0.1 \times (Ub - Lb)$.

The third part describes the model of population competition under environmental intervention. When populations x_i and x_j are competing with each other, the population size of x_i is

$$p_{\mathrm{i}} = p_i + a_i p_i \left(1 - p_i - \frac{f(x_j)}{f(x_i)} p_j\right), \tag{10}$$

where x_j is randomly selected and different from x_i. Calculate the distance between x_i and x_j; then compare it to the set threshold TH. TH is calculated as follows:

$$\mathrm{TH} = (Ub - Lb) \times 0.1 \times \frac{\mathrm{MaxGen} + 1 - t}{\mathrm{MaxGen}}, \tag{11}$$

where Ub and Lb represent upper and lower boundaries. MaxGen represents the maximum number of iterations. t is the current iteration. If the distance between x_i and x_j is less than TH, the two populations begin to compete each other, and the evolutionary trend update equation is

$$evt^{t+1} = evt^{t+1} + \frac{f(x_j) - f(x_i)}{f(x_j)} (x_j - x_i). \tag{12}$$

Among them, if the population is too small or it is growing abnormally, the solution will be replaced by a new population.

According to the formulas above, the pseudocode of PPE is shown in Figure 1.

3. Proposed Method

In this study, we propose multigroup-based Phasmatodea population evolution algorithm with multiple strategies, including a two-stage parallel communication strategy and the introduction of step factor, which improve the PPE algorithm in many aspects.

3.1. Parallel Strategy. In the previous research on the improvement of optimization algorithm, the parallel method is often taken into account [40]. The parallel method can effectively improve the global exploration ability of the algorithm through population division. In this study, we divide stick insect populations into multiple groups, make each group evolve in parallel, and implement a two-stage intergroup communication strategy at each iteration. The first stage of communication is to accelerate the convergence of PPE by mutating the worst particles in each group to the best particles in some dimensions. The second stage is to improve the search accuracy of PPE by randomly replacing the best particles with random particles from each group. To balance the convergence speed and global exploration accuracy, we use the first strategy for the first half of the iterations and the second strategy for the second half.

```
Input: $Np$, $d$, $MaxGen$;
Output: $Gbest$ and minimum fitness $fmin$
    Initialize $Np$ population, $p$, $evt$, $h$ using Eq. (3), (4);
    Initialize fitness $f(x)$, $ho$ and $Gbest$;
    for $iter = 2$ to $MaxGen$
        Update $x$ to $xnew$ using Eq. (5);
        Calculate $f(xnew)$, update $Gbest$ and $ho$;
        for $n = 1$ to $Np$
        if $f(x) \geq f(xnew)$ then
            $x = xnew$, update $f(x)$;
            Update $p_i$ using Eq. (7);
            Update $evt$ using Eq. (6), (8);
        else if $rand < p_i$ then
            $x = xnew$, calculate $f(x)$;
            Update $p_i$ using Eq. (7);
        end if
        Update $evt$ using Eq. (9);
        end if
        select $x_j$ randomly, $(i \neq j)$;
        if $distance\ (x_i, x_j) < TH$ then
        Update $p_i$ using Eq. (10), update $evt$ using Eq. (12);
        end if
        end for
    end for
```

Figure 1: The pseudocode of PPE.

In the first communication strategy, the particles with the worst fitness values in each group are mutated individually, and the mutation is close to the current global optimal particles. The equation for the mutation is as follows:

$$X_d = \text{totalbest}_d \times (m + n \times \text{rand}), \tag{13}$$

where X_d represents the position of the particle after the d-dimensional mutation, totalbest_d is the position of the current global optimal particle of the d-dimensional, rand is a random number of 0-1. m and n are parameters that control the degree to which the mutant particle approaches the global optimal particle. In general, $m, n \in (0, 1)$, and the sum of m and n is 1. In order to make mutant particles close to the current global optimal particles, the value of m should be set larger. In this article, we set m to 0.8 and n to 0.2. In this way, the positions of mutated particles are randomly distributed near the current global optimal position.

In the second communication strategy, some random particles are replaced with the current global optimal particles. The specific operation is to randomly generate indexes of the positions of some particles ranging from 1 to the total number of particles per group, and then replace these particles with the current global optimal particles in each group corresponding to these indexes. Finally, the fitness values of the replaced particles are updated.

Figure 2 shows the above introduction of proposed two-stage parallel communication strategy. This parallel strategy combines two different methods, which leads to closer communication between groups. Therefore, what we expect is to make the algorithm more stable in the process of search.

3.2. Step Factor of the Flower Pollen Algorithm. The flower pollen algorithm simulates the pollination of flowering plants [41–43], which is divided into global pollination and local pollination. This study was inspired by global pollination, which simulates the process by a flying insect carries pollen from one flowering plant over a distance to another. The equation of pollen position movement for global pollination is

$$x_i^{t+1} = x_i^t + \text{Levy} \times \left(x_i^t - g^t\right), \tag{14}$$

where g^t is the current global optimum. Levy represents the flight of flying insects in global pollination, which is calculated as follows:

$$\text{Levy} = \frac{\sin(\pi r/2) \times \lambda\Gamma(r)}{\pi \times s^{i+r}}, \tag{15}$$

where $\Gamma(r)$ is the gamma function, the distribution is valid for the large steps of $s > 0$. In this article, we set r to 1.5. As the position index i changes, so does the value of Levy, which indicates the degree of influence of global pollination can be seen as a step factor. In order to allow the population of stick insect to evolve with different growth trends and prevent algorithms from tending to locally optimal solutions when solving some optimization problems, we add the step factor Levy to the population growth model of stick insect in some groups. This method mainly changes the degree of influence of convergence evolution in (6), so that the original influence factor c was replaced by a changing step factor Levy. The equation after the change of A is

$$A = \left(s\left(ho, x^t\right) - x^t\right) \cdot \text{Levy} \cdot \gamma, \tag{16}$$

where γ indicates the scaling factor that controls step Levy. If the value of Levy is too large, it will inevitably break the evolutionary rule of the population, at which point the γ will narrow the Levy when it exceeds a certain range.

3.3. Multigroup-Based Phasmatodea Population Evolution Algorithm with Multistrategy. According to the introduction of the above two-stage communication strategy and step factor of the flower pollen algorithm, we propose the multigroup-based Phasmatodea population evolution algorithm with multistrategy and named it as MPPE. The flowchart for this method is shown in Figure 3. Figure 4 shows the pseudocode of MPPE. In the initialization stage, the stick insect population is divided into several groups, and each group has the same population size. Then, the initial solution and position of each group were generated according to the initialization method of PPE. In the update iteration stage, no more than half of the groups are introduced into the population growth model with step factor *Levy*, and the remaining groups are updated in the original method. After each group completes an iteration update at the same time, the two-stage intergroup communication strategy described above will be carried out according to the number of iterations. The solution obtained after each intergroup communication will be the current global optimal solution.

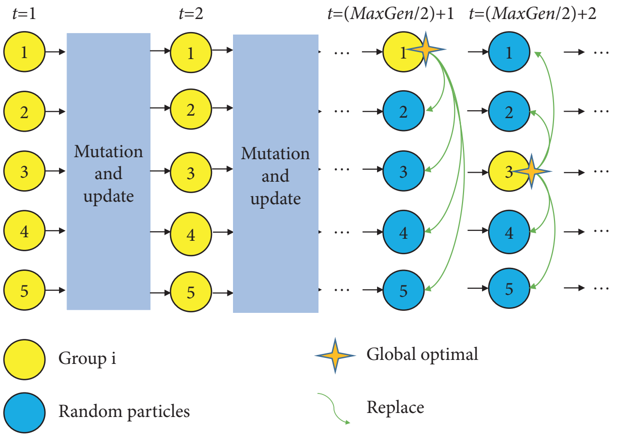


Figure 2: Two-stage communication strategy.

This research is based on a parallel method, and multiple strategies are implemented on the basis of multiple groups in parallel. The diversity of population evolution trends can be used to find the best particles through communication between groups. In general, the limitations of a single strategy cannot achieve the effect of algorithm improvement. Only the combination of multiply strategies can work together.

4. Experiment and Discussion

To test the optimization effect of MPPE, we conduct two comparative experiments. The first set of experiments is compared with PPE algorithm and five standard metaheuristic algorithms. The second is compared with other parallel metaheuristic algorithms. In this article, we select CEC 2013 Benchmark Suite to test the optimization problems of all algorithms. CEC 2013 Benchmark Suite is composed of 28 classic functions and has been widely used in experiments to test the performance of algorithms, where f_1~f_5 are single-mode functions, f_6~f_{20} are basic multimode functions, and f_{21}~f_{28} are combination functions. These optimization problems have a common search range of −100 to 100, with minimum values increasing from −1400 to 1400.

All experiments in this article are completed in MATLAB R2020b. The experimental equipment is a laptop computer with 2.60 GHz CPU, 16 g memory, and 64-bit Win10 system. Generally speaking, a larger population size and number of iterations can make the optimization result of the algorithm better. To ensure fair comparison and fully reflect the performance of these algorithms, we set the population of all algorithms to 100, the dimension to 30, and the number of function evaluations to 200,000. Each algorithm is independently experimented with 20 averages as an optimization result.

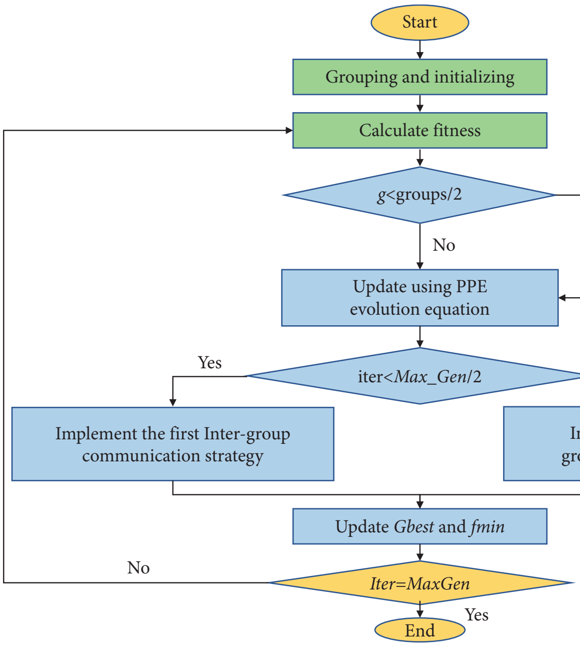

Start
Grouping and initializing
Calculate fitness
g<groups/2
Yes
Add step factor Levy to the next step
No
Update using PPE evolution equation
Yes
iter<Max_Gen/2
No
Implement the first Inter-group communication strategy
Implement the second Inter-group communication strategy
Update Gbest and fmin
No
Iter=MaxGen
Yes
End

FIGURE 3: The entire phases of the proposed method.

These experiments mainly compare the exploration accuracy and convergence performance of algorithms and implemented appropriate nonparametric hypothesis tests [44, 45]. In order to further analyze the proposed algorithm, we discuss the time complexity of MPPE in the last part of this section.

4.1. Compared to Standard Algorithms. In this part, we compared MPPE with PPE, PSO [46], SCA [47], WOA [48], GWO [49], and GSK [10]. Table 1 lists the parameter settings for these algorithms. Table 2 shows the comparison results in 28 benchmark functions. It shows that MPPE can achieve satisfactory results in most optimization problems. Among them, the optimal value obtained by the optimization of f_1, f_5, and f_{10} are their actual minimum value. In comparison with PPE, except for f_1 and f_5, which have achieved the minimum of the function together, optimal values are better in 18 functions and equal in 7 functions, and only f_{13} is worse. In addition, the optimization effects of MPPE on single-mode problems and multimode problems are significantly improved. It can be inferred that these improvement strategies make the algorithm no longer tend to the local optimal solution when solving most problems. Compared with PSO, the optimization effect of PSO is not worse than that of MPPE in some functions. However, as a classic swarm intelligence algorithm, PSO has been recognized as a drawback of premature convergence, so it is easier to tend to local optimal solutions when solving practical problems. For example, its optimization effects in f_4, f_{15}, and f_{23} are far behind that of MPPE, and its stability is not good. In comparison with SCA and WOA, MPPE has great advantages, with better results in almost all function optimizations than these two algorithms. In the comparison of GWO, MPPE algorithm is superior in the optimization of most single-mode and multimode problems, but poor in the optimization of combinatorial problems. This is because GWO algorithm has a better result in solving these problems from its clustering cooperation mechanism. Overall, MPPE is better than GWO at solving most problems. In comparison with GSK, it can be seen that the overall optimization effect of GSK is slightly better than that of MPPE. But in the optimization of some multimode problems, MPPE performs significantly better.

The comparison of these seven algorithms shows that MPPE can achieve better or equal results than other six algorithms in the optimization results of 14 functions. Table 3 counts the number of the best results obtained by each of the seven algorithms on three types of functions and their rankings. As can be seen from the rankings, the optimization performance of MPPE in this benchmark suite is second only to GSK. MPPE is overall better than other five classic algorithms. It should be emphasized that MPPE performs best in multimode functions.

Table 4 shows the results of Wilcoxon signed rank-sum test between MPPE and other six algorithms at a significant level of 0.05. The "+" symbol in the table indicates the optimization results in 28 functions of MPPE is better than that algorithm. The meaning of the "−" symbol is the optimization results of MPPE is worse than that algorithm. The "≈" symbol indicates that the optimization results almost have no difference. R^+ and R^- both represent the sum of ranks calculated by comparing all differences. p value records the significance of MPPE compared to each algorithm. "Sig." represents the significance level of MPPE compared to each algorithm. It can be concluded that the optimization results of MPPE is only significantly better than SCA, and there is no significant difference from the other five algorithms. From the perspective of rank-sum, in the comparison between MPPE and PPE, SCA, and WOA, R^+ is

```
Input: $Np, d, MaxGen, groups$;
Output: $Gbest$ and minimum fitness $fmin$
Initialize Np population, $evt, p, h$ of each group using Eq. (3), (4);
Initialize fitness $f(x)$, $ho$, $Gbest$ of each group;
for $iter = 2$ to $MaxGen$
    for $g = 1$ to $groups$ do
        Update $x$ to $xnew$ using Eq. (5);
            Calculate $f(xnew)$, $ho$, $Gbest$;
        for $i = 1$ to $Np$
            if $f(x) \geq f(xnew)$ then
                $x = xnew$, update $f(x)$;
                Update $p_i$ using Eq. (7);
                If $g \leq groups/2$ then
                    Update $evt$ using Eq. (6), (8);
                else Update $evt$ using Eq. (6), (16);
                end if
            else if $rand < p_i$ then
                $x = xnew$, update $f(x)$;
                Update $p_i$ using Eq. (7);
            end if
                Update $evt$ using Eq. (9);
            end if
                select $x_j$ randomly, $(i \neq j)$;
            if $distance(x_i, x_j) < TH$ then
                Update $p_i$ using Eq. (8), $evt$ using Eq. (12);
            end if
        end for
    end for
    for $g = 1$ to $groups$ do
        Compare, find $Gbest$ and $fmin$;
    end for
    for $g = 1$ to $groups$ do
        if $iter < MaxGen/2$ do
            Use communication strategy 1 and Eq. (13) in each group;
            else use communication strategy 2 in each group;
        end if
    end for
end for
```

Figure 4: The pseudocode of MPPE.

significantly greater than R^-. Compared with GSK, R^+ is worse than R^-.

Figure 5 shows the convergence performance of seven algorithms in different benchmark functions. We select the experimental results of two single-mode functions, two multimode functions, and two combination functions to compare the convergence performance of these algorithms from multiple perspectives. To make the comparison more obvious, we set the minimum fitness ordinate to log scale. It shows that MPPE has a fast convergence speed and is not prone to premature convergence. Compared with PPE, the convergence curve of MPPE has a faster downward trend during the early period, and the convergence in the later period improves the search accuracy more obviously. The convergence speeds of GWA, WOA, and SCA in the optimization of these functions are slow. The convergence performance of PSO is unstable. GSK converges better than MPPE in the optimization of f_2 but falls into the local optimal solution early in the optimization of f_{14}. Therefore, MPPE's comprehensive convergence performance in the optimization of these functions is better than the other six algorithms.

Table 1: Parameter settings of seven algorithms.

Algorithm	Parameter settings
MPPE	$N_p = 100$, groups $= 5$
PPE	$N_p = 100,\ c = 0.2$
PSO	$N_p = 100$, $c_1 = 2$, $c_2 = 2$, inertia constant from 0.8 to 0.2
SCA	$N_p = 100$
WOA	$N_p = 100$
GWO	$N_p = 100$
GSK	$N_p = 100,\ k_f = 0.5,\ k_f = 0.9,\ p = 0.1,\ K = 10$

4.2. Compared to Parallel Algorithms. To further verify the overall performance of MPPE, three parallel algorithms are selected to compare with this algorithm. These algorithms are PPSO [50], PWOA [51], and MMSCA [52]. The specific parameter settings are shown in Table 5. These algorithms were also compared on 28 functions of CEC 2013 and the experimental results of MPPE remain unchanged. Table 6 shows the optimization results of the four algorithms. MPPE can achieve the best results in the optimization of 20 functions, with multimode functions accounting for a large proportion. The results of PWOA and MMSCA are worse than MPPE in most functions, but PPSO can achieve better optimization results in 7 functions. Though the intergroup communication strategy of PPSO can avoid premature convergence to a certain extent, it still has the drawback of decreased accuracy when solving complex problems. Therefore, its overall optimization performance is worse than MPPE.

Table 7 shows the statistics of optimization results of four algorithms on three types of functions. MPPE ranks first with the most wins compared to the other three parallel algorithms. Especially in the optimization results of multimode and combination problems, MPPE occupies a significant advantage.

Table 8 shows the results of Wilcoxon signed rank-sum test between MPPE and the other three algorithms. In these sets of statistical results, the advantages of MPPE for the other three parallel algorithms are not significant. Therefore, we can at least infer that in this benchmark suite test, there is no obvious gap in the optimization results of MPPE compared to most algorithms under the parameter settings of this experiment.

Figure 6 shows convergence performance of the four parallel algorithms. We select three multimode and combination functions, respectively, for analysis. From the comparison of convergence speed, both MMSCA and PWOA are significantly slower than MPPE. PPSO and MPPE are similar in the early stage, but PPSO has a slower

Table 2: Optimization results for MPPE, PPE, PSO, SCA, WOA, GWO, and GSK over CEC 2013.

Function	MPPE	PPE	PSO	SCA	WOA	GWO	GSK
f_1	**−1.40E + 03**	**−1.40E + 03**	**−1.40E + 03**	1.13E + 04	**−1.40E + 03**	−6.66E + 02	**−1.40E + 03**
f_2	2.61E + 06	4.51E + 06	1.84E + 07	1.47E + 08	3.84E + 07	1.66E + 07	**2.57E + 05**
f_3	3.62E + 08	7.31E + 08	1.32E + 08	3.86E + 10	1.75E + 10	4.38E + 09	**−1.20E + 03**
f_4	**−2.06E + 02**	3.36E + 03	6.75E + 04	3.45E + 04	5.27E + 04	2.77E + 04	6.43E + 02
f_5	**−1.00E + 03**	**−1.00E + 03**	**−1.00E + 03**	1.45E + 03	−8.91E + 02	−2.95E + 02	**−1.00E + 03**
f_6	−8.33E + 02	−8.33E + 02	−8.20E + 02	−9.17E + 01	−7.74E + 02	−7.64E + 02	**−8.85E + 02**
f_7	−7.20E + 02	−6.95E + 02	−7.57E + 02	−6.07E + 02	−1.61E + 02	−7.49E + 02	**−8.00E + 02**
f_8	**−6.79E + 02**	**−6.79E + 02**	**−6.79E + 02**	**−6.79E + 02**	**−6.79E + 02**	**−6.79E + 02**	**−6.79E + 02**
f_9	−5.69E + 02	−5.69E + 02	−5.78E + 02	−5.60E + 02	−5.62E + 02	**−5.81E + 02**	**−5.63E + 02**
f_{10}	**−5.00E + 02**	−4.99E + 02	−4.99E + 02	1.21E + 03	−4.13E + 02	−2.65E + 02	**−5.00E + 02**
f_{11}	**−3.87E + 02**	−3.52E + 02	−3.77E + 02	−4.18E + 01	1.04E + 02	−3.08E + 02	−2.66E + 02
f_{12}	**−2.76E + 02**	−4.46E + 00	−2.20E + 02	8.75E + 01	2.10E + 02	−1.77E + 02	−1.30E + 02
f_{13}	1.29E + 02	1.79E + 02	1.72E + 02	1.91E + 02	2.96E + 02	−1.98E + 01	**−2.88E + 01**
f_{14}	**9.02E + 02**	1.15E + 03	2.59E + 03	7.06E + 03	5.07E + 03	2.93E + 03	5.93E + 03
f_{15}	**3.46E + 03**	4.22E + 03	7.13E + 03	7.63E + 03	5.61E + 03	3.80E + 03	7.38E + 03
f_{16}	**2.01E + 02**	**2.01E + 02**	2,02E + 02	2.03E + 02	2.02E + 02	2.03E + 02	2.02E + 02
f_{17}	**3.38E + 02**	3.79E + 02	3.62E + 02	7.84E + 02	9.25E + 02	4.54E + 02	4.74E + 02
f_{18}	6.23E + 02	6.74E + 02	6.32E + 02	9.04E + 02	9.73E + 02	6.45E + 02	**5.80E + 02**
f_{19}	**5.03E + 02**	5.07E + 02	**5.03E + 02**	3.95E + 03	5.67E + 02	2.03E + 03	5.13E + 02
f_{20}	6.15E + 02	6.15E + 02	6.15E + 02	6.14E + 02	6.15E + 02	**6.12E + 02**	6.13E + 02
f_{21}	1.09E + 03	1.09E + 03	**1.00E + 03**	2.61E + 03	1.09E + 03	1.54E + 03	**1.00E + 03**
f_{22}	**2.24E + 03**	2.48E + 03	5.05E + 03	8.58E + 03	7.12E + 03	3.91E + 03	5.04E + 03
f_{23}	5.67E + 03	5.69E + 03	5.77E + 03	8.83E + 03	7.62E + 03	**5.52E + 03**	8.02E + 03
f_{24}	1.28E + 03	1.28E + 03	1.29E + 03	1.32E + 03	1.31E + 03	1.25E + 03	**1.20E + 03**
f_{25}	1.42E + 03	1.44E + 03	1.41E + 03	1.43E + 03	1.42E + 03	**1.37E + 03**	1.41E + 03
f_{26}	**1.40E + 03**	1.47E + 03	1.48E + 03	1.41E + 03	1.53E + 03	1.53E + 03	**1.40E + 03**
f_{27}	2.42E + 03	2.28E + 03	2.29E + 03	2.70E + 03	2.63E + 03	**2.09E + 03**	**1.67E + 03**
f_{28}	2.52E + 03	4.69E + 03	3.47E + 03	4.04E + 03	5.41E + 03	2.63E + 03	**1.70E + 03**

Bold values represent the best results relative to other comparison methods.

Table 3: Statistics of optimization results of seven algorithms on three types of functions.

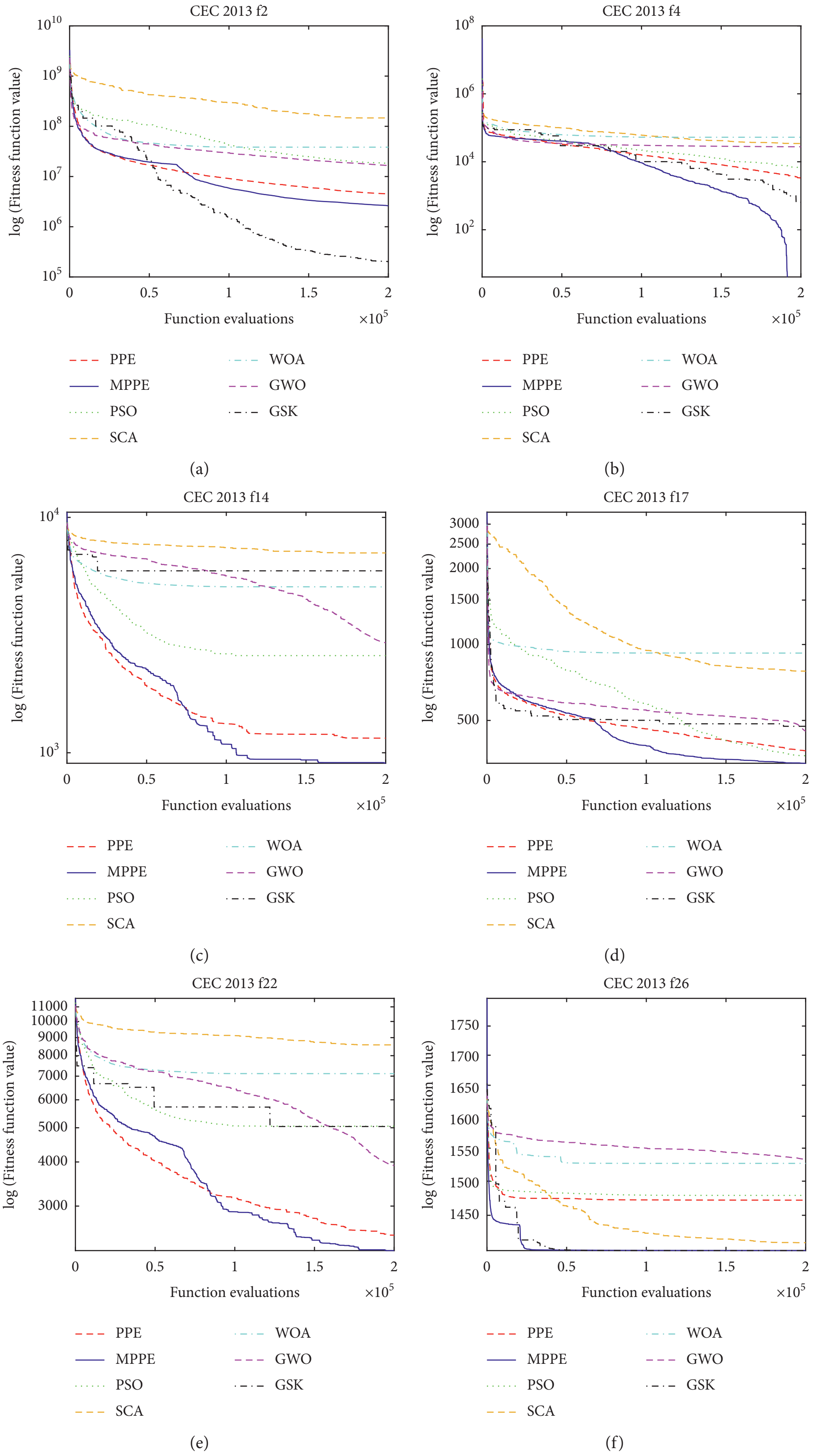
Algorithm	Single-mode	Multimode	Combination	Wins	Ranking
GSK	4	7	5	16	1
MPPE	3	9	2	14	2
GWO	0	3	3	6	3
PSO	2	2	1	5	4
PPE	2	2	0	4	5
WOA	1	1	0	2	6
SCA	0	1	0	1	7

Table 4: The results of Wilcoxon signed rank-sum test between MPPE and other six algorithms.

Algorithm	R^+	R^-	p value	+	≈	−	Sig.
MPPE versus PPE	340	21	0.63	18	9	1	≈
MPPE versus PSO	254	124	0.59	14	7	7	≈
MPPE versus SCA	400	0	0.035	24	3	0	+
MPPE versus WOA	391	0	0.33	23	5	0	≈
MPPE versus GWO	307	93	0.46	17	3	8	≈
MPPE versus GSK	172	206	0.81	9	7	12	≈

convergence rate in the late period. We can infer from this that MPPE is more likely to get rid of the local optimal solution and continue to seek the global minimum in the late period. Therefore, its overall convergence performance is better than the other three parallel algorithms.

4.3. Analysis of Algorithm Complexity. The maximum time complexity of PPE is described as $O\ (p \cdot n \cdot I)$, where p is the total number of stick insect populations, n is the dimension of the problem, and I is the maximum number of iterations. MPPE adopted a parallel method on the basis of PPE, and each



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5: Convergence performance of seven algorithms. (a) f_2, (b) f_4, (c) f_{14}, (d) f_{17}, (e) f_{22}, and (f) f_{26}.

Table 5: Parameter settings of four parallel algorithms.

Algorithm	Parameter settings
MPPE	$N_p = 100$, groups = 5
PPSO	$N_p = 100$, groups = 5, $c_1 = 2$, $c_2 = 2$, inertia constant from 0.8 to 0.2,
PWOA	$N_p = 100$, groups = 5
MMSCA	$N_p = 100$, groups = 5

Table 6: Optimization results for MPPE, PPSO, PWOA, and MMSCA over CEC 2013.

Function	MPPE	PPSO	PWOA	MMSCA
f_1	**−1.40E + 03**	**−1.40E + 03**	−1.39E + 03	6.77E + 03
f_2	2.61E + 06	**8.09E + 05**	3.71E + 07	7.42E + 07
f_3	**3.62E + 08**	6.52E + 08	1.31E + 10	2.02E + 10
f_4	−2.06E + 02	**−7.43E + 02**	4.75E + 04	2.81E + 04
f_5	**−1.00E + 03**	**−1.00E + 03**	−8.88E + 02	6.56E + 02
f_6	−8.33E + 02	**−8.43E + 02**	−7.71E + 02	−4.33E + 02
f_7	**−7.20E + 02**	−7.08E + 02	−5.93E + 02	−6.68E + 02
f_8	**−6.79E + 02**	**−6.79E + 02**	**−6.79E + 02**	**−6.79E + 02**
f_9	−5.69E + 02	**−5.72E + 02**	−5.67E + 02	−5.62E + 02
f_{10}	**−5.00E + 02**	**−5.00E + 02**	−3.41E + 02	7.27E + 02
f_{11}	**−3.87E + 02**	−1.01E + 02	−3.99E + 01	−6.81E + 01
f_{12}	**−2.76E + 02**	−4.21E + 01	1.90E + 02	3.62E + 01
f_{13}	**1.29E + 02**	1.49E + 02	2.30E + 02	1.51E + 02
f_{14}	**9.02E + 02**	3.39E + 03	3.78E + 03	6.74E + 03
f_{15}	**3.46E + 03**	4.22E + 03	5.69E + 03	7.21E + 03
f_{16}	**2.01E + 02**	**2.01E + 02**	2.02E + 02	2.02E + 02
f_{17}	**3.38E + 02**	4.81E + 02	8.25E + 02	7.39E + 02
f_{18}	6.23E + 02	**5.61E + 02**	7.88E + 02	8.55E + 02
f_{19}	**5.03E + 02**	5.06E + 02	5.35E + 02	3.22E + 03
f_{20}	6.15E + 02	6.15E + 02	6.15E + 02	**6.14E + 02**
f_{21}	1.09E + 03	**1.03E + 03**	1.18E + 03	2.33E + 03
f_{22}	**2.24E + 03**	5.61E + 03	5.12E + 03	7.56E + 03
f_{23}	**5.67E + 03**	6.22E + 03	6.53E + 03	8.58E + 03
f_{24}	**1.28E + 03**	1.29E + 03	1.30E + 03	1.31E + 03
f_{25}	**1.42E + 03**	1.43E + 03	**1.42E + 03**	**1.42E + 03**
f_{26}	**1.40E + 03**	1.51E + 03	1.48E + 03	1.41E + 03
f_{27}	2.42E + 03	**2.40E + 03**	2.66E + 03	2.62E + 03
f_{28}	**2.52E + 03**	3.86E + 03	5.09E + 03	3.74E + 03

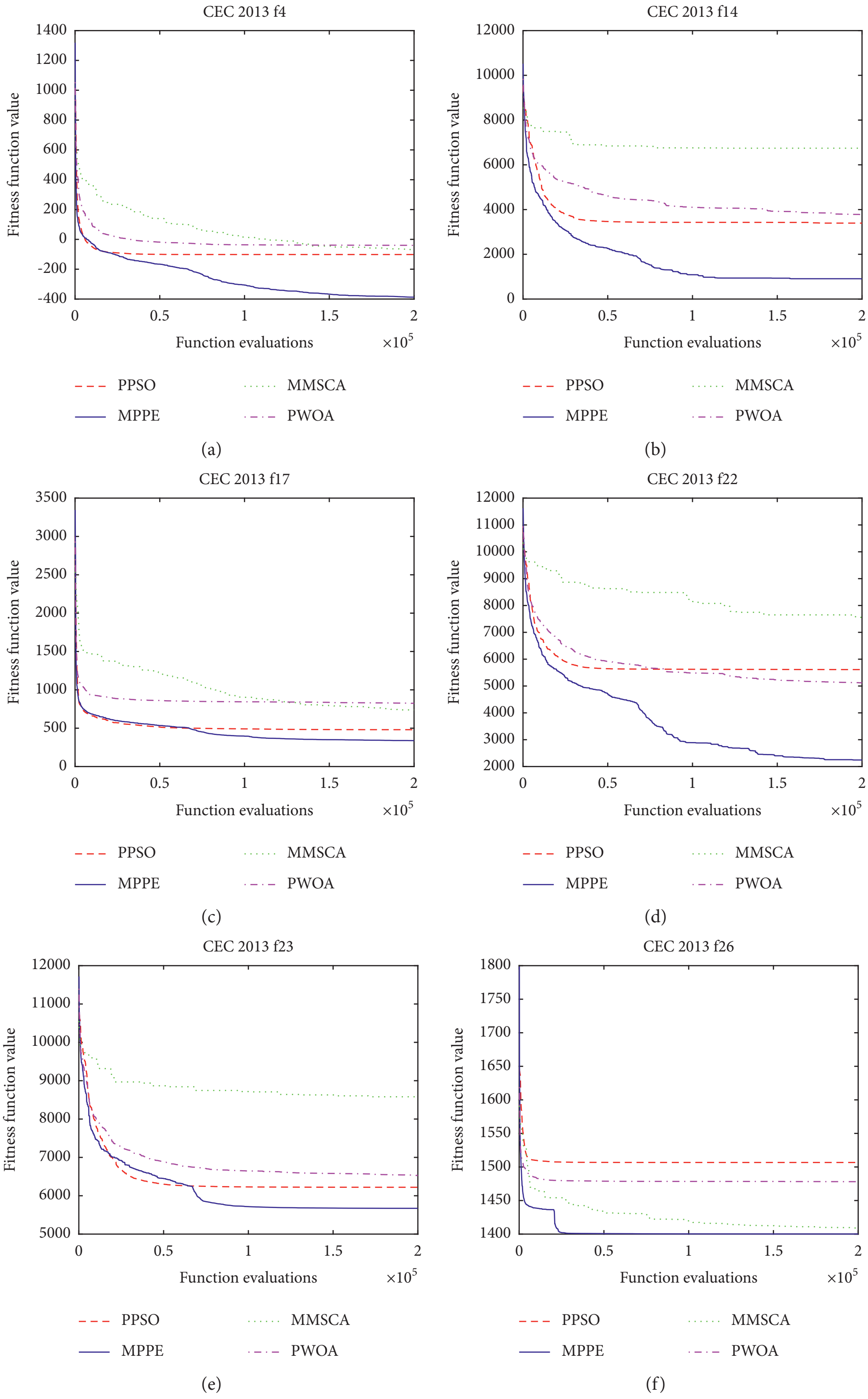
Bold values represent the best results relative to other comparison methods.

Table 7: Statistics of optimization results of four algorithms on three types of functions.

Algorithm	Single-mode	multimode	Combination	Wins	Ranking
MPPE	3	11	6	20	1
PPSO	4	6	2	12	2
MMSCA	0	2	1	3	3
PWOA	0	1	1	2	4

Table 8: The results of Wilcoxon signed rank-sum test between MPPE and the other three algorithms.

Algorithm	R^+	R^-	p value	+	=	−	Dec.
MPPE versus PPSO	245	125	0.87	13	8	7	≈
MPPE versus PWOA	391	0	0.39	23	5	0	≈
MPPE versus MMSCA	396	0	0.067	24	4	0	≈



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 6: Convergence performance of four parallel algorithms. (a) f_4, (b) f_{14}, (c) f_{17}, (d) f_{22}, (e) f_{23}, and (f) f_{26}.

Table 9: Passenger flow in different time periods.

Period of time	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
6:00–9:00	401	101	317	339	106	11	10	9	20	12
9:00–12:00	410	65	107	154	117	86	151	107	33	8
12:00–16:00	177	32	50	28	98	100	138	126	10	16
16:00–19:00	144	72	154	254	177	138	162	250	90	23
19:00–21:00	70	62	35	48	47	9	41	33	11	8

Table 10: Comparison of results in $w_1 = 0.7$ and $w_2 = 0.3$.

Algorithm	Average result	Best result	Worst result	Scheduling scheme
SCA	2920.64	2893.12	2995.92	(2.3 2.7 5.5 2.8 3.5)
MCSCA	2944.07	2878.63	2995.92	(2.1 3.5 7.4 5.1 3.2)
PSO	2968.77	2895.50	3180.31	(3.1 5.5 1.6 3.0 3.3)
PPE	2868.47	2866.52	2873.99	(2.4 3.1 5.6 3.3 5.0)
MPPE	**2866.75**	**2865.77**	**2868.36**	(2.5 3.1 5.6 3.3 4.6)

Bold values represent the best results relative to other comparison methods.

Table 11: Comparison of results in $w_1 = 0.5$ and $w_2 = 0.5$.

Algorithm	Average result	Best result	Worst result	Scheduling scheme
SCA	3238.80	3176.37	3288.67	(2.9 4.3 4.2 3.1 3.2)
MCSCA	3291.97	3184.37	3296.17	(2.0 3.6 5.0 3.7 3.3)
PSO	3905.68	3151.46	5526.75	(8.2 8.2 7.9 1.5 9.6)
PPE	3139.95	3138.58	3141.18	(2.5 3.0 4.6 3.2 4.0)
MPPE	**3138.57**	**3137.83**	**3139.24**	(1.6 2.0 3.8 2.1 3.2)

Bold values represent the best results relative to other comparison methods.

Table 12: Comparison of results in $w_1 = 0.3$ and $w_2 = 0.7$.

Algorithm	Average result	Best result	Worst result	Scheduling scheme
SCA	3102.84	2912.69	3382.02	(2.2 3.3 5.0 3.6 3.2)
MCSCA	3129.38	2937.53	3382.02	(1.4 1.8 2.5 2.6 3.2)
PSO	4269.70	2925.66	6325.52	(8.6 9.5 7.8 3.7 3.6)
PPE	2884.01	2883.21	2885.38	(3.0 3.3 4.4 3.4 3.9)
MPPE	**2882.87**	**2882.41**	**2883.75**	(2.2 2.5 3.5 2.7 3.2)

Bold values represent the best results relative to other comparison methods.

iteration is accompanied by the simultaneous evolution of each group of populations. So, the maximum time complexity of MPPE can be described as $O\ ((p_1 \cdot n + p_2 \cdot n + \cdots + p_g \cdot n) \cdot I)$, where p_i is the population size of the ith group and g is the number of groups. Since the number of populations in each group is equal to p/g, the maximum complexity of MPPE is described as $O\ (p \cdot n \cdot I)$.

Although the amount of calculation added by the intergroup communication strategy does not exceed the maximum time complexity, the actual running time of MPPE has increased considerably compared to PPE.

5. MPPE Applicating in IoT Electric Bus Scheduling

In this optimization algorithm applied to the bus scheduling experiment for urban waterlogging situation, the bus operating time is set to 6 am–9 pm, the total number of kilometers one way to 10 km, and the number of stations to 10; each site is named S_1 to S_{10}. The data simulation of passenger flow is shown in Table 9.

In this experiment, MPPE was compared with PPE, PSO, SCA, and MCSCA [53]. In the search process, all algorithms find their current optimal solutions through each iteration. The optimal solution is the minimum value of the objective function of IoT electric bus scheduling. The position corresponding to the best solution is the time interval of bus departure in each time period. Therefore, these algorithms are all feasible in solving this problem, but their results are usually different.

In the parameter settings, both Q_1 and Q_2 are set to 1. The maximum number of iterations for each algorithm is set to 100. The population size of all algorithms is set to 30. This is because in solving practical problems, it is necessary to set a small number of iterations and populations to improve efficiency. The value of the bus departure time interval in each time period is accurate to one decimal place, and their range is controlled in 0–10. All the algorithms are run 10 times in this experiment. Finally, we calculate the average result, worst result, and best result of these algorithms. To make the experimental results more convincing, we set the weights w_1 and w_2 to three different groups of values. They are as follows: (1) $w_1 = 0.7$ and $w_2 = 0.3$, (2) $w_1 = 0.5$ and $w_2 = 0.5$, and (3) $w_1 = 0.3$ and $w_2 = 0.7$.

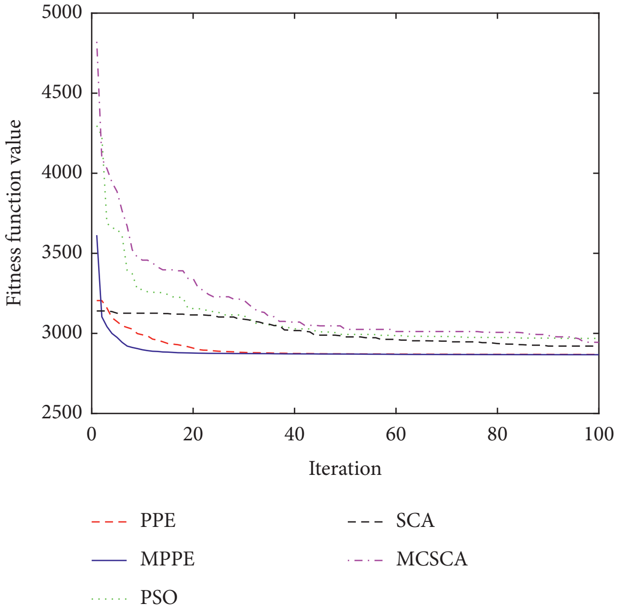


Figure 7: Convergence performances in $w_1 = 0.7$ and $w_2 = 0.3$.

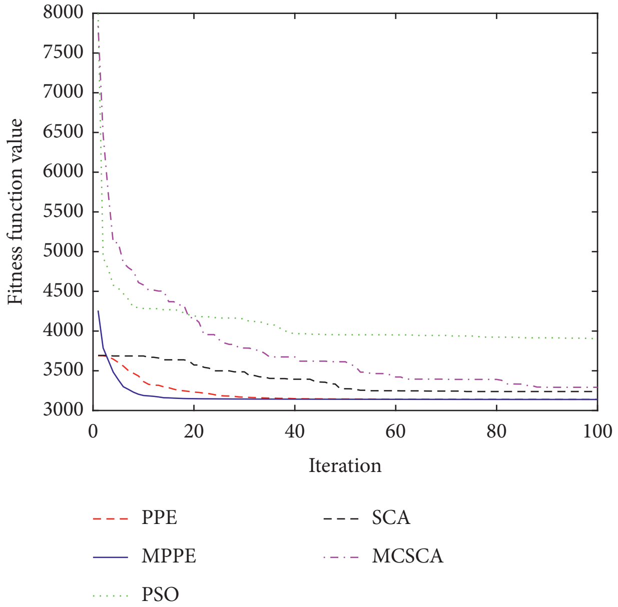


Figure 8: Convergence performances in $w_1 = 0.5$ and $w_2 = 0.5$.

Tables 10 to 12 show the comparison of fitness values and solutions of these algorithms under different weights. Figures 7 to 9 show the convergence performance of these algorithms.

The above three sets of experiments show that the MPPE can perform better in different weight selection, and its average results are better than the other three algorithms in all three cases. From the point of view of the optimal value, although the three results of MPPE are slightly smaller than PPE, MPPE also has better stability. Compared with MPPE, PSO, MCSCA, and SCA all have a big gap. In terms of convergence performance, MPPE has a faster convergence rate in the early stage. Compared to PPE, MPPE almost find the optimal value before 20 iterations, and PPE tends to end convergence in the middle of iterations. Although PSO also has a fast convergence rate in the early stage, it does not perform well in the later stage and tends to fall into local optimum.

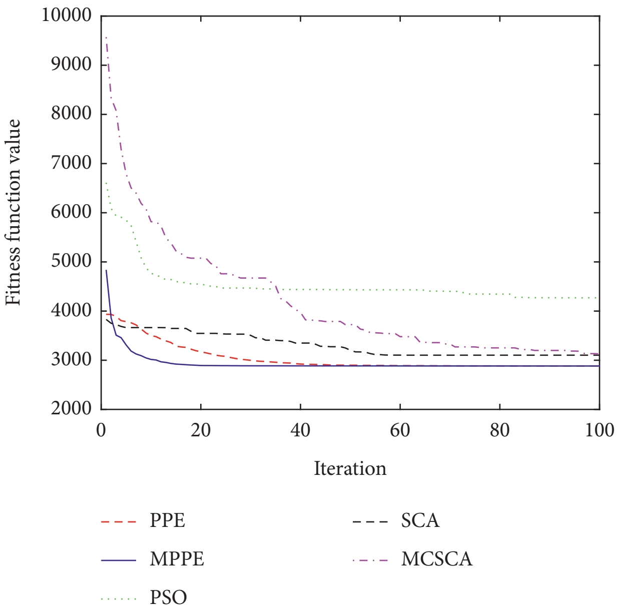


Figure 9: Convergence performances in $w_1 = 0.3$ and $w_2 = 0.7$.

Table 13: Ranking of average optimization results of MPPE under different weight combinations.

Weight combinations	Average result	Scheduling scheme	Ranking
$w_1 = 0.1$, $w_2 = 0.9$	1861.91	(5.1 6.0 9.6 6.7 8.6)	1
$w_1 = 0.9$, $w_2 = 0.1$	1890.99	(2.7 2.9 3.4 2.9 3.2)	2
$w_1 = 0.8$, $w_2 = 0.2$	2494.98	(3.3 3.9 7.3 4.2 6.7)	3
$w_1 = 0.2$, $w_2 = 0.8$	2518.76	(2.5 2.7 3.5 3.8 3.2)	4
$w_1 = 0.7$, $w_2 = 0.3$	2866.75	(2.5 3.1 5.6 3.3 4.6)	5
$w_1 = 0.3$, $w_2 = 0.7$	2882.87	(2.2 2.5 3.5 2.7 3.2)	6
$w_1 = 0.6$, $w_2 = 0.4$	3070.65	(1.9 2.4 4.6 2.6 3.8)	7
$w_1 = 0.4$, $w_2 = 0.6$	3079.02	(2.1 2.4 3.8 2.6 3.2)	8
$w_1 = 0.5$, $w_2 = 0.5$	3138.57	(1.6 2.0 3.8 2.1 3.2)	9

In summary, the MPPE algorithm can effectively solve the IoT electric bus rainy days scheduling problem. This is mainly reflected in the fast convergence speed and good optimization accuracy. The result of the solution is also relatively stable, and it is not prone to large deviations. From the perspective of people's experience, the scheduling scheme optimized by this algorithm is also reasonable.

In order to analyze the relationship between the weight setting and the optimization result, we only use the MPPE algorithm to solve this problem and test all possible weight combinations. Table 13 lists the rankings of the optimization results of all weight combinations and their corresponding scheduling schemes. The experimental results show that the theoretically best combination of weights is $w_1 = 0.1$ and $w_2 = 0.9$, because they achieve the best average results. However, decision-makers should make corresponding adjustments to the setting of weights based on actual conditions.

6. Conclusions

According to the limitations of the PPE algorithm, this research made improvements based on multiple groups and multiple strategies. The experiments of the benchmark CEC 2013 proved that our proposed MPPE algorithm has good convergence speed and optimization accuracy. In addition, we apply this algorithm to the IoT electric bus scheduling problem and achieve a smaller loss value compared to PPE, PSO, SCA, and MCSCA algorithm.

However, the MPPE algorithm still has shortcomings, which is mainly reflected in its long actual running time. In future research, we will consider this shortcoming and make further improvements. At the same time, some representative novel metaheuristic algorithms, such as MBO, SMA, GSK, and HHO, can also be used to solve optimization problems similar to bus scheduling. In the next step, we will consider combining these novel algorithms and expanding this type of optimization problem.

Data Availability

Relevant data used in this study are included in this article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

Acknowledgments

The authors are grateful for the financial support received from the National Key Project of China (no. 2020AAA0109300), the Open Research Project of Shanghai Key Laboratory of Information Security Integrated Management Technology (no. AGK2019004), and the Research on Intelligent Scheduling of Urban Waterlogging Emergency Facilities Based on Multisource Information (no. 21511103704).

References

[1] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Metaheuristic algorithms: a comprehensive review," *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*, Elsevier, Amsterdam, Netherlands, 2018.

[2] M. A. Al-Betar, M. A. Awadallah, I. Abu Doush, A. I. Hammouri, M. Mafarja, and Z. A. A. Alyasseri, "Island flower pollination algorithm for global optimization," *The Journal of Supercomputing*, vol. 75, no. 8, pp. 5280–5323, 2019.

[3] B. H. Abed-alguni, N. A. Alawad, M. Barhoush, and R. Hammad, "Exploratory cuckoo search for solving single-objective optimization problems," *Soft Computing*, vol. 25, pp. 1–14, 2021.

[4] S. Zhou, L. Xing, X. Zheng, N. Du, L. Wang, and Q. Zhang, "A self-adaptive differential evolution algorithm for scheduling a single batch-processing machine with arbitrary job sizes and release times," *IEEE Transactions on Cybernetics*, vol. 51, no. 3, pp. 1430–1442, 2021.

[5] B. H. Abed-alguni and N. A. Alawad, "Distributed Grey Wolf Optimizer for scheduling of workflow applications in cloud environments," *Applied Soft Computing*, vol. 102, Article ID 107113, 2021.

[6] S. Tiachacht, S. Khatir, C. Le Thanh, R. V. Rao, S. Mirjalili, and M. A. Wahab, "Inverse problem for dynamic structural health monitoring based on slime mould algorithm," *Engineering with Computers*, pp. 1–24, 2021.

[7] F. Zhao, X. He, and L. Wang, "A two-stage cooperative evolutionary algorithm with problem-specific knowledge for energy-efficient scheduling of no-wait flow-shop problem," *IEEE Transactions on Cybernetics*, vol. 51, no. 11, pp. 5291–5303, 2021.

[8] F. Zhao, L. Zhao, L. Wang, and H. Song, "An ensemble discrete differential evolution for the distributed blocking flowshop scheduling with minimizing makespan criterion," *Expert Systems with Applications*, vol. 160, Article ID 113678, 2020.

[9] F. Zhao, R. Ma, and L. Wang, "A self-learning discrete jaya algorithm for multiobjective energy-efficient distributed No-idle flow-shop scheduling problem in heterogeneous factory system," *IEEE Transactions on Cybernetics*, pp. 1–12, 2021.

[10] A. W. Mohamed, A. A. Hadi, and A. K. Mohamed, "Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 7, pp. 1501–1529, 2020.

[11] R. Moghdani and K. Salimifard, "Volleyball premier league algorithm," *Applied Soft Computing*, vol. 64, pp. 161–185, 2018.

[12] M. A. Awadallah, M. A. Al-Betar, A. L. A. Bolaji, I. A. Doush, A. I. Hammouri, and M. Mafarja, "Island artificial bee colony for global optimization," *Soft Computing*, vol. 24, pp. 1–27, 2020.

[13] Y. Feng, S. Deb, G.-G. Wang, and A. H. Alavi, "Monarch butterfly optimization: a comprehensive review," *Expert Systems with Applications*, vol. 168, Article ID 114418, 2021.

[14] G.-G. Wang, "Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems," *Memetic Computing*, vol. 10, no. 2, pp. 151–164, 2018.

[15] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.

[16] A. Kaveh and A. Dadras, "A novel meta-heuristic optimization algorithm: thermal exchange optimization," *Advances in Engineering Software*, vol. 110, pp. 69–84, 2017.

[17] A. Tabari and A. Ahmad, "A new optimization method: Electro-Search algorithm," *Computers and Chemical Engineering*, vol. 103, pp. 1–11, 2017.

[18] A. F. Nematollahi, A. Rahiminejad, and B. Vahidi, "A novel physical based meta-heuristic optimization method known as Lightning attachment procedure optimization," *Applied Soft Computing*, vol. 59, pp. 596–621, 2017.

[19] P. C. Song, S. C. Chu, J. S. Pan, and H. Yang, "Phasmatodea population evolution algorithm and its application in length-changeable incremental extreme learning machine," in *Proceedings of the 2020 2nd International Conference on Industrial Artificial Intelligence (IAI)*, pp. 1–5, IEEE, Shenyang, China, October 2020.

[20] P. C. Song, S. C. Chu, J. S. Pan, and H. Yang, "Simplified Phasmatodea population evolution algorithm for optimization," *Complex and Intelligent Systems*, pp. 1–19, 2021.

[21] X. Yuan, T. Zhang, X. Dai, and L. Wu, "Master-slave model-based parallel chaos optimization algorithm for parameter identification problems," *Nonlinear Dynamics*, vol. 83, no. 3, pp. 1727–1741, 2016.

[22] A. Lančinskas, P. M. Ortigosa, and J. Žilinskas, "Parallel optimization algorithm for competitive facility location," *Mathematical Modelling and Analysis*, vol. 20, no. 5, pp. 619–640, 2015.

[23] S. Lalwani, H. Sharma, S. C. Satapathy, K. Deep, and J. C. Bansal, "A survey on parallel particle swarm optimization algorithms," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 2899–2923, 2019.

[24] K. Shen, T. De Pessemier, L. Martens, and W. Joseph, "A parallel genetic algorithm for multi-objective flexible flowshop scheduling in pasta manufacturing," *Computers and Industrial Engineering*, vol. 161, Article ID 107659, 2021.

[25] Z. Yang, B. Yu, and C. Cheng, "A Parallel ant colony algorithm for bus network optimization," *Computer-Aided Civil and Infrastructure Engineering*, vol. 22, no. 1, pp. 44–55, 2007.

[26] F. Yan, Y. Hu, J. Jia, Z. Ai, and X. Liu, "Interactive WEBVR visualization for online fire evacuation training," *Multimedia Tools and Applications*, vol. 79, no. 4, pp. 31541–31565, 2020.

[27] F.-t. Yan, Y.-h. Hu, J.-y. Jia, Q.-h. Guo, H.-h. Zhu, and Z.-g. Pan, "RFES: a real-time fire evacuation system for mobile

web3d," *Frontiers of Information Technology and Electronic Engineering*, vol. 20, no. 8, pp. 1061–1074, 2019.

[28] J. Zhang, S. Lin, H. Liu, Y. Chen, M. Zhu, and Y. Xu, "A small-population based parallel differential evolution algorithm for short-term hydrothermal scheduling problem considering power flow constraints," *Energy*, vol. 123, pp. 538–554, 2017.

[29] S. Lin and J. Wang, "Carbon emission reduction effect of transportation structure adjustment in China: an approach on multi-objective optimization model," *Environmental Science and Pollution Research*, vol. 29, pp. 1–18, 2021.

[30] S. Chen, "Highway transportation optimization control system based on OD forecast information," *International Journal of System Assurance Engineering and Management*, vol. 12, pp. 1–9, 2021.

[31] C.-H. Chen, "An arrival time prediction method for bus system," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4231-4232, 2018.

[32] F. T. Yan, J. Y. Jia, Y. H. Hu, Q. H. Guo, and H. H. Zhu, "Smart fire evacuation service based on Internet of Things computing for Web3D," *Journal of Internet Technology*, vol. 20, no. 2, pp. 521–532, 2019.

[33] Y. Bie, M. Hao, and M. Guo, "Optimal electric bus scheduling based on the combination of all-stop and short-turning strategies," *Sustainability*, vol. 13, no. 4, p. 1827, 2021.

[34] Y. Alwesabi, Y. Wang, R. Avalos, and Z. Liu, "Electric bus scheduling under single depot dynamic wireless charging infrastructure planning," *Energy*, vol. 213, Article ID 118855, 2020.

[35] W.-L. Liu, Y.-J. Gong, W.-N. Chen, Z. Liu, H. Wang, and J. Zhang, "Coordinated charging scheduling of electric vehicles: a mixed-variable differential evolution approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 5094–5109, 2020.

[36] Z. Yang, S. Zhao, and Q. Zhao, "Research on bus scheduling based on artificial immune algorithm," in *Proceedings of the 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–4, IEEE, Dalian, China, October 2008.

[37] C. Wang, C. Guo, and X. Zuo, "Solving multi-depot electric vehicle scheduling problem by column generation and genetic algorithm," *Applied Soft Computing*, vol. 112, Article ID 107774, 2021.

[38] A. Xiao, "A self-adaptive quantum genetic algorithm for network flow vehicle scheduling problem," *Journal of Computer and Communications*, vol. 9, no. 7, pp. 43–54, 2021.

[39] Y. Zhou, Q. Luo, and J. Liu, "Glowworm swarm optimization for dispatching system of public transit vehicles," *Neural Processing Letters*, vol. 40, no. 1, pp. 25–33, 2014.

[40] Z. Fu, P. Hu, W. Li, J.-S. Pan, and S.-C. Chu, "Parallel equilibrium optimizer algorithm and its application in capacitated vehicle routing problem," *Intelligent Automation and Soft Computing*, vol. 27, no. 1, pp. 233–247, 2021.

[41] X.-S. Yang, "Flower pollination algorithm for global optimization," in *Proceedings of the International Conference on Unconventional Computing and Natural Computation*, pp. 240–249, Berlin, Heidelberg, September 2012.

[42] X.-S. Yang, M. Karamanoglu, and X. He, "Flower pollination algorithm: a novel approach for multiobjective optimization," *Engineering Optimization*, vol. 46, no. 9, pp. 1222–1237, 2014.

[43] T. Adithiyaa, D. Chandramohan, and T. Sathish, "Flower pollination algorithm for the optimization of stair casting parameter for the preparation of AMC," *Materials Today: Proceedings*, vol. 21, pp. 882–886, 2020.

[44] A. W. Mohamed, A. A. Hadi, and K. M. Jambi, "Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization," *Swarm and Evolutionary Computation*, vol. 50, Article ID 100455, 2019.

[45] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[46] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the ICNN'95-International Conference on Neural Networks*, pp. 1942–1948, IEEE, Perth, WA, Australia, November 1995.

[47] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.

[48] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[49] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[50] S. C. Chu, J. F. Roddick, and J. S. Pan, "A parallel particle swarm optimization algorithm with communication strategies," *Journal of Information Science and Engineering*, vol. 21, no. 4, p. 9, 2005.

[51] Q.-w. Chai, S.-C. Chu, J.-S. Pan, P. Hu, and W.-m. Zheng, "A parallel WOA with two communication strategies applied in DV-Hop localization method," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 4, 2020.

[52] Q. Yang, S.-C. Chu, J.-S. Pan, and C.-M. Chen, "Sine cosine algorithm with multigroup and multistrategy for solving CVRP," *Mathematical Problems in Engineering*, vol. 2020, no. 12, pp. 1–10, 2020.

[53] M. Zhu, S.-C. Chu, Q. Yang, W. Li, and J.-S. Pan, "Compact sine cosine algorithm with multigroup and multistrategy for dispatching system of public transit vehicles," *Journal of Advanced Transportation*, vol. 2021, no. 2, pp. 1–16, 2021.