Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 5526127, 16 pages
https://doi.org/10.1155/2021/5526127

Research Article

Compact Sine Cosine Algorithm with Multigroup and Multistrategy for Dispatching System of Public Transit Vehicles

Minghui Zhu,[1] Shu-Chuan Chu,[1] Qingyong Yang,[1] Wei Li,[2] and Jeng-Shyang Pan[1]

[1]*College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China*
[2]*College of Computer Science and Technology, Harbin Engineering University, Harbin150001, China*

Correspondence should be addressed to Jeng-Shyang Pan; jengshyangpan@gmail.com

Received 7 January 2021; Revised 4 March 2021; Accepted 20 March 2021; Published 30 March 2021

Academic Editor: Chi-Hua Chen

This paper studies the problem of intelligence optimization, a fundamental problem in analyzing the optimal solution in a wide spectrum of applications such as transportation and wireless sensor network (WSN). To achieve better optimization capability, we propose a multigroup Multistrategy Compact Sine Cosine Algorithm (MCSCA) by using the compact strategy and grouping strategy, which makes the initialized randomly generated value no longer an individual in the population and avoids falling into the local optimum. New evolution formulas are proposed for the intergroup communication strategy. Performance studies on the CEC2013 benchmark demonstrate the effectiveness of our new approach regarding convergence speed and accuracy. Finally, we apply MCSCA to solve the dispatch system of public transit vehicles. Experimental results show that MCSCA can achieve better optimization.

1. Introduction

The optimization problem is the problem of finding the optimal solution according to the optimization direction in a feasible solution domain defined by constraints [1]. Inspired by existing natural phenomena, a large number of researchers have devoted themselves to heuristic algorithms. Among them, the metaheuristic algorithm which shares the collective information of all individuals based on species behavior is a recent hotspot. Swarm intelligence algorithm is a metaheuristic algorithm [2] that can get better results without consuming too much computing time. Swarm intelligence generally simulates a living population, such as a particle swarm that simulates bird swarm [3, 4], wolf swarm [5], fish swarm [6], cat swarm [7–9], and artificial bee colony [10, 11]. There are also inanimate objects like fireworks [12]. Swarm intelligence optimization algorithms generally have the following characteristics: there are many particles (representing each type of agent), the individuals are independent of each other, individuals move position according to different mechanisms to explore the solution space, and the position movement mechanism generally introduces random numbers for better exploration.

The Sine Cosine Algorithm (SCA) [1] is a new swarm intelligence optimization algorithm proposed by the Australian scholar Mirjalili in 2016. This algorithm has a simple structure and fewer parameters and is easy to implement. In recent years, many scholars have further optimized this algorithm to solve different problems. It is mainly optimized from two aspects. The one aspect is to optimize the algorithm itself, such as the multiobjective version of the SCA algorithm [13] and the binary version of the SCA algorithm [14, 15], based on the opposite learning improved SCA algorithm [16, 17]. The other is to combine the algorithm with other algorithms, such as the SCA algorithm combined with particle swarm optimization algorithm [18, 19] and the SCA algorithm combined with differential evolution algorithm [20–23].

Sine Cosine Algorithm with Multigroup and Multistrategy (MMSCA) is an improved SCA algorithm [24]. MMSCA divides the population into groups and uses different update strategies in different groups. It uses two strategies: rand strategy, and best strategy. In the rand strategy, the next generation of solving goals is chosen by using multiple roulette methods. In the best strategy, the best individual in all populations is still the solution target.

The SCA and the improved SCA algorithms [24–26] are used in many different fields to solve different problems. For example, the SCA algorithm can solve the economic and emission dispatch problem [27, 28], a design damping controller [29], and a system for battery charging [30], predict wind speed [31], reconfigure the power distribution network [32], segment image [33], and control the load frequency of power system [34]. The SCA and improved SCA algorithms still have some disadvantages. A disadvantage of the SCA algorithm is the algorithm that requires evaluation of each solution in the swarm and needs to call the target function multiple times. It increases the complexity of the algorithm. Another disadvantage is the slow convergence speed.

Based on the above discussions, it is key to effectively deal with high complexity and slow convergence speed. Unlike the existing SCA methods, we develop a compact SCA with multigroup and multistrategy, named MCSCA, to make the initialized randomly generated value no longer an individual in the population and avoid falling into the local optimum. Compact strategy [35–38] is a technique based on a probabilistic model. This technique reduces the memory usage and computation time of the algorithm by using a probabilistic model to replace the population from a macroperspective. The idea of grouping [39–41] implies that many populations can be iteratively updated simultaneously. The advantage of this method is that it ensures population diversity and further improves the search capability and performance of the algorithm. In particular, when solving complex optimization problems, grouping populations is an effective way to improve the efficiency and accuracy of the algorithm.

The superiority of the proposed MCSCA is proved by the comparison between MCSCA and several heuristic algorithms on benchmark functions. To test the ability of MCSCA in solving practical problems, we apply it to the dispatching system of public transit vehicles, which is one of the problems that transportation needs to solve, which is related to the economic and social benefits of public transportation companies.

The main contributions of the paper are listed as follows:

(1) We develop a new compact SCA algorithm for intelligent optimization problems
(2) We propose an optimization approach by designing new evolution formulas and utilizing a grouping strategy
(3) We extend the MCSCA algorithm in the application of the dispatching system of public transit vehicles

We conduct extensive empirical studies on the CEC2013 benchmark in Section 4. The empirical studies confirm that our new approach significantly outperforms the state-of-the-art approaches.

The rest of the paper is organized as follows. Section 2 gives a brief retrospect to the SCA algorithm and the basic principles of the compact. Section 3 formally proposes the MCSCA algorithm. In Section 4, the results of numerical experiments are presented and discussed. Section 5 introduces the application of this algorithm in the dispatching system of public transit vehicles. Finally, Section 6 gives a conclusion.

2. Related Works

Besides the related works discussed above, other related works are categorized as follows.

2.1. Sine Cosine Algorithm. The SCA algorithm is a metaheuristic algorithm that uses mathematical equations to estimate the global optimal solution of the optimization problem. It creates multiple initial random candidate solutions and requires them to use mathematical models based on sine and cosine functions to fluctuate outward or toward the best solution. The algorithm also integrates several random and adaptive variables to emphasize the exploration and use of the search space at different stages of optimization. The update formula is as follows:

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times \sin(r_2) \times \left| r_3 \times P_b^t - X_i^t \right|, & r_4 < 0.5, \\ X_i^t + r_1 \times \cos(r_2) \times \left| r_3 \times P_b^t - X_i^t \right|, & r_4 \geq 0.5, \end{cases} \tag{1}$$

where X_i^t is the position of the i-th dimension in the t-th iteration. r_1, r_2, r_3, and r_4 are random numbers, $r_2 \in [0, 2\pi]$, $r_3 \in [0, 2]$, and $r_4 \in [0, 1]$, and p_b^t represents the best individual position after t iterations. The updated formula of r_1 is as follows:

$$r_1 = a - a\frac{t}{T}, \tag{2}$$

where a is a constant, t is the number of current iterations, and T is the maximum number of iterations.

2.2. Fundamentals of Compact and Grouping. The essence of the compact is based on a probability model to represent the distribution of all particles. The virtual population replaces the actual solution population. This virtual population is encoded in a data structure. This data structure is named the disturbance vector and represented by PV. There are K particles in the original population, and each particle has n dimensions. After adding the compact, each dimension can be represented by a normal distribution, so that the original $K \times n$ matrix becomes a $2 \times n$ matrix:

$$PV^t = \left[\mu^t, \sigma^t\right], \tag{3}$$

where μ and σ are, respectively, vectors containing, for each design variable, mean and standard deviation values of a Gaussian Probability Distribution Function (PDF) [42] truncated within the interval [-1, 1].

The sampling mechanism of a design variable $x[i]$ associated with a generic candidate solution x from PV requires an extensive explanation. For each design variable indexed by i, a truncated Gaussian PDF with mean value $\mu[i]$ and the standard deviation $\sigma[i]$ is associated.

$$\mathrm{PDF}(\mathrm{truncNorm}(x)) = \frac{e^{-\left(\left((x-\mu[i])^2\right)/\left(2\sigma[i]^2\right)\right)\sqrt{2/\pi}}}{\sigma[i]\left(\mathrm{erf}\left((\mu[i]+1)/\left(\sqrt{2}\,\sigma[i]\right)\right) - \mathrm{erf}\left((\mu[i]-1)/\sqrt{2\sigma[i]}\right)\right)}, \tag{4}$$

where erf is the error function; see [43]. The update rule for μ and σ is

$$\mu^{t+1}[i] = \mu^t[i] + \frac{1}{N_p}(\mathrm{winner}[i] - \mathrm{loser}[i]), \tag{5}$$

$$\begin{aligned}\left(\sigma^{t+1}[i]\right)^2 &= \left(\sigma^t[i]\right)^2 + \left(\mu^t[i]\right)^2 - \left(\mu^{t+1}[i]\right)^2 \\ &\quad + \frac{1}{N_p}\left(\mathrm{winner}[i]^2 - \mathrm{loser}[i]^2\right),\end{aligned} \tag{6}$$

where i represents the i-th dimension, N_p is the virtual population number, and winner and loser are obtained by comparing the two particles generated by PV.

By constructing a Chebyshev polynomial, the probability density function can correspond to a cumulative distribution function with values ranging from 0 to 1. The cumulative distribution function is described by the following equation:

$$\begin{aligned}\mathrm{CDF} &= \int_{-1}^{x} \mathrm{PDF}dx = \int_{-1}^{x} \frac{\sqrt{2/\pi}e^{-\left((x-\mu)^2/2\sigma^2\right)}}{\sigma\mathrm{erf}\left((\mu+1)/\sqrt{2}\,\sigma\right) - \mathrm{erf}\left((\mu-1)/\sqrt{2}\,\sigma\right)}dx, \\ \mathrm{CDF} &= \frac{\mathrm{erf}\left((\mu+1)/\sqrt{2}\,\sigma\right) + \mathrm{erf}\left((x-\mu)/\sqrt{2}\,\sigma\right)}{\mathrm{erf}\left((\mu+1)/\sqrt{2}\,\sigma\right) - \mathrm{erf}\left((\mu-1)/\sqrt{2}\,\sigma\right)}.\end{aligned} \tag{7}$$

The inverse function of the cumulative distribution is as follows:

$$y = \sqrt{2}\delta\,\mathrm{erf}^{-1}\left(-\mathrm{erf}\left(\frac{\mu+1}{\sqrt{2}\,\delta}\right) - x\mathrm{erf}\left(\frac{\mu-1}{\sqrt{2}\,\delta}\right) + x\mathrm{erf}\left(\frac{\mu+1}{\sqrt{2}\,\delta}\right)\right) + \mu. \tag{8}$$

Many metaheuristic algorithms have applied the idea of grouping, such as parallel PSO [39], parallel ACO [40], and parallel QUATRE [41]. This method changes the basic characteristics of the traditional SCA algorithm. Each group evolves independently, and the entire population adopts a unified mechanism to evolve. Therefore, it can effectively increase the speed of calculation. Different groups adopt different update formulas, which helps prevent falling into the local optimum while preventing premature convergence.

Although the compact strategy reduces memory usage and computation time, it limits the search capability and convergence speed of the algorithm. Therefore, the compact strategy and the group strategy are combined to improve the performance of the algorithm [44].

2.3. Dispatching System of Public Transit Vehicles. There are many issues worth studying in the direction of transportation, such as vehicle speed estimation and prediction [45, 46], traffic monitoring [47], traffic accident prediction [48], bus arrival time prediction [49], and dispatching system of public transit vehicles [50]. Public transit vehicles have the advantages of large passenger capacity, low road traffic conditions, and access to many areas. So, public transit vehicle is an important direction in this field. It has many issues worth studying, such as the choice of bus types, site construction and road selection [51], calculation of greenhouse gas emissions from public transit vehicles [52], integration of unmanned driving, and public transportation.

In this paper, the improved SCA algorithm is applied to the dispatching system of public transit vehicles. Intelligent dispatch of operating vehicles is one of the problems that transportation needs to solve, which is related to the economic and social benefits of public transportation companies. The bus scheduling problem is a complex nonlinear dynamic optimization problem using a bus dispatching model that takes into account the interests of both the company and the passengers. It is to find an optimal solution that satisfies the proposed objective function in the solution space that satisfies the scheduling constraints. Many algorithms have been used for the dispatching system of public transit vehicles, such as ACO [53], GA [54, 55], and Immune Algorithms (IA) [56]. The dispatching system of public transit vehicles still has many problems such as the fact that weights of the multiobjective function are difficult to be determined, the original data source is not accurate, and the algorithm convergence is slow.

3. Compact SCA Algorithm with Multigroup and Multistrategy

In this section, we first propose a compact SCA algorithm in Section 3.1 and then develop optimization techniques inspired by the idea of grouping to avoid the local optimum issue in Section 3.2.

3.1. Compact Sine Cosine Algorithm (CSCA). The pseudocode of our compact SCA algorithm denoted by CSCA is shown in Algorithm 1. We initially set the dimension as d, the population size as Np, and the upper and lower bounds as ub and lb. The vectors μ and σ of PV are initialized to 0 and 10. The variable best is set as ub, which is used to store the global optimum, and Fmin is set to infinity, which is used to store the fitness value corresponding to the global optimum (Line 1–4). Then, we iteratively use equation (8) to generate a random solution ranging from -1 to 1 and use equation (9) to control the generated solution in the decision space (Line 6). The solution is updated according to equation (1) (Line 7). The variables winner and loser are set as a random solution and updated solution. The values of the objective function corresponding to the two solutions are compared, and winner is set as the better solution (Line 8). After that, for the solution, we first update PV according to equations(5)and(6) (Line 10). The variable $\mathrm{fit}_{\mathrm{winner}}$ represents the objective function value corresponding to winner. If $\mathrm{fit}_{\mathrm{winner}}$ is less than the given threshold, the best and Fmin values will be updated to winner and $\mathrm{fit}_{\mathrm{winner}}$, respectively (Line 11–13). Finally, $\mathrm{fit}_{\mathrm{winner}}$ and Fmin are compared to update the global optimum.

$$x_1 = \frac{x}{2}(\mathrm{ub} - \mathrm{lb}) + \frac{1}{2}(\mathrm{ub} + \mathrm{lb}). \tag{9}$$

3.2. Compact Sine Cosine Algorithm with Multigroup and Multistrategy (MCSCA). The SCA algorithm can generate a set of values per iteration. After adding the idea of compact, the initialized randomly generated value is no longer an individual in the population, but a set of vectors used for compact. In this case, it is easy to fall into the local optimum. Therefore, the idea of grouping is added. Grouping is a common idea in optimization algorithms. After adding group ideas, a set of values can be generated per iteration, which increases the diversity of the population and jumps out of the local optimal. So, the searchability and convergence speed are improved, and the solution is found faster (Algorithm 1).

The communication strategy between groups adopts the idea of differential evolution. The differential evolution algorithm is a random parallel optimization algorithm based on population evolution and is a simple and efficient random global optimization method [57]. This article uses the following differential evolution formula and improved differential evolution formula:

$$\text{DE/best/2: } x = x_{\mathrm{best}} + F(x_{r1} - x_{r2}) + F(x_{r3} - x_{r4}), \tag{10}$$

$$\begin{aligned}\text{DE/current} - \text{to} - \text{rand/1: } x = x_{\mathrm{current}} &+ F(x_{\mathrm{current}} - x_{r1}) \\ &+ F(x_{r2} - x_{r3}),\end{aligned} \tag{11}$$

$$\begin{aligned}\text{DE/best} - \text{to} - \text{rand/1: } x = x_{\mathrm{best}} &+ F(x_{\mathrm{best}} - x_{r1}) \\ &+ F(x_{r2} - x_{r3}),\end{aligned} \tag{12}$$

$$\begin{aligned}\text{DE/best} - \text{to} - \text{current/1: } x = x_{\mathrm{best}} &+ F(x_{\mathrm{best}} - x_{\mathrm{current}}) \\ &+ F(x_{r1} - x_{r2}),\end{aligned} \tag{13}$$

where F stands for mutation factor, generally taking the value 0.5, and x_{r1}, x_{r2}, x_{r3}, and x_{r4} represent random individuals in the population. x_{best} is the global optimal individual and x_{current} represents the current individual.

The original differential evolution strategy is to update the individuals in the population according to the above formula. This algorithm groups the population, and x in the corresponding formula also changes from the individual to the optimal of a group. For example, x_r originally represents a random individual in the population, which represents the optimal individual in a random group in this algorithm. Four formulas are used for communication between groups, among which equations (11)–(13) are improved on common formulas. DE/current-to-rand/1 represents updating the current solution with a random solution, DE/best-to-rand/1 represents updating the global optimum with a random solution, and DE/best-to-current/1 represents updating the global optimum with the current solution. Using the random solution and the current solution to influence the global optimum can speed up the convergence speed, and updating the current solution can increase the randomness of the solution.

The pseudocode of our Compact Sine Cosine Algorithm with Multigroup and Multistrategy denoted by MCSCA is shown in Algorithm 2. We initially set the variable globalbest to represent the global optimum, set the variable globalfmin to represent the fitness value corresponding to the global optimum, set the variable g to represent the number of groups, set $G[i] \cdot \mathrm{best}$ to store the best in the group, and set $G[i] \cdot F\min$ to store the fitness value corresponding to $G[i] \cdot \mathrm{best}$(Line 1–3). Dimension, population size, upper, lower, and vectors of PV are consistent with those set by CSCA. Then, we iteratively use equations (8) and (9) to generate a random solution in the decision space and update this solution according to equation (1) (Line 6–7). winner and loser are set as this random solution and updated solution. The values of the objective function corresponding to the two solutions are compared, and winner is set as the better solution. $\mathrm{fit}_{\mathrm{winner}}$ represents the objective function value corresponding to the winner, and we update PV according to equations (5) and (6) (Line 9).

After that, the optimum in the group and the corresponding fitness value are updated by comparing $\mathrm{fit}_{\mathrm{winner}}$ with $G[i] \cdot F\min$. The global optimum is updated by communication strategy between groups. The specific intergroup communication strategy is to cycle the number of groups, and the following four strategies are executed with equal probability.

Strategy 1 compared the fitness value of the current group and the global optimum to determine whether to replace it. If there is no need to replace, update the global optimum according to equation (10). Strategy 2 compares the optimal value of the current group and the optimal value of the random group. If the fitness value of the random group is better, the best value and fitness value of the current group are replaced with the best value and fitness value of this random group. Otherwise, the random group is updated according to equation (11). Strategy 3 and Strategy 4 update the global optimum according to equations (12) and (13), respectively. After executing the intergroup strategy, use the sorting statement to find the worst and the best of optimal individuals of all groups and update the worst to the best (Algorithm 2).

4. Experiments

We conduct extensive empirical studies to evaluate the efficiency and effectiveness of our proposed algorithms for the dispatching system of public transit vehicles. In specific, we evaluate the following algorithms:

(1) SCA: the algorithm in [1].

(2) MMSCA: the algorithm in [24].

(3) CSCA: the approach discussed in Section 3.1

(4) MCSCA: the approach discussed in Section 3.2

All algorithms are implemented in MATLAB2018a. All experiments are conducted on a machine with an Intel (R) Core (TM) 2.60 GHz CPU and 12 GB memory running 64-bit Windows 10. We evaluate the performance of all algorithms on the CEC2013 benchmark functions as follows:

(1) $\mathbf{f}_1 - \mathbf{f}_5$ are unimodal benchmark functions

(2) $\mathbf{f}_6 - \mathbf{f}_{20}$ are basic multimodal functions

(3) $\mathbf{f}_{21} - \mathbf{f}_{28}$ are composite functions

There are four key parameters in the intergroup communication strategy of the MCSCA algorithm, that is, the mutation factor corresponding to the four formulas. The value of the mutation factor will affect the convergence of the algorithm. If the value of the mutation factor is too small, it will easily fall into the local optimum, and if the value of the mutation factor is too large, the algorithm is difficult to converge. So, Taguchi's method [58] is used to obtain a reasonable combination of the four mutation factors. Taguchi's method detects part of the possible combination of factors, but not all combinations. This method uses the smallest number of trials to detect the best combination. The value of F ranges from 0 to 2. So, the level settings of the four factors are 0.5, 1, and 1.5. A full-factorial analysis needs $3^4 = 81$ experiments. Taguchi's method adopts the orthogonal arrays. Therefore, an orthogonal array L_9, (3^4) that contains only 9 experiments is adopted in our experiment. We conducted ten tests on CEC2013 to find the average value. When the mutation factors are all 0.5, the effect is the best.

4.1. Comparing CSCA and MCSCA with SCA and MMSCA.

We compare our proposed algorithms, CSCA and MCSCA, with the existing algorithms, SCA and MMSCA. Here, we selected 15 figures with obvious optimization effects to show in Figure 1. Table 1 presents the results in the 28 CEC2013 benchmark functions. Generally, MCSCA outperforms CSCA; the only difference between these two algorithms is that MCSCA uses the proposed Multigroup and Multistrategy. Thanks to the compact technique, the performance of CSCA is much better than the performance of SCA in most functions.

As shown in Table 1, compared with the SCA algorithm, the MCSCA achieves better results in 16 benchmark functions. Compared with MMSCA, the proposed MCSCA algorithm performs better on 17 benchmark test functions. Compared with CSCA, the MCSCA algorithm is better in all benchmark functions. Among the unimodal benchmark functions, our MCSCA approach over $\mathbf{f}_1$, $\mathbf{f}_2$, and $\mathbf{f}_5$ can achieve the best accuracy compared to that of SCA and MMSCA algorithms. From Figure 1, we can see that the convergence speed in our MCSCA algorithm increases significantly. Among the eight composite functions, for $\mathbf{f}_{26}$ and $\mathbf{f}_{28}$, the MMSCA algorithm is better optimized. For the remaining six composite functions, the MCSCA algorithm is better optimized. The reason for the significant optimization effect is the intergroup communication strategy of MCSCA, which uses the idea of differential evolution. The basic idea of the communication strategy between groups is to use the difference between the two group optimal values to affect the current group optimal value or the global optimal value. It increases the randomness of the solution. After intergroup communication is complete, the worst individual in all groups is found by a ranking statement and replaced with the global optimum. This method speeds up convergence.

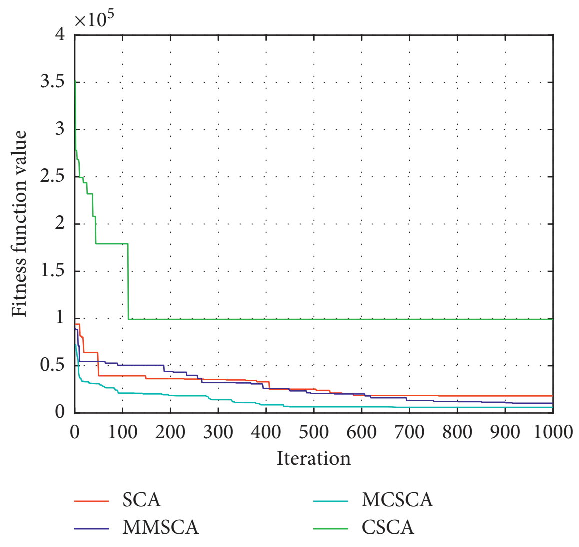
4.2. Comparison with the Existing Compact Methods.

To further verify the effectiveness of the algorithm, this paper also compares it with three other compact algorithms, including CBA, PCABC, and CPSO. The number of iterations is one thousand. Each algorithm is tested 30 times for the average. The experimental results are shown in Table 2.

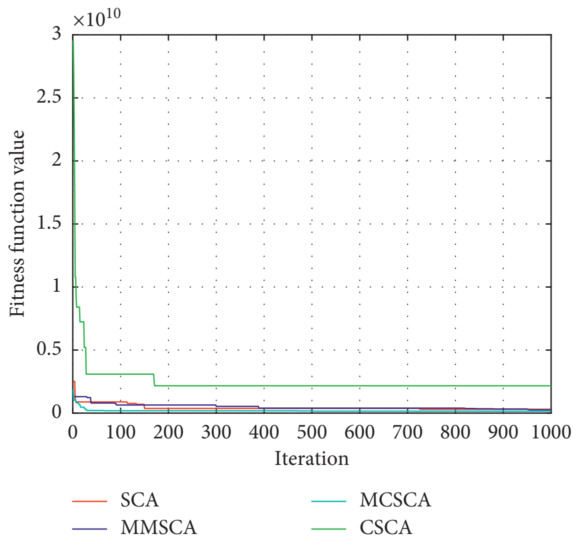
It can be seen that the MCSCA algorithm has better solution accuracy compared to CBA, CPSO, and PCABC, except for $\mathbf{f}_1$. CBA and CPSO approach only add compact ideas to the original SCA algorithm, which saves memory but is easy to fall into a local optimum and converge ahead of time. The PCABC, like the MCSCA algorithm, adds grouping ideas to the compact algorithm.

But the intergroup communication strategy of the MCSCA algorithm uses the idea of differential evolution and multiple strategies. It updates the group optimum to increase the diversity of population solutions and updates the global optimum to maintain the strong local optimization ability of the original algorithm and avoid the instability of the algorithm caused by the introduction of the rand strategy. Thus, the MCSCA algorithm has a more pronounced optimization effect.

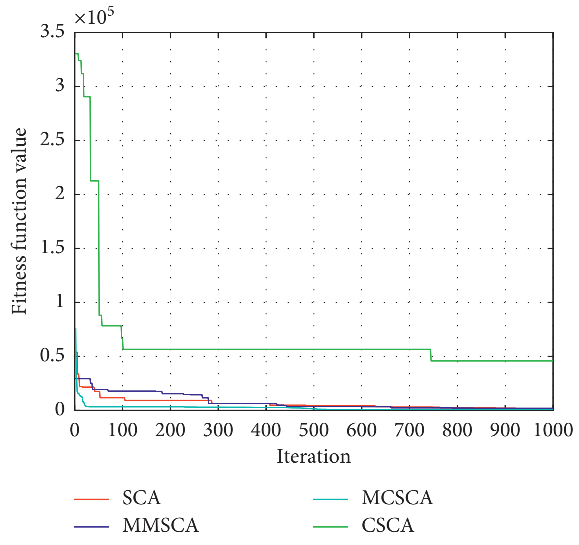
According to the comparison results of the two phases of experiments, the MCSCA algorithm shows strong competitiveness compared with other algorithms of the same type. To further study the feasibility and effectiveness of this



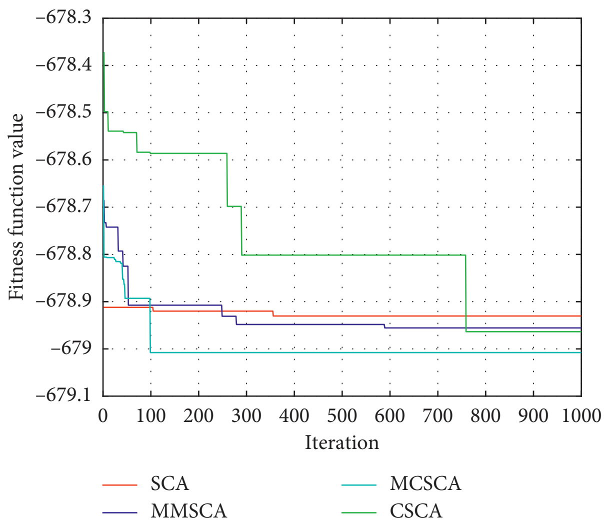
(a)



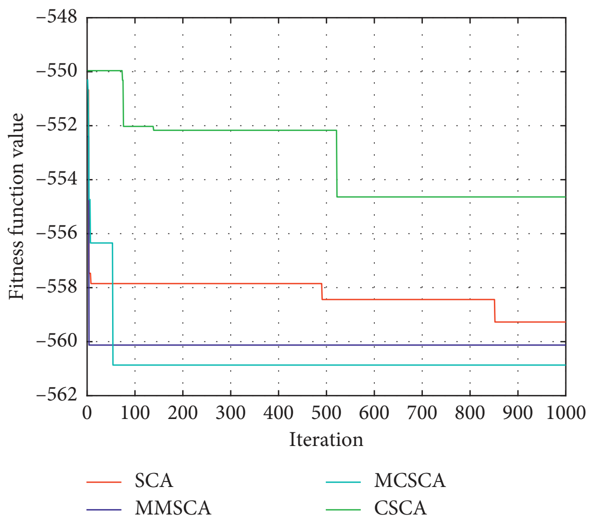
(b)



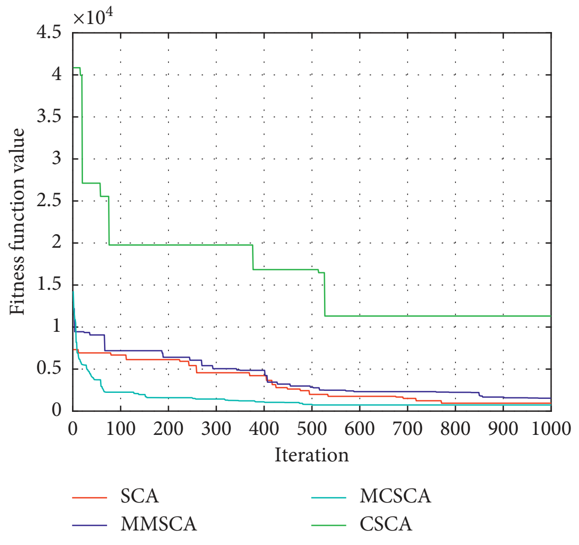
(c)



(d)

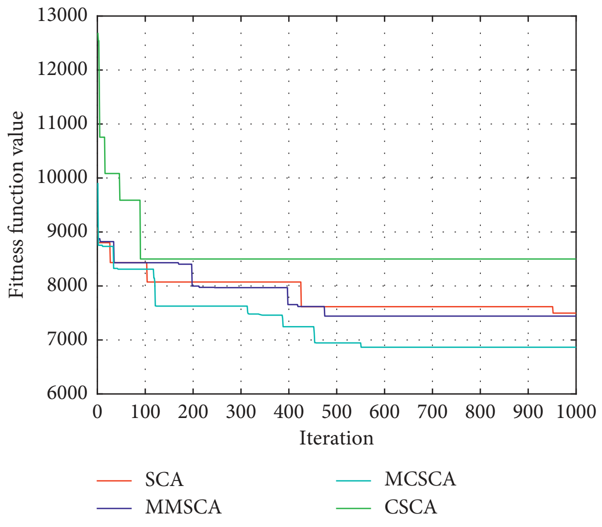


(e)

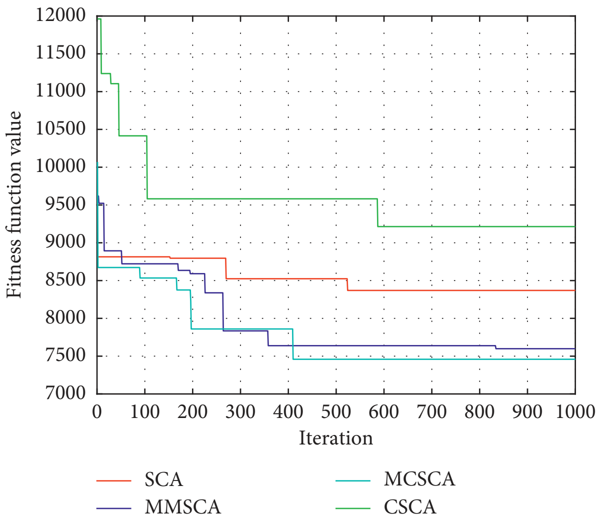


(f)

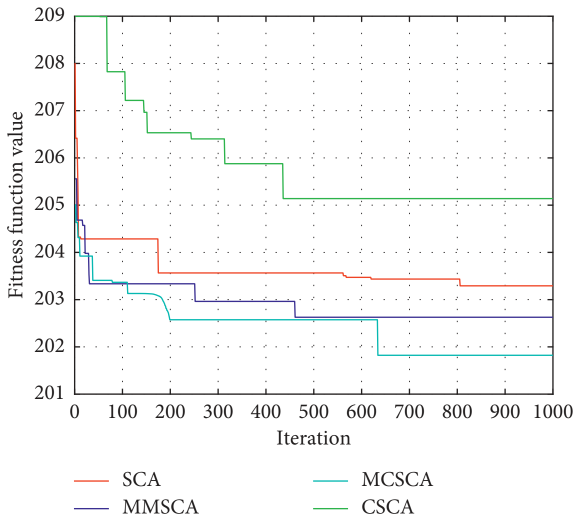
Figure 1: Continued.



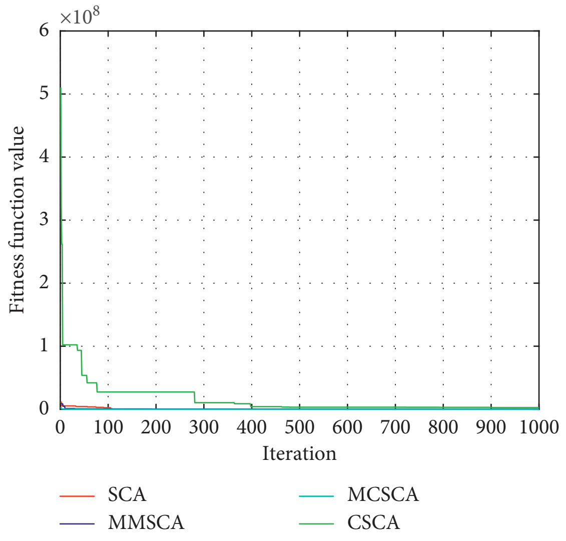
(g)



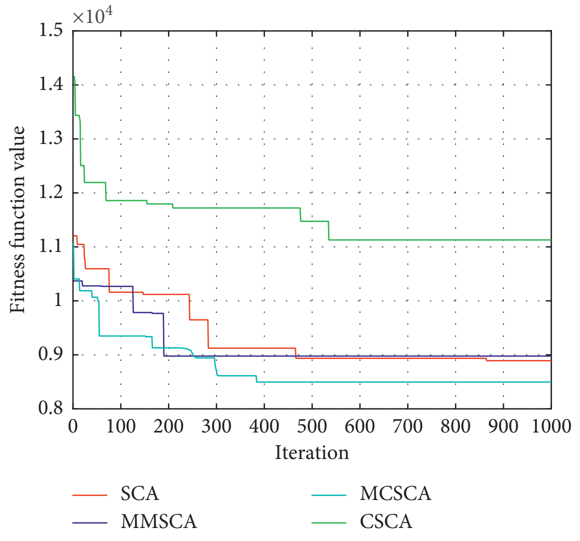
(h)



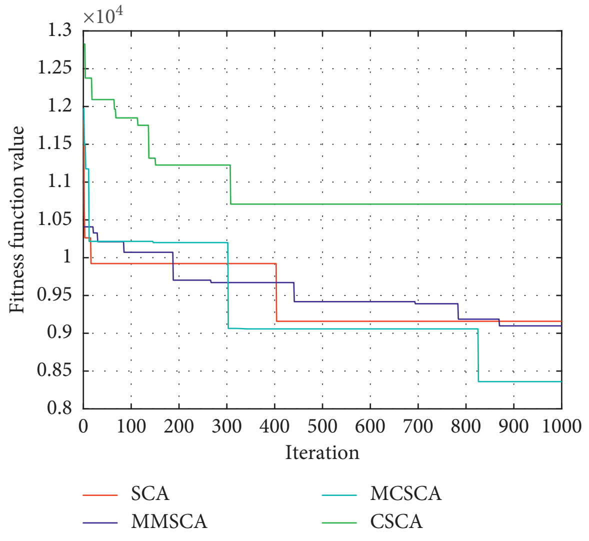
(i)



(j)

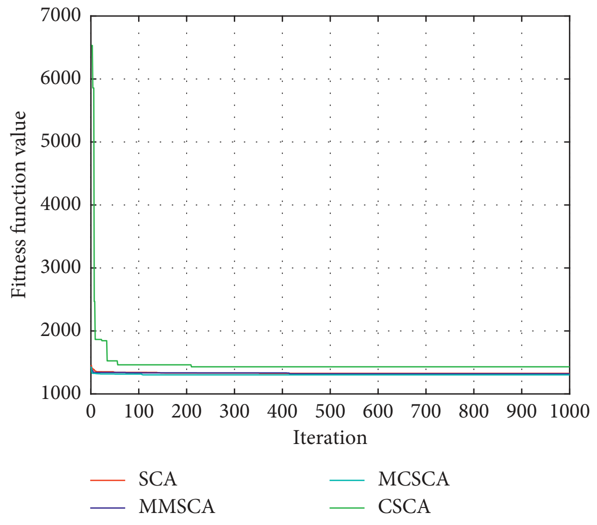


(k)

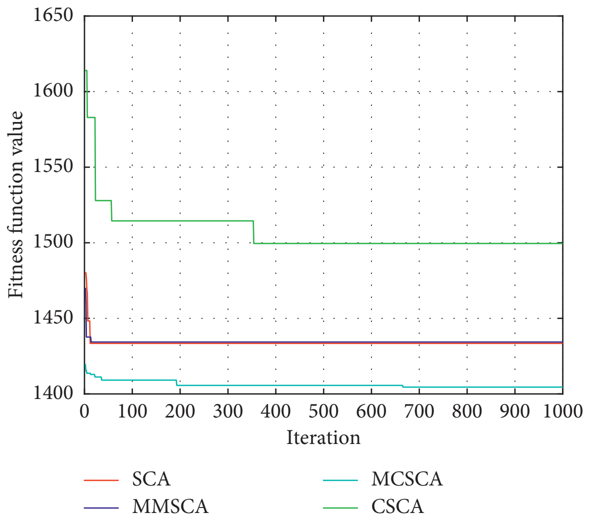


(l)

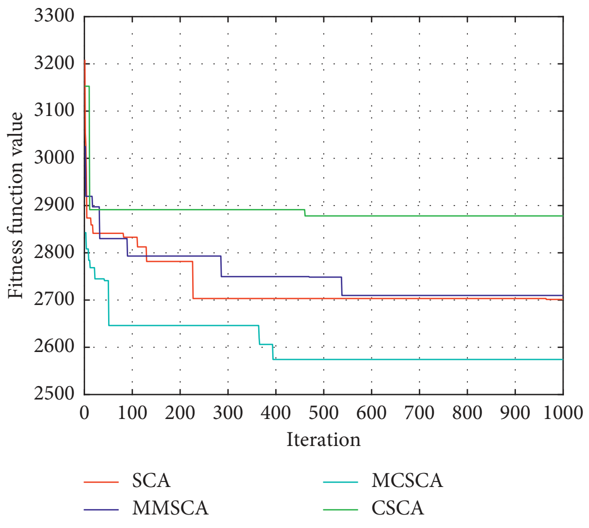
Figure 1: Continued.



(m)



(n)



(o)

FIGURE 1: Convergence plots for optimization in the CEC2013 benchmark functions. (a) $\mathbf{f}_1$, (b) $\mathbf{f}_2$, (c) $\mathbf{f}_5$, (d) $\mathbf{f}_8$, (e) $\mathbf{f}_9$, (f) $\mathbf{f}_{10}$, (g) $\mathbf{f}_{14}$, (h) $\mathbf{f}_{15}$, (i) $\mathbf{f}_{16}$, (j) $\mathbf{f}_{19}$, (k) $\mathbf{f}_{22}$, (l) $\mathbf{f}_{23}$, (m) $\mathbf{f}_{24}$, (n) $\mathbf{f}_{25}$, and (o) $\mathbf{f}_{27}$.

```
   Input: Parameters $d$, $N_p$, ub, and lb
Output: Global optimum best and its fitness value $F\min$
 (1)   for $i = 1{:}d$ do
 (2)      initialize $\mu = 0$, $\sigma = \lambda = 10$;
 (3)      initialize best = ub, $F\min = \inf$ ;
 (4)   end for
 (5)   while t<Max Generation do
 (6)      Get $x_1$ from $PV$ via equations (8), (9);
 (7)      Update $x_1$ to get $x_2$ via equation (1);
 (8)      [winner, loser] = compete ($x_1$ ,$x_2$ );
 (9)      for $i = 1$:d do
(10)         Update $PV$ via equations (5), (6);
(11)         if $\mathrm{fit}_{\mathrm{winner}} <$ Fmin then
(12)         best = winner; $F\min = \mathrm{fit}_{\mathrm{winner}}$ ;
(13)         end if
(14)      end for
(15)   end while
```

ALGORITHM 1: CSCA.

```
     Input: Parameters $d$, $N_p$, ub , lb, and $g$
     Output: Global optimum globalbest and its fitness value globalfmin
(1)    Set the number of groups $g$, each group is $G_i$;
(2)    initialize $G[i] \cdot PV$ , $G[i] \cdot$ best and $G[i] \cdot F$ min of each group;
(3)    Initialize globalbest = $G[1] \cdot$ best, globalfmin = $G[1] \cdot F$ min;
(4)    while $t$<Max Gerneration do
(5)      for $i = 1{:}g$ do
(6)        Get $x_1$ , $x_2$ via equations (8). (9) and (1);
(7)        [winner, loser, $\text{fit}_{\text{winner}}$] = compete ($x_1$ ,$x_2$ );
(8)        for $i = 1$:d do
(9)          Update $G[i] \cdot PV$ via equations (5), (6);
(10)       end for
(11)       if $\text{fit}_{\text{winner}} < G[i] \cdot F$ min then
(12)         $G[i] \cdot$ best = winner; $G[i] \cdot F$ min = $\text{fit}_{winner}$;
(13)       end if
(14)       Update globalbest and globalfmin;
(15)       if rand<0.25 then
(16)         Execute Strategy 1;
(17)       else if rand<0.5 then
(18)         Execute Strategy 2;
(19)       else if rand<0.75 then
(20)         Execute Strategy 3;
(21)       else
(22)         Execute Strategy 4;
(23)       end if
(24)     end for
(25)   Change the worst individual to the best individual by the sorting statement;
(26) end while
```

Algorithm 2: MCSCA.

algorithm in practical applications, this paper applies the algorithm to the dispatching system of public transit vehicles.

4.3. Comparing MCSCA with DE and DE Variants. The intergroup communication strategy of the proposed algorithm uses a differential evolution mechanism. So, in order to fully demonstrate the effectiveness of the proposed method, this paper also compares the proposed algorithm with DE and DE variants, including JADE, SHADE, and LSHADE [59–61]. The test results are shown in Table 3. JADE algorithm incorporates parameter adaptation and greedy compilation strategies in the traditional DE algorithm. SHADE uses a historical memory of recently successful parameter sets based on JADE. LSHADE linearly reduces the population size based on SHADE.

According to the data in Table 3, compared with the DE algorithm and improved DE algorithms, the MCSCA algorithm has better results than these algorithms in 13 functions. But the effect on other benchmark functions is not the best. Compared with the original DE algorithm, the MCSCA algorithm is only unsatisfactory in the effects of f_{17}. The main reason for this effect is that the MCSCA algorithm divides the population into groups and adopts a variety of intragroup communication strategies. Different communication strategies focus on different goals. This method can accelerate the convergence speed without falling into the local optimum while reducing memory.

5. MCSCA Algorithm on the Dispatching System of Public Transit Vehicles

Public transportation is an important part of urban transportation. Good bus dispatching is of great significance for improving the urban transportation environment, improving the travel conditions of citizens, and improving the economic and social benefits of bus companies. Optimal bus dispatching is a rhythmic and repetitive driving plan. It organizes a large number of vehicles on a prescribed route and is made according to the number, direction, and time of passenger flow after studying the rules of passenger flow. It is a complex nonlinear dynamic optimization problem.

Before describing the optimal scheduling of buses, the following assumptions need to be made:

(1) The application is aimed at the dispatch of public transportation vehicles on a certain route in the urban public transportation system

(2) The model is built considering only the one-way case of this route

(3) The buses of this route are of the same vehicle type

(4) When each bus of this route passes through various stations, there are no passengers left

Table 1: Performance evaluation for CSCA, MCSCA, SCA, and MMSCA over CEC2013.

Function	MCSCA	MMSCA	SCA	CSCA
f$_1$	**1.0012e+04**	1.1018e+04	1.5563e+04	1.1834e+05
f$_2$	**1.5879e+08**	1.7454e+08	2.4311e+08	2.1502e+09
f$_3$	8.9088e+12	5.9098e+10	**5.3981e+10**	1.2261e+17
f$_4$	4.6156e+04	4.6370e+04	**4.5172e+04**	2.7359e+05
f$_5$	**1.8310e+03**	2.0867e+03	2.1676e+03	4.1026e+04
f$_6$	278.9563	**15.6414**	206.0107	2.5995e+04
f$_7$	320.5652	**−605.5122**	−533.9551	1.3614e+06
f$_8$	**−679.0443**	−679.0068	−678.9740	−678.8082
f$_9$	**−560.8198**	−560.1837	−558.7135	−554.2032
f$_{10}$	**1.2036e+03**	1.2840e+03	1.5836e+03	1.5415e+04
f$_{11}$	90.0472	**−10.3529**	12.7879	928.8658
f$_{12}$	227.4888	**85.8620**	106.4320	1.3068e+03
f$_{13}$	321.5240	**193.1019**	196.4991	1.4176e+03
f$_{14}$	**5.9463e+03**	7.2369e+03	7.2578e+03	9.1906e+03
f$_{15}$	**6.8223e+03**	7.5356e+03	7.9090e+03	9.3590e+03
f$_{16}$	**201.7178**	202.8841	202.9772	204.9818
f$_{17}$	944.1183	**819.8032**	876.3162	3.7177e+03
f$_{18}$	1.0708e+03	**935.6093**	949.7953	4.2394e+03
f$_{19}$	**3.6962e+03**	7.9335e+03	2.0738e+04	3.1437e+06
f$_{20}$	614.7672	614.4340	**614.3648**	615
f$_{21}$	**2.1325e+03**	2.6757e+03	2.7754e+03	9.3564e+03
f$_{22}$	**7.6664e+03**	8.5483e+03	8.7796e+03	1.1109e+04
f$_{23}$	**8.3404e+03**	9.0079e+03	8.9873e+03	1.0256e+04
f$_{24}$	**1.3035e+03**	1.3196e+03	1.3216e+03	1.3885e+03
f$_{25}$	**1.3992e+03**	1.4321e+03	1.4338e+03	1.4979e+03
f$_{26}$	1.5405e+03	**1.4144e+03**	1.4243e+03	1.6308e+03
f$_{27}$	**2.6168e+03**	2.7104e+03	2.7086e+03	2.8722e+03
f$_{28}$	5.2591e+03	**4.2623e+03**	4.4768e+03	1.3878e+04

(5) In the same time period, two adjacent buses have the same time interval

(6) Passengers arriving at the station at each time period obey even distribution

(7) The bus has a certain running time between two adjacent stops

(8) The ticket price is the same throughout the journey

Under the above assumptions, the optimal dispatching problem of a certain route is described as follows.

In a dispatching problem of public transportation, it is known that a bus route with a total mileage of L has J stations in total, and the operating time of vehicles in one day is $[t_{\text{early}}, t_{\text{night}}]$. The operating time can be divided into K time periods. The departure interval of the k-th time period is Δt_k. The bus models of this route are the same and arrive at the station on time. The bus fare for each passenger is n. The problem needs to consider both the operating profit of the bus company and the service level of the bus company (the length of waiting time for passengers). According to the passenger flow and operating conditions of each station in a day, it solves the vehicle operating timetable of this route, that is, the departure time interval of each period of operating time.

The purpose of the optimal bus dispatching system is to balance the interests of both the bus company and the passengers, from the perspective of the bus company and the passengers. From the perspective of the bus company, in order to ensure the lowest operating cost of the bus company, it is necessary to control the number of departures of the bus company. From the perspective of the passengers, it is necessary to consider that the long waiting time of the passengers will make the passengers impatient. Therefore, the following objective function is established:

$$F(\Delta t_k) = \alpha C_1 L \sum_{k=1}^{k} \frac{T_k}{\Delta t_k} + \beta C_2 \sum_{k=1}^{k} \sum_{j=1}^{J} m_k \left(\frac{\rho_{kj} \Delta t_k^2}{2} \right). \quad (14)$$

The objective function can be written as

$$F(\Delta t_k) = \alpha f_1(\Delta t_k) + \beta f_2(\Delta t_k), \quad (15)$$

$$f_1(\Delta t_k) = C_1 L \sum_{k=1}^{k} \frac{T_k}{\Delta t_k}, \quad (16)$$

$$f_2(\Delta t_k) = C_2 \sum_{k=1}^{k} \sum_{j=1}^{J} m_k \left(\frac{\rho_{kj} \Delta t_k^2}{2} \right), \quad (17)$$

where α is the weighting coefficient of the consumption of the bus company, β is the weighting coefficient of the consumption of the passengers, $f_1(\Delta t_k)$ represents the objective function cost of the bus company, and $f_2(\Delta t_k)$ represents the objective function of passenger loss. In equation (16), L is the total kilometers of the route, K is the number of time period, k is the time period, T_k represents the time length of the period, Δt_k is the departure interval of the k period, and C_1 is the cost of consumption per bus per kilometer. In equation (17), J is the number of stations, j is the station, and C_2 is the cost of loss per passenger waiting for a unit of time. m_k is the total number of departures in the k-th period and it is equal to the ratio of the length of the period to the departure interval. ρ_{kj} is the passenger arrival rate at the j-th station in the k-th period, and it is equal to the ratio of the number of passengers on the j-th station in the k-th period to the length of time.

Table 2: Performance evaluation for MCSCA, CBA, PCABC, and CPSO over CEC2013.

Function	MCSCA	CBA	PCABC	CPSO
$\mathbf{f}_1$	$1.0012e+04$	$5.8592e+04$	$5.4265e+04$	$\mathbf{8.2174e+03}$
$\mathbf{f}_2$	$\mathbf{1.5879e+08}$	$4.6005e+08$	$7.1326e+08$	$3.4631e+08$
$\mathbf{f}_3$	$\mathbf{8.9088e+12}$	$3.4373e+21$	$2.7048e+15$	$1.9944e+15$
$\mathbf{f}_4$	$\mathbf{4.6156e+04}$	$6.8506e+04$	$1.0192e+05$	$1.0063e+05$
$\mathbf{f}_5$	$\mathbf{1.8310e+03}$	$5.4879e+03$	$1.5106e+04$	$1.5106e+04$
$\mathbf{f}_6$	**278.9563**	$1.8592e+04$	$7.4003e+03$	$5.5797e+03$
$\mathbf{f}_7$	**320.5652**	$2.8640e+04$	$3.5116e+04$	576.36333
$\mathbf{f}_8$	**−679.0443**	−678.7539	−678.9390	−678.7612
$\mathbf{f}_9$	**−560.8198**	−542.1405	−557.5668	−556.9122
$\mathbf{f}_{10}$	$\mathbf{1.2036e+03}$	$1.1691e+04$	$6.9226e+03$	$1.7629e+03$
$\mathbf{f}_{11}$	**90.0472**	598.9505	468.3890	325.8092
$\mathbf{f}_{12}$	**227.4888**	675.6116	600.3618	402.0084
$\mathbf{f}_{13}$	**321.5240**	938.8861	702.9930	496.6388
$\mathbf{f}_{14}$	$\mathbf{5.9463e+03}$	$6.6365e+03$	$8.0827e+03$	$8.3116e+03$
$\mathbf{f}_{15}$	$\mathbf{6.8223e+03}$	$6.1357e+03$	$8.2723e+03$	$9.0474e+03$
$\mathbf{f}_{16}$	**201.7178**	203.1301	203.3737	204.0167
$\mathbf{f}_{17}$	**944.1183**	$1.2152e+03$	$1.3819e+03$	$1.1049e+03$
$\mathbf{f}_{18}$	$\mathbf{1.0708e+03}$	$1.2951e+03$	$1.4863e+03$	$1.2152e+03$
$\mathbf{f}_{19}$	$\mathbf{3.6962e+03}$	$7.5477e+05$	$1.1702e+06$	$1.2111e+06$
$\mathbf{f}_{20}$	**614.7672**	615	614.9994	615
$\mathbf{f}_{21}$	$\mathbf{2.1325e+03}$	$3.2044e+03$	$3.4450e+03$	$2.7068e+03$
$\mathbf{f}_{22}$	$\mathbf{7.6664e+03}$	$1.1060e+04$	$9.5938e+03$	$1.0378e+03$
$\mathbf{f}_{23}$	$\mathbf{8.3404e+03}$	$9.6684e+03$	$9.6065e+03$	$1.0629e+03$
$\mathbf{f}_{24}$	$\mathbf{1.3035e+03}$	$1.8042e+03$	$1.3712e+03$	$1.3198e+03$
$\mathbf{f}_{25}$	$\mathbf{1.3992e+03}$	$1.5600e+03$	$1.4826e+03$	$1.4109e+03$
$\mathbf{f}_{26}$	$\mathbf{1.5405e+03}$	$2.3145e+03$	$1.4961e+03$	$1.5641e+03$
$\mathbf{f}_{27}$	$\mathbf{2.6168e+03}$	$4.3918e+03$	$2.8685e+03$	$2.7369e+03$
$\mathbf{f}_{28}$	$\mathbf{5.2591e+03}$	$1.0705e+04$	$7.6122e+04$	$7.9081e+04$

Besides, there is a hidden constraint. If the bus company wants to achieve a profitable goal, it must make the total fare collected by the bus company greater than the bus company's minimum consumption cost. That is, the following formula must be satisfied:

$$n \times \frac{\sum_{k=1}^{K}\sum_{j=1}^{J} u_{kj}}{\sum_{k=1}^{K}\left(T_k/\Delta t_k\right) > C_1 L}, \tag{18}$$

where u_{kj} is the passenger arrival rate at the j-th station in the k-th period.

The code part of the constraint condition adopts the penalty function method. The penalty function is a commonly used method to deal with constraint conditions. It transforms constrained optimization problems into unconstrained optimization problems. If the fare collected by the bus company is less than its minimum consumption cost, the departure interval is increased, so that the consumption of the bus company is reduced. The departure interval is in minutes. If the condition is not met in one cycle, the time interval of a time period is randomly selected again and incremented by 1.

Assuming that there are 10 stations on the bus route, the total kilometers of the route is 8 km and the operating hours of the bus company are from 6 am to 9 pm every day. The variable x represents the departure interval. The passenger flow of each station in each time period is shown in Table 4.

The other parameters of the MCSCA algorithm are set as follows: $C_1 = 1$, $C_2 = 1$, $n = 1$, and the max of iteration equals 200. This application performs 10 optimization tests for the settings of 3 different parameters α and β and compares with SCA and MMSCA algorithms.

Situation 1. Let $\alpha = 0.2$, $\beta = 0.8$, the operation result is $\min(F(\Delta t_k)) = 3037.2$, $\Delta t_k = (2, 3, 4, 2, 3)$, and the unit is the minute. We obtained the optimized results' comparison as shown in Table 5 and curves of the objective function value as shown in Figure 2.

Situation 2. Let $\alpha = 0.5$, $\beta = 0.5$, the operation result is $\min(F(\Delta t_k)) = 3506.75$, $\Delta t_k = (2, 2, 3, 2, 4)$, and the unit is the minute. We obtained the optimized results' comparison as shown in Table 6 and curves of the objective function value as shown in Figure 3.

Situation 3. Let $\alpha = 0.8$, $\beta = 0.2$, the operation result is $\min(F(\Delta t_k)) = 2743.4033$, $\Delta t_k = (3, 4, 6, 3, 7)$, and the unit is the minute. We obtained the optimized results' comparison as shown in Table 7 and curves of the objective function value as shown in Figure 4.

The simulation result shows that, in the morning peak and evening peak, the passenger flow is large, and the departure frequency is higher, and when the passenger flow is low, the departure frequency is also reduced. The optimization result basically conforms to the actual situation. In actual application, the value of α and β depends on whether the decision-maker is more inclined to benefit the bus company or the consumer. The purpose

Table 3: Performance evaluation for MCSCA, DE, JADE, SHADE, and LSHADE over CEC2013.

Function	MCSCA	DE	JADE	SHADE	LSHADE
$\mathbf{f}_1$	**1.0012e+04**	1.8334e +04	1.6936e +04	1.3601e +04	1.1211e +04
$\mathbf{f}_2$	1.5879e +08	2.2830e +08	2.4714e +08	1.8775e +08	**1.5016e+08**
$\mathbf{f}_3$	8.9088e +12	2.0809e +12	2.0411e +11	**1.5216e+11**	9.3578e +11
$\mathbf{f}_4$	**4.6156e+04**	5.3020e +04	5.5735e +04	4.6472e +04	5.1199e +04
$\mathbf{f}_5$	1.8310e +03	3.8654e +03	3.5120e +03	2.7086e +03	**1.7677e+03**
$\mathbf{f}_6$	**278.9563**	1.4409e +03	1.0112e +03	1.2955e +03	351.1929
$\mathbf{f}_7$	320.5652	2.2144e +03	581.7276	**204.7736**	235.5335
$\mathbf{f}_8$	**−679.0443**	−678.9891	−679.0055	−679.0331	−679.0418
$\mathbf{f}_9$	−560.8198	−560.0210	−559.3174	**−561.1851**	−561.0092
$\mathbf{f}_{10}$	**1.2036e+03**	2.1623e +03	2.1844e +03	1.5209e +03	1.8154e +03
$\mathbf{f}_{11}$	90.0472	120.7334	114.3473	**73.1779**	80.3474
$\mathbf{f}_{12}$	227.4888	246.6691	247.9068	182.5079	**173.5435**
$\mathbf{f}_{13}$	321.5240	334.2122	292.0396	**283.9232**	330.5799
$\mathbf{f}_{14}$	**5.9463e+03**	7.1437e +03	7.2404e +03	6.5639e +03	6.5771e +03
$\mathbf{f}_{15}$	**6.8223e+03**	7.3093e +03	7.5481e +03	7.4742e +03	7.6433e +03
$\mathbf{f}_{16}$	**201.7178**	202.6536	202.5590	202.4048	202.3882
$\mathbf{f}_{17}$	944.1183	880.5067	**828.7338**	893.9824	908.6225
$\mathbf{f}_{18}$	1.0708e +03	1.0352e +03	1.0004e +03	1.0181e +03	**986.8915**
$\mathbf{f}_{19}$	**3.6962e+03**	7.3193e +03	9.0551e +03	6.2207e +03	6.2646e +03
$\mathbf{f}_{20}$	614.7672	614.8392	614.74552	**614.4299**	614.6522
$\mathbf{f}_{21}$	**2.1325e+03**	2.6780e +03	2.5064e +03	2.3300e +03	2.2905e +03
$\mathbf{f}_{22}$	**7.6664e+03**	8.5134e +03	8.8210e +03	8.1707e +03	8.5568e +03
$\mathbf{f}_{23}$	**8.3404e+03**	9.1687e +03	9.0751e +03	8.9719e +03	8.9474e +03
$\mathbf{f}_{24}$	1.3035e +03	1.3004e +03	1.3059e +03	1.3007e +03	**1.2964e+03**
$\mathbf{f}_{25}$	1.3992e +03	1.4012e +03	1.4021e +03	1.3996e +03	**1.3985e+03**
$\mathbf{f}_{26}$	1.5405e +03	1.5791e +03	1.5341e +03	1.5403e +03	**1.5075e+03**
$\mathbf{f}_{27}$	2.6168e +03	2.6780e +03	2.6312e +03	2.6169e +03	**2.6157e+03**
$\mathbf{f}_{28}$	**5.2591e+03**	8.5134e +03	5.3486e +03	5.5304e +03	5.3490e +03

Table 4: Number of passengers at various time sections and various stations.

Time	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
6 : 00–8:30	506	168	417	209	26	23	20	19	10	2
8 : 30–12 : 00	330	165	187	174	67	66	141	140	40	12
12 : 00–16 : 00	127	64	60	58	116	110	158	132	20	6
16 : 00–19 : 00	344	172	254	224	177	178	162	150	70	24
19 : 00–21 : 00	60	32	45	43	17	14	45	42	15	3

Table 5: Optimized results' comparison in $\alpha = 0.2$ and $\beta = 0.8$.

Algorithm	Average value	Best value	Worst value
MCSCA	3155.6667	2902.8	3408.4
SCA	3624.5055	2902.8	4198.8
MMSCA	3216.8	2862.8	3575.2

of designing three sets of different weight values is to prove that MCSCA has a better optimization effect in any case.

According to Tables 5–7, regardless of the values of α and β, the average value of the convergence accuracy of the MCSCA algorithm on this application is significantly better than the SCA algorithm and the MMSCA algorithm. And the difference between the best value and the worst value is significantly smaller than the other two algorithms. It indicates that the MCSCA is more stable. According to Figures 2–4, not only has the MCSCA algorithm a significant advantage in convergence accuracy but also the convergence speed is significantly better than the SCA algorithm and the MMSCA algorithm.

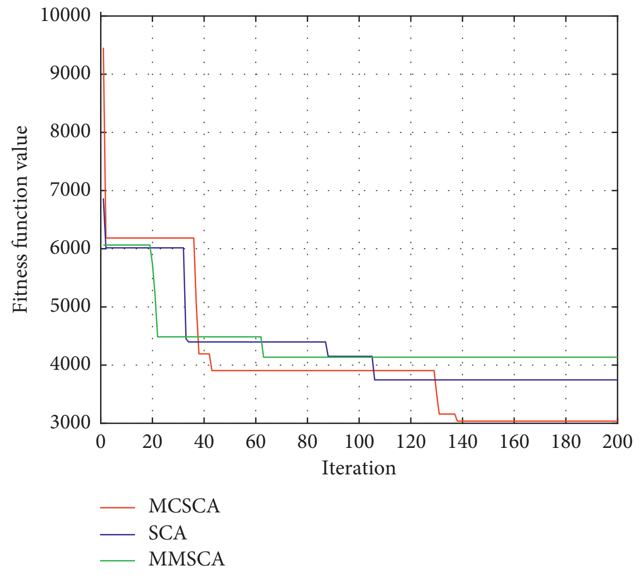


Figure 2: Curves of objective function F value in $\alpha = 0.2$, $\beta = 0.8$.

Table 6: Optimized results' comparison in $\alpha = 0.2$ and $\beta = 0.8$.

Algorithm	Average value	Best value	Worst value
MCSCA	3562.9946	3474	3671.4464
SCA	3678.7678	3474	3987
MMSCA	3601.9887	3525	3662.5

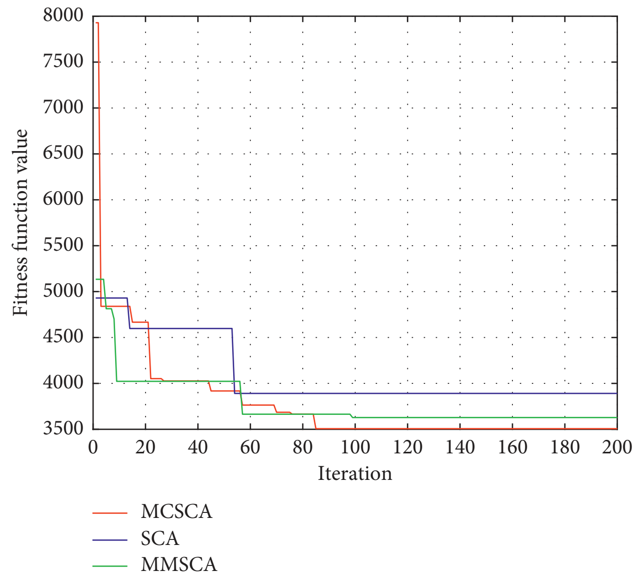


Figure 3: Curves of objective function F value in $\alpha = 0.5$, $\beta = 0.5$.

Table 7: Optimized results' comparison in $\alpha = 0.2$ and $\beta = 0.8$.

Algorithm	Average value	Best value	Worst value
MCSCA	3562.9946	3474	3671.4464
SCA	3678.7678	3474	3987
MMSCA	3601.9887	3525	3662.5

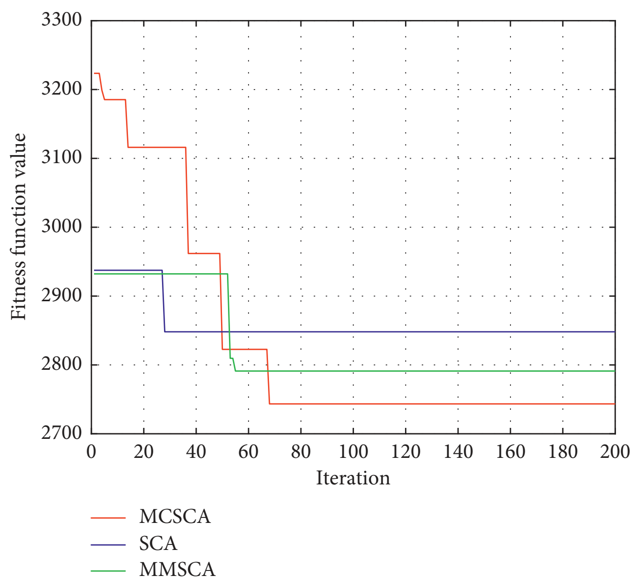


Figure 4: Curves of objective function F value in $\alpha = 0.8$, $\beta = 0.2$.

6. Conclusion

In this paper, we developed a compact-based SCA algorithm, CSCA, which can reduce the number of optimized variables and the required running memory. To improve the performance of our approach, we further proposed the MCSCA algorithm, which combines grouping strategy to speed up the convergence speed. Extensive empirical studies on benchmark CEC2013 demonstrate the effectiveness of our approach and the efficiency of our techniques. Furthermore, applying the MCSCA algorithm to the dispatching system of public transit vehicles can get a minimum cost value than that of SCA and MMSCA algorithms, which further testify the effectiveness and superiority of the proposed MCSCA algorithm. A possible future work is considered to continue the improvement of the SCA algorithm and apply it to more practical engineering problems.

Data Availability

Public CEC2013 was used. No data were used to support this study.

Conflicts of Interest

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

References

[1] S. Mirjalili, “SCA: a sine cosine algorithm for solving optimization problems,” *Knowledge-Based Systems*, vol. 96, pp. 809–818, 2016.

[2] S. C. Chu, H. C. Huang, J. F. Roddick et al., “Overview of algorithms for swarm intelligence,” in *Proceedings of the International Conference on Computational Collective Intelligence*, pp. 28–41, Da Nang, Vietnam, November 2011.

[3] C.-l. Sun, J.-c. Zeng, and J.-s. Pan, “An improved vector particle swarm optimization for constrained optimization problems,” *Information Sciences*, vol. 181, no. 6, pp. 1153–1163, 2011.

[4] W. Hao, M. Liang, C. Sun et al., “Multiple-strategy learning particle swarm optimization for large-scale optimization problems,” *Complex & Intelligent Systems*, vol. 7, 2020.

[5] P. Hu, J. S. Pan, and S. C. Chu, “Improved binary grey wolf optimizer and Its application for feature selection,” *Knowledge-Based Systems*, p. 105746, 2020.

[6] Q. W. Chai, S. C. Chu, J. S. Pan et al., “Applying adaptive and self assessment fish migration optimization on localization of wireless sensor network on 3-D terrain,” *Journal of Information Hiding and Multimedia Signal Processing*, vol. 11, no. 2, pp. 90–102, 2020.

[7] S. C. Chu, P. W. Tsai, and J. S. Pan, “Cat swarm optimization,” in *Proceedings of the Pacific Rim International Conference on Artificial intelligence*, pp. 854–858, Berlin, Germany, August 2006.

[8] P. W. Tsai, J. S. Pan, S. M. Chen et al., “Parallel cat swarm optimization,” in *Proceedings of the 2008 international conference on machine learning and cybernetics*, pp. 3328–3333, Kunming, China, July 2008.

[9] P.-W. Tsai, J.-S. Pan, S.-M. Chen, and B.-Y. Liao, “Enhanced parallel cat swarm optimization based on the Taguchi method,” *Expert Systems with Applications*, vol. 39, no. 7, pp. 6309–6319, 2012.

[10] H. Wang, Z. Wu, S. Rahnamayan, H. Sun, Y. Liu, and J.-s. Pan, “Multi-strategy ensemble artificial bee colony algorithm,” *Information Sciences*, vol. 279, pp. 587–603, 2014.

[11] P. W. Tsai, J. S. Pan, B. Y. Liao et al., “Enhanced artificial bee colony optimization,” *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 12, pp. 5081–5092, 2009.

[12] Y. Tan, C. Yu, S. Zheng, and K. Ding, “Introduction to fireworks algorithm,” *International Journal of Swarm Intelligence Research*, vol. 4, no. 4, pp. 39–70, 2013.

[13] M. A. Tawhid and V. Savsani, “Multi-objective sine-cosine algorithm (MO-SCA) for multi-objective engineering design problems,” *Neural Computing and Applications*, vol. 31, no. 2, pp. 915–929, 2019.

[14] A. I. Hafez, H. M. Zawbaa, E. Emary et al., “Sine cosine optimization algorithm for feature selection,” in *Proceedings of the 2016 International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, pp. 1–5, Sinaia, Romania, August 2016.

[15] K. S. Reddy, L. K. Panwar, B. Panigrahi, and R. Kumar, “A new binary variant of sine-cosine algorithm: development and application to solve profit-based unit commitment problem,” *Arabian Journal for Science and Engineering*, vol. 43, no. 8, pp. 4041–4056, 2018.

[16] M. Abd Elaziz, D. Oliva, and S. Xiong, “An improved opposition-based sine cosine algorithm for global optimization,” *Expert Systems with Applications*, vol. 90, pp. 484–500, 2017.

[17] D. Bairathi and D. Gopalani, “Opposition-based sine cosine algorithm (OSCA) for training feed-forward neural networks,” in *Proceedings of the 2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pp. 438–444, Jaipur, India, December 2017.

[18] K. Chen, F. Zhou, L. Yin, S. Wang, Y. Wang, and F. Wan, “A hybrid particle swarm optimizer with sine cosine acceleration coefficients,” *Information Sciences*, vol. 422, pp. 218–241, 2018.

[19] M. Issa, A. E. Hassanien, D. Oliva, A. Helmi, I. Ziedan, and A. Alzohairy, “ASCA-PSO: adaptive sine cosine optimization

algorithm integrated with particle swarm for pairwise local sequence alignment," *Expert Systems with Applications*, vol. 99, pp. 56–70, 2018.

[20] S. Bureerat and N. Pholdee, "Adaptive sine cosine algorithm integrated with differential evolution for structural damage detection," in *Proceedings of the International Conference on Computational Science and its Applications*, pp. 71–86, Trieste, Italy, July 2017.

[21] M. Abd Elaziz, A. A. Ewees, D. Oliva et al., "A hybrid method of sine cosine algorithm and differential evolution for feature selection," in *Proceedings of the International Conference on Neural information Processing*, pp. 145–155, Guangzhou, China, July 2017.

[22] H. Nenavath and R. K. Jatoth, "Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking," *Applied Soft Computing*, vol. 62, pp. 1019–1043, 2018.

[23] C. Zhou, L. Chen, Z. Chen, X. Li, and G. Dai, "A sine cosine mutation based differential evolution algorithm for solving node location problem," *International Journal of Wireless and Mobile Computing*, vol. 13, no. 3, pp. 253–259, 2017.

[24] Q. Yang, S. C. Chu, J. S. Pan et al., "Sine cosine algorithm with multigroup and multistrategy for solving CVRP," *Mathematical Problems in Engineering*, vol. 2020, Article ID 8184254, 10 pages, 2020.

[25] M. Abd Elfattah, S. Abuelenin, A. E. Hassanien et al., "Handwritten Arabic manuscript image binarization using sine cosine optimization algorithm," in *Proceedings of the International Conference on Genetic and Evolutionary Computing*, pp. 273–280, Fuzhou, China, November 2016.

[26] J. S. Pan, T. T. Nguyen, A. H. Vu et al., "A new optimization based on flower and sine cosine for saving fuel consumption problem," *Journal of Computers and Applied Science Education*, vol. 5, no. 3, 2018.

[27] E. Mostafa, M. Abdel-Nasser, and K. Mahmoud, "Performance evaluation of metaheuristic optimization methods with mutation operators for combined economic and emission dispatch," in *Proceedings of the 2017 Nineteenth International Middle East Power Systems Conference (MEPCON)*, pp. 1004–1009, Cairo, Egypt, December 2017.

[28] P. P. Singh, R. Bains, G. Singh et al., "Comparative analysis on economic load dispatch problem optimization using moth flame optimization and sine cosine algorithms," *International Journal Of Advance Research And Innovative Ideas In Education*, vol. 3, no. 2, pp. 65–75, 2017.

[29] B. Rout, B. B. Pati, and S. Panda, "Modified SCA algorithm for SSSC damping controller design in power system," *ECTI Transactions on Electrical Engineering, Electronics, and Communications*, vol. 16, no. 1, pp. 46–63, 2018.

[30] N. Kumar, I. Hussain, B. Singh, and B. K. Panigrahi, "Single sensor-based MPPT of partially shaded PV system for battery charging by using Cauchy and Gaussian sine cosine optimization," *IEEE Transactions on Energy Conversion*, vol. 32, no. 3, pp. 983–992, 2017.

[31] J. Wang, W. Yang, P. Du, and T. Niu, "A novel hybrid forecasting system of wind speed based on a newly developed multi-objective sine cosine algorithm," *Energy Conversion and Management*, vol. 163, pp. 134–150, 2018.

[32] U. Raut and S. Mishra, "Power distribution network reconfiguration using an improved sine–cosine algorithm-based meta-heuristic search," *Soft Computing for Problem Solving*, pp. 1–13, 2019.

[33] L. Khriddi, N. E. Akkad, H. Satori et al., "Clustering method and sine cosine algorithm for image segmentation," *Evolutionary Intelligence*, vol. 1, pp. 1–14, 2021.

[34] S. Mishra, S. Gupta, and A. Yadav, "Design and application of controller based on sine-cosine algorithm for load frequency control of power system," *Intelligent Systems Design and Applications*, 2020.

[35] X. Xue and J. Chen, "A compact co-firefly algorithm for matching ontologies," in *Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 2629–2632, Xiamen, China, September 2019.

[36] X. Xue and J. Chen, "Matching biomedical ontologies through compact differential evolution algorithm," *Systems Science & Control Engineering*, vol. 7, no. 2, pp. 85–89, 2019.

[37] X. Xue, J. Chen, and D. Chen, "Matching biomedical ontologies through compact hybrid evolutionary algorithm," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 10, no. 1, pp. 110–117, 2019.

[38] S. C. Chu, X. Xue, J. S. Pan et al., "Optimizing ontology alignment in vector space," *Journal of Internet Technology*, vol. 21, no. 1, pp. 15–22, 2020.

[39] J. F. Chang, J. F. Roddick, J. S. Pan et al., "A parallel particle swarm optimization algorithm with communication strategies," *Journal of Information Science and Engineering*, vol. 21, pp. 809–818, 2005.

[40] S. C. Chu, J. F. Roddick, J. S. Pan et al., "Parallel ant colony systems," in *Proceedings of the International Symposium on Methodologies for Intelligent Systems*, pp. 279–284, Knoxville, TE, USA, October 2003.

[41] Z.-G. Du, J.-S. Pan, S.-C. Chu, H.-J. Luo, and P. Hu, "Quasi-affine transformation evolutionary algorithm with communication schemes for application of RSSI in wireless sensor networks," *IEEE Access*, vol. 8, pp. 8583–8594, 2020.

[42] H. Wang, S. Rahnamayan, H. Sun, and M. G Omran, "Gaussian bare-bones differential evolution," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 634–647, 2013.

[43] W. J. Cody, "Rational Chebyshev approximations for the error function," *Mathematics of Computation*, vol. 23, no. 107, 631 pages, 1969.

[44] P. C. Song, J. S. Pan, and S. C. Chu, "A parallel compact cuckoo search algorithm for three-dimensional path planning," *Applied Soft Computing*, p. 106443, 2020.

[45] D. Jo, B. Yu, H. Jeon et al., "Image-to-image learning to predict traffic speeds by considering area-wide spatio-temporal dependencies," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1188–1197, 2018.

[46] W.-K. Lai, T.-H. Kuo, and C.-H. Chen, "Vehicle speed estimation and forecasting methods based on cellular floating vehicle data," *Applied Sciences*, vol. 6, no. 2, p. 47, 2016.

[47] S. Chen, Z. Sun, and B. Bryan, "Traffic monitoring using digital sound field mapping," *IEEE Transactions on Vehicular Technology*, vol. 50, no. 6, pp. 1582–1589, 2001.

[48] W. Hu, X. Xiao, D. Xie, T. Tan, and S. Maybank, "Traffic accident prediction using 3-d model-based vehicle tracking," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 3, pp. 677–694, 2004.

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

[50] 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.

[51] Z. Hassan and R. Mahmud, “Locating stations of public transportation vehicles for improving transit accessibility,” *Transport*, vol. 22, 2007.

[52] B. A. Weigel, F. Southworth, and M. D. Meyer, “Calculators to estimate greenhouse gas emissions from public transit vehicles,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2143, no. 1, pp. 125–133, 2010.

[53] M. Wei, W. Jin, W. Fu et al., “Improved ant colony algorithm for multi-depot bus scheduling problem with route time constraints,” in *Proceedings of the 2010 8th World Congress on Intelligent Control and Automation*, pp. 4050–4053, Jinan, China, July 2010.

[54] F. Cevallos and F. Zhao, “A genetic algorithm for bus schedule synchronization,” in applications of advanced technology in transportation,” in *Proceedings of the Ninth International Conference of ASCE*, Harbin, China, August 2006.

[55] C. Wang, H. Shi, and X. Zuo, “A multi-objective genetic algorithm based approach for dynamical bus vehicles scheduling under traffic congestion,” *Swarm and Evolutionary Computation*, vol. 54, p. 100667, 2020.

[56] 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*, Dalian, China, October 2008.

[57] E. Mininno, F. Neri, F. Cupertino et al., “Compact differential evolution,” *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 32–54, 2010.

[58] S. Gao, M. Zhou, Y. Wang et al., “Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 2, pp. 601–614, 2018.

[59] J. Zhang and S. Member, “JADE: adaptive differential evolution with optional external archive,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

[60] R. Tanabe and A. Fukunaga, “Success-history based parameter adaptation for Differential Evolution,” in *Proceedings of the 2013 IEEE Congress on Evolutionary Computation*, pp. 71–78, Cancun, Mexico, June 2013.

[61] R. Tanabe and A. Fukunaga, “Improving the search performance of SHADE using linear population size reduction,” in *Proceedings of the 2014 IEEE Congress on Evolutionary Computation*, pp. 1658–1665, Beijing, China, July 2014.