

Research Article

Sine Cosine Algorithm with Multigroup and Multistrategy for Solving CVRP

Qingyong Yang, Shu-Chuan Chu, Jeng-Shyang Pan , and Chien-Ming Chen

College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, Shandong 266590, China

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

Received 3 February 2020; Accepted 15 February 2020; Published 27 March 2020

Guest Editor: Chi-Hua Chen

Copyright © 2020 Qingyong Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Sine Cosine Algorithm (SCA) has been proved to be superior to some existing traditional optimization algorithms owing to its unique optimization principle. However, there are still disadvantages such as low solution accuracy and poor global search ability. Aiming at the shortcomings of the sine cosine algorithm, a multigroup multistrategy SCA algorithm (MMSCA) is proposed in this paper. The algorithm executes multiple populations in parallel, and each population executes a different optimization strategy. Information is exchanged among populations through intergenerational communication. Using 19 different types of test functions, the optimization performance of the algorithm is tested. Numerical experimental results show that the performance of the MMSCA algorithm is better than that of the original SCA algorithm, and it also has some advantages over other intelligent algorithms. At last, it is applied to solving the capacitated vehicle routing problem (CVRP) in transportation. The algorithm can get better results, and the practicability and feasibility of the algorithm are also proved.

1. Introduction

With the rapid development of human society and the continuous improvement of computer technology, people are facing more and more optimization problems in scientific research and engineering practice. The optimization problems can be understood as choosing a most suitable solution among many options for solving a complex problem without changing the constraints. However, with the increasing scale of the problem, the computing time also shows an exponential growth trend. When solving large-scale optimization problems, traditional algorithms seem powerless. For quite some time, people have been working hard to find new ways to better solve optimization problems. With the robustness, parallelism, and intelligence of its methods, computational intelligence (CI) can well solve many optimization problems, which has attracted the attention of many scholars. Also, computational intelligence (CI) plays an important role in scientific research and production practice.

Computational intelligence is a general term for a class of algorithms that are inspired by natural and human wisdom. Computational intelligence includes three basic areas: fuzzy computing [1–3], neural networks [4, 5], and evolutionary computing [6]. Evolutionary computing is one of the important research fields of intelligent computing. Up to now, scholars have put forward many excellent intelligent optimization algorithms. Typical examples are as follows: genetic algorithm (GA) by simulating the genetic evolution of the biological world [7]; particle swarm optimization (PSO) by simulating bird swarm predation [8–10]; and ant colony algorithm (ACO) by simulating ant colony foraging behavior [11, 12]. There are many intelligent optimization algorithms proposed by simulating other phenomena in nature, such as cat swarm algorithm (CSO) [13–15], artificial bee colony algorithm (ABC) [16, 17], differential evolution algorithm (DE) [18–21], multiverse optimizer (MVO) [22, 23], flower pollination algorithm [24, 25], gray wolf algorithm (GWO) [26–28], pigeon-inspired algorithm (PIO)

[29, 30], bat algorithm (BA) [31, 32], symbiotic organism search algorithm (SOS) [33, 34], and QUATRE [35–37].

At present, there are so many optimization algorithms, which can solve some complex optimization problems well. Why do we need so many optimization algorithms? As demonstrated by the No Free Lunch (NFL) [38] proposed by Wolpert and Macready, no single optimization algorithm is applicable to all problems. Inspired by this, a new intelligent optimization algorithm was proposed by Australian scholar Mirjalili in 2016, which is called Sine Cosine Algorithm (SCA) [39]. The SCA algorithm iterates through the properties of the sine and cosine functions to achieve optimization. It has fewer parameter settings, is easy to implement, and has a strong optimization ability. It has been proved that it is better than PSO algorithm, genetic algorithm (GA), and firefly algorithm (FA) in convergence with accuracy and speed [39].

With the rapid development of software and hardware, parallel computing has become a form of high-performance computing. In evolutionary computing, parallelism often represents the iterative updating of multiple populations at the same time. The advantage of this method is to ensure population diversity, to further improve the search ability and performance of the algorithm. Especially when solving complex optimization problems, parallelizing the algorithm is an effective way to improve the efficiency and accuracy of the algorithm. At present, many existing algorithms have successfully applied the parallel mechanism, such as parallel PSO [40], parallel ACO [41], and parallel QUATRE [42]. Inspired by this, this paper introduces the multigroup and multistrategy optimization mechanism to further improve the SCA algorithm. It is called MMSCA. When the algorithm is solved, multiple populations execute in parallel, and each population adopts different updating strategies. By comparing the results of test functions, MMSCA is better than SCA in solving accuracy and convergence ability and has certain advantages over other intelligent optimization algorithms.

At present, transportation is gradually moving towards intelligent transportation. One of the important components of an intelligent transportation system is wireless sensor network technology (WSN). With the help of wireless sensor network technology [43–48], the safety, unblocking, and environmental protection problems that plague modern transportation can be fundamentally alleviated. However, there are still many transportation problems to be solved. The capacitated vehicle routing problem (CVRP) in transportation problems is taken as the application object of MMSCA in this paper. On the one hand, the practicability of the algorithm is further confirmed. On the other hand, it brings a new solution to CVRP problem.

The remaining chapters of the article are arranged as follows: the original SCA algorithm and related improvements around it are reviewed in Section 2. Section 3 describes the MMSCA design process in detail. The results of numerical experiments are presented and discussed in Section 4. How to apply MMSCA to CVRP is covered in Section 5. At last, Section 6 is a summary of the article and an outlook for future work.

2. Related Works

2.1. Original Sine Cosine Algorithm. In the SCA algorithm, a population X is initialized first, where the population X contains m individuals, and each individual has n dimensions. The fitness function is used to evaluate each individual, and the best individual position $P_b = (P_{b1}, P_{b2}, \dots, P_{bn})^T$ among all the current individuals is selected. Then, the algorithm iterates according to the set number of iterations. During the iteration, the position of the individual is updated. The update formula is as follows:

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

where X_i^t represents the position of the i th ($i = 1, 2, \dots, m$) individual in the t th iteration; $r_2 \in [0, 2\pi]$, $r_3 \in [0, 2]$, and $r_4 \in [0, 1]$ are three random numbers. P_b^t represents the best individual position after t iterations. r_1 is the control parameter, which balances the exploration and exploitation stages of the algorithm. The updated formula is as follows:

$$r_1 = a - a \frac{t}{T}, \quad (2)$$

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

Since equation (1) uses sine and cosine functions, the algorithm is named Sine Cosine Algorithm (SCA). Four important parameters play different roles in the process of iterative optimization of the algorithm: the parameter r_1 is used as a control parameter to determine the moving direction of the individual position in the next iteration; the parameter r_2 determines how far to move in the next iteration; as a random weight, parameter r_3 affects the role of the optimal individual in the next iteration; the parameter r_4 enables the algorithm to iteratively update the individual positions in the sine and cosine parts with equal probability. Like other intelligent optimization algorithms, there need to be two phases: local exploitation and global exploration. In SCA algorithm, when $r_1 \sin(r_2)$ or $r_1 \cos(r_2)$ function value is between -1 and 1 , the algorithm performs local exploitation; when $r_1 \sin(r_2)$ or $r_1 \cos(r_2)$ function value is less than -1 or greater than 1 , the algorithm performs global exploration.

2.2. Related Improvements to the SCA Algorithm. With its unique optimization principles and theoretical advantages, the SCA algorithm has been proved to be superior to PSO, genetic algorithm (GA), and other existing algorithms in the most of the examples and has been applied in many practical problems. However, there are still some disadvantages such as low solution accuracy and poor global search ability. This is because when the SCA algorithm updates the individual location, the next generation location of the individual is only expanded around the current optimal individual location. In particular, with the decrease of the diversity of the population in the later period of the algorithm, the global

search ability becomes worse, which easily leads to the algorithm falling into the local optimal trap.

In view of the shortcomings of the SCA algorithm, scholars at home and abroad have made some improvements. Some scholars introduced search strategies of other algorithms or improved the setting of the parameters of the SCA algorithm itself: Long et al. [49] introduced nonlinear weight factor and inertial weight based on Gaussian distribution, respectively, to improve the ability to avoid falling into the trap of local optimization and convergence speed; Qu et al. [50] proposed an improved SCA algorithm based on neighborhood search and Greedy Levy Mutation to better balance the phases of local exploitation and global exploration of the algorithm. Some scholars have also integrated the SCA algorithm with other algorithms to further improve the optimization ability of the algorithm: Chegini et al. [51] proposed to mix the SCA algorithm with the PSO algorithm to improve the global search capability of the algorithm and the calculation accuracy; Nenavath and Jatoth [52] mixed the SCA algorithm and the DE algorithm to further improve the convergence speed of the algorithm and the ability to avoid local optimization.

There are many improvements to the SCA algorithm, and the improvement methods can be summarized into the abovementioned two categories. Aiming at the shortcomings of the SCA algorithm, this paper adopts a multigroup and multistrategy optimization mechanism to further improve the accuracy of the solution and the global search ability of the algorithm. The next section will detail how to improve the SCA algorithm.

3. Sine Cosine Algorithm with Multigroup and Multistrategy (MMSCA)

This section describes in detail how to apply the multigroup and multistrategy mechanism to improve the Sine Cosine Algorithm (SCA).

3.1. Algorithm Idea. The use of a single population in basic SCA algorithm inevitably leads to the problem of falling into the local optimal trap. At the same time, when the algorithm updates individual location in each iteration, the location of candidate individuals is only expanded around the global optimal location, which results in the strong local search capability of the basic SCA algorithm and the weak global search capability. Based on the shortcomings of the abovementioned basic SCA algorithm, the concept of multigroup and multistrategy is introduced to enhance the optimization ability of SCA algorithm. In the multigroup multistrategy SCA algorithm (MMSCA), each population has the same number of individuals and uses a different update strategy. Before a certain number of generations, each population is optimized according to its own strategy, and the population does not interfere with each other. After reaching a certain number of generations, the populations communicate with each other and perform the optimal individual migration between the populations. The worst individual in each population will be replaced by the

optimal individual in the other populations. Repeat the abovementioned steps until the algorithm iterates to the maximum number of iterations or the optimal solution is found.

In this article, the update strategies are divided into two types: rand strategy and best strategy. In the rand strategy, the roulette wheel selection scheme in the genetic algorithm (GA) is borrowed, and the optimal individual in the group is no longer selected as the solution target of other individuals. Instead, the algorithm uses multiple roulette methods to choose the next generation of solving goals to increase the diversity of population solutions. In this way, the ability of global optimization can be enhanced and the possibility of local optimal trap will be increased. In the best strategy, the best individual in all populations is still the solution target. The purpose of doing this is 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.

The update formula is as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t + r_1 \times \sin(r_2) \times |r_3 \times P_b^t - X_{i,j}^t|, & r_4 < 0.5, \\ X_{i,j}^t + r_1 \times \cos(r_2) \times |r_3 \times P_b^t - X_{i,j}^t|, & r_4 \geq 0.5, \end{cases} \quad (3)$$

where $X_{i,j}^t$ represents the position of the j th individual in the i th population in the t th iteration and P_b^t represents the selected target individual after multiple generations of roulette wheel selection in the rand strategy executing population. Also, in the population that executes the best strategy, it represents the globally optimal individual.

3.2. Algorithm Steps. Based on the abovementioned description, the specific steps of the MMSCA algorithm are given below:

Step 1: randomly generate N (N is an even number) populations of size m , each population having n dimensions. Initialize the maximum number of iterations, independent evolution algebra, iteration counter etc.

Step 2: evaluate each population and calculate the fitness value of each individual in each population. The population with odd group numbers executes the rand strategy, and the target individual is selected using the roulette wheel selection scheme; the population with even group numbers executes the best strategy, sorts and selects the best individual.

Step 3: use formula (3) to update the individual positions in the population.

Step 4: if the communication conditions are met, interpopulation communication is conducted. The worst individual in a population is replaced by the best individual in another population. If the communication condition is not met, return to step 2.

Step 5: if the algorithm iterates to the maximum number of iterations or the optimal individual is found

to meet the convergence conditions, the algorithm is terminated; otherwise return to step 2.

The algorithm flow chart is shown in Figure 1.

4. Results and Discussion

4.1. Comparison with SCA Algorithm. In order to test the influence of multigroup and multistrategy mechanism on SCA algorithm, 19 standard test functions ($f_1 - f_7$ as unimodal benchmark function, $f_8 - f_{13}$ as multimodal benchmark function, and $f_{14} - f_{19}$ as composite benchmark function) in the SCA algorithm literature [39] are used for testing. The minimum value of test function $f_1 - f_{15}$ is 0 except f_8 , f_8 and $f_{16} - f_{19}$ are -2094.9145 , $-1.0360.398$, 3 , -3.86 , respectively. For the fairness of comparison, the maximum number of iterations is 1000 and each population size is set to 30. In the proposed algorithm, the independent evolutionary algebra is set to 50 and the number of groups is two. For each test function, each algorithm is run independently for 30 times and the average value is calculated. Table 1 records the experimental results, with the best optimization results in bold. All simulation experiments are implemented on the same computer, and the programming tool is Matlab2018a.

As shown in the abovementioned table, in the test functions $f_1 - f_{15}$, except f_8 , the MMSCA algorithm has better solution accuracy than the original SCA algorithm. For function $f_{16} - f_{19}$, both algorithms show the same optimization performance, and the results are close to or reach the expected minimum. In general, the optimization ability of the SCA algorithm has been improved to some extent after the introduction of the multigroup and multistrategy mechanism. Figure 2 shows the convergence curves of the two algorithms on the unimodal functions f_2 and f_3 , the multimodal functions f_9 and f_{11} , and the composite function f_{14} . As can be seen from the figure, MMSCA algorithm has also significantly improved the convergence speed compared with SCA algorithm.

4.2. Comparison with Other Intelligent Optimization Algorithms. To further verify the effectiveness of the MMSCA algorithm, this article also compares it with other three intelligent optimization algorithms: the PSO algorithm, the parallel PSO algorithm, and the DE algorithm. The population size is set to 30. 1000 is the maximum number of iterations. For each test function, each algorithm is tested 30 times and the average value is obtained. The other parameters of other algorithms are shown in Table 2. The test results of MMSCA and other three intelligent optimization algorithms on 19 test functions are shown in Table 3. The bold type in the table indicates the best results of all comparison algorithms.

As shown in Table 3, compared with the PSO algorithm, PPSO algorithm, and DE algorithm, the MMSCA algorithm has obtained better results on 11 test functions. The MMSCA algorithm has the same test results on the test functions f_6 and $f_{18} - f_{19}$ as the other three algorithms, and they all obtain the expected minimum. However, the test results on

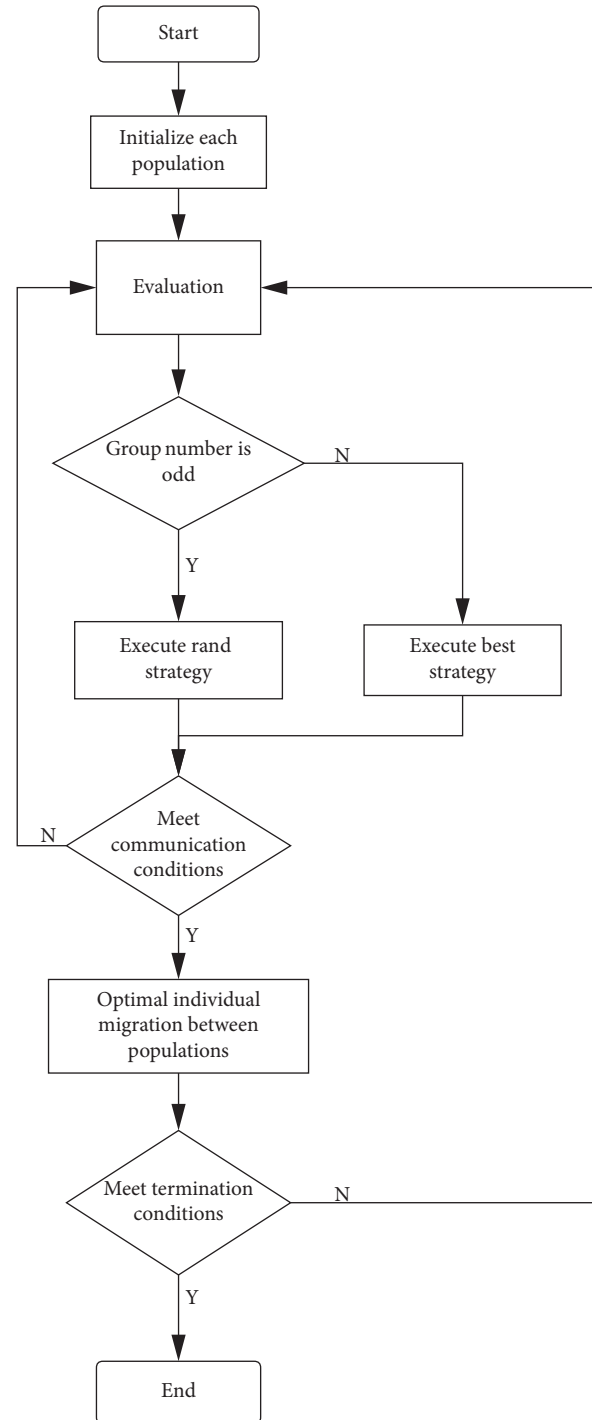


FIGURE 1: Flowchart of sine cosine algorithm with multigroup and multistrategy.

the test functions f_6 and f_{13} are significantly worse than those of the other three optimization algorithms; the MMSCA algorithm obtains better results on the test function f_{11} , and obtains a minimum value of 0.

Based on the comparison results of the two subsections, compared with the original SCA algorithm, the optimization performance of the algorithm is improved after the introduction of multigroup and multistrategy mechanism;

TABLE 1: Comparison of test results with SCA algorithm.

Function	SCA	MMSCA
f_1	$5.86E-28$	2.95E-30
f_2	$2.19E-17$	8.02E-20
f_3	$1.46E-08$	6.73E-30
f_4	$1.31E-07$	1.56E-08
f_5	$7.05E+00$	7.02E+00
f_6	$3.73E-01$	2.98E-01
f_7	$1.82E-03$	1.83E-04
f_8	-2.22E+03	$-2.29E+03$
f_9	$6.83E-01$	3.30E-06
f_{10}	$6.48E-14$	2.91E-14
f_{11}	$2.34E-02$	0.00E+00
f_{12}	$6.92E-02$	5.60E-02
f_{13}	$2.68E-01$	2.59E-01
f_{14}	$1.86E+00$	9.98E-01
f_{15}	$9.20E-04$	6.85E-04
f_{16}	-1.03E+00	$-1.03E+00$
f_{17}	3.99E-01	$3.99E-01$
f_{18}	3.00E+00	$3.00E+00$
f_{19}	-3.86E+00	$-3.86E+00$

compared with other intelligent optimization algorithms, it also shows strong competitiveness. So as to further study the feasibility and practicability of the MMSCA algorithm in processing practical problems, this paper selects the capacitated vehicle routing problem (CVRP) in transportation as the application object. The next section will describe in detail how to apply the MMSCA algorithm and solve it.

5. Applying MMSCA to Solve CVRP

VRP (Vehicle Routing Problem) has always been the focus of research in the field of transportation. At the same time, there are many applications in real life, including express delivery, air transportation, bus route planning, and so on. VRP was first proposed by scholars Dantzing and Ramser [53] in the late 1950s. Since it was proposed, it has received extensive attention from scholars at home and abroad, and scholars have also carried out a lot of fruitful work on it. After years of research and development, more models of VRP have been derived, and the solving algorithms have become more diverse. Solution algorithms can be divided into two categories: the first type is accurate algorithm and the second is approximate algorithm or heuristic algorithm. Since the VRP problem has been proved to be an NP-Hard problem, the second kind of solution algorithm is the main method to solve it [54]. The GA algorithm, the ACO algorithm, the SA (simulated annealing) algorithm, the PSO algorithm, and other algorithms have been successfully applied to solve VRP problems and achieved good results. In this paper, CVRP (Capacitated Vehicle Routing Problem) is selected as the application object of MMSCA algorithm to further verify the practicability of the algorithm.

5.1. CVRP Problem Description and Model. Problem description: a central warehouse has k distribution vehicles, and the maximum load capacity of each distribution vehicle

is Q . Now, we need to transport and distribute materials to n customers (nodes), how to arrange a certain number of vehicles to meet the needs of these customers, and make the total driving distance of vehicles shortest. The following constraints must also be met:

- (1) All delivery vehicles start from the central warehouse and eventually return to the central warehouse
- (2) Each delivery vehicle is not allowed to exceed the maximum load capacity Q
- (3) Each customer can only be served once, and each vehicle can only take one route

For the convenience of description, the following symbols are defined:

D : total length of all vehicle paths

k : total number of vehicles

n : total number of customers

v : vehicle number (1, 2, ..., k)

$$x_{ijv} = \begin{cases} 1, & \text{if vehicle } v \text{ travels from customer } i \text{ to customer } j, \\ 0, & \text{else,} \end{cases}$$

$$y_{iv} = \begin{cases} 1, & \text{if customer } i \text{ needs to be fulfilled by vehicle } v, \\ 0, & \text{else,} \end{cases} \quad (4)$$

c_{ij} : the distance between customer point i and j

g_i : the demand of the i -th customer ($i = 1, 2, \dots, n$)

Q : the maximum loading capacity of the vehicle

The central warehouse number is 0, and the customer number is i ($i = 1, 2, \dots, n$). The CVRP mathematical model is expressed as follows:

Objective function:

$$\min D = \sum_{i=0}^n \sum_{j=0}^n \sum_{v=0}^k c_{ij} x_{ijv}. \quad (5)$$

Constraint conditions:

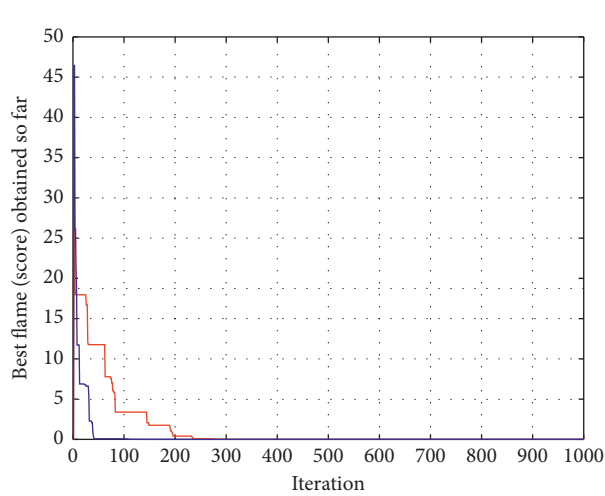
$$\sum_{i=0}^n x_{ijv} = y_{jv}, \quad j = 1, 2, \dots, n; v = 1, 2, \dots, k, \quad (6)$$

$$\sum_{j=0}^n x_{ijv} = y_{iv}, \quad i = 1, 2, \dots, n; v = 1, 2, \dots, k, \quad (7)$$

$$\sum_{i=0}^n g_i y_{iv} \leq Q, \quad v = 1, 2, \dots, k, \quad (8)$$

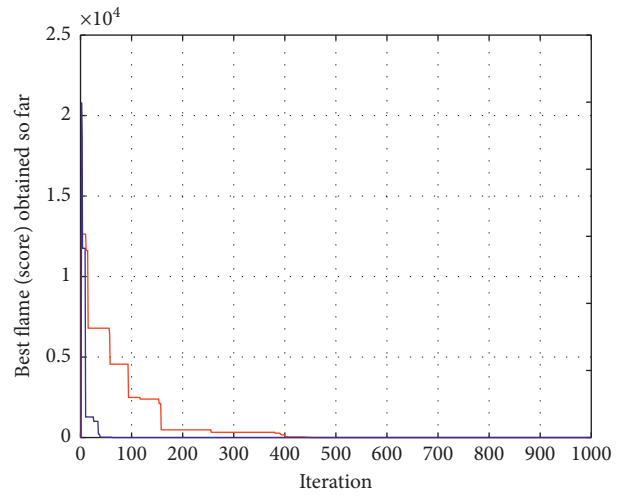
$$\sum_{v=1}^k y_{iv} = 1, \quad i = 1, 2, \dots, n. \quad (9)$$

Among them, equations (6) and (7) are traffic flow constraints, requiring a car to leave a customer node after it



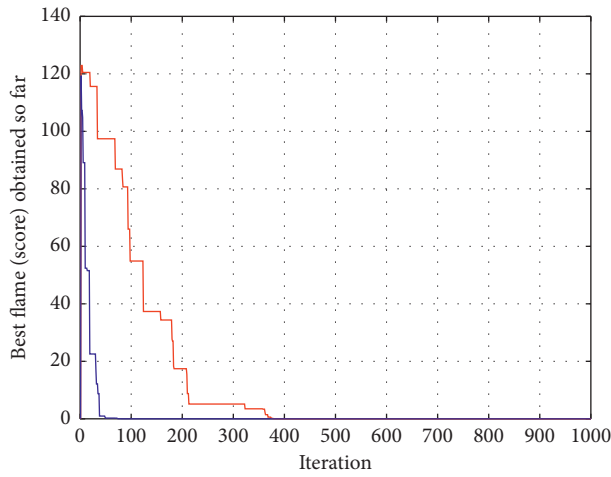
— SCA
— MMSCA

(a)



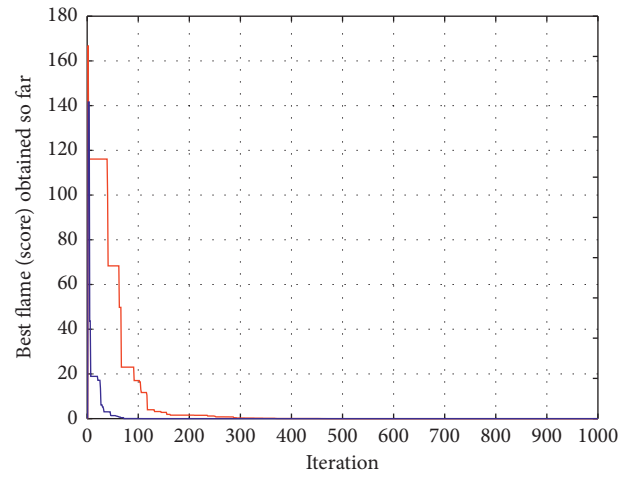
— SCA
— MMSCA

(b)



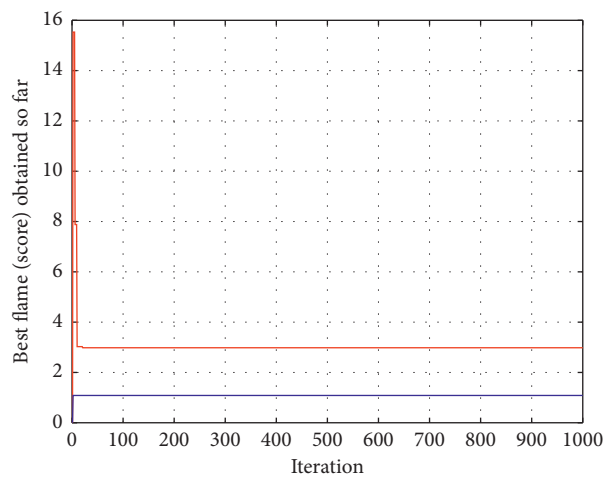
— SCA
— MMSCA

(c)



— SCA
— MMSCA

(d)



— SCA
— MMSCA

(e)

FIGURE 2: Convergence curve of benchmark functions: f_2 , f_3 , f_9 , f_{11} , f_{14} . (a) f_2 . (b) f_3 . (c) f_9 . (d) f_{11} . (e) f_{14} .

TABLE 2: Parameter settings.

Algorithm	Parameters
PSO	$c1 = c2 = 2, \omega = 0.9$
PPSO	The number of groups is 2, $c1 = c2 = 2, \omega = 0.7$
DE	$pCR = 0.2, \beta_{max} = 0.8, \beta_{min} = 0.2$

TABLE 3: Test results compared with other optimization algorithms.

Function	MMSCA	PSO	PPSO	DE
f_1	2.95E-30	6.92E-03	3.03E-28	8.14E-15
f_2	8.02E-20	2.46E-01	5.93E-13	1.58E-08
f_3	6.73E-30	8.03E-02	1.08E-25	6.38E+01
f_4	1.56E-08	7.92E-02	6.67E-08	9.47E-02
f_5	7.02E+00	1.70E+01	1.16E+00	2.94E+01
f_6	2.98E-01	6.48E-03	1.50E-09	6.89E-15
f_7	1.83E-04	1.30E-02	2.07E-03	2.87E-02
f_8	-2.29E+03	-2.40E+03	-2.70E+03	-1.18E+02
f_9	3.30E-06	3.34E+00	6.70E+00	5.81E+01
f_{10}	2.91E-14	2.77E-01	3.68E-13	5.08E-08
f_{11}	0.00E+00	2.29E-01	3.32E-01	9.88E-15
f_{12}	5.60E-02	6.44E-02	1.04E-02	2.25E-16
f_{13}	2.59E-01	7.68E-03	2.28E-03	1.86E-15
f_{14}	9.98E-01	3.86E+00	1.99E+00	3.26E+00
f_{15}	6.85E-04	4.52E-03	1.80E-03	7.23E-04
f_{16}	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
f_{17}	3.99E-01	3.99E-01	3.99E-01	3.98E-01
f_{18}	3.00E+00	3.00E+00	3.00E+00	3.00E+00
f_{19}	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00

TABLE 4: Comparison of MMSCA and other two algorithms.

Data set	Best known results	MMSCA		PSO		GA	
		Best	Dev (%)	Best	Dev (%)	Best	Dev (%)
A-n32-k5	784	808	3.06	829	5.73	818	4.34
A-n33-k5	661	690	4.39	705	6.65	674	1.97
A-n39-k6	831	868	4.45	872	6.08	866	5.35
A-n46-k7	914	964	5.47	977	6.89	957	4.7
A-n60-k9	1408	1426	1.28	1476	4.83	1410	0.14

has completed service; formula (8) represents the vehicle capacity constraint and indicates that the total customer demand on the route served by vehicle v cannot exceed the vehicle loading capacity Q ; formula (9) ensures that each customer can only be served by one vehicle.

5.2. *MMSCA Algorithm for CVRP.* The MMSCA algorithm is a continuous space algorithm, but the CVRP problem is an integer programming problem, so it needs to be modified in the implementation process.

(1) Encoding and decoding

Encoding: using natural number coding, each individual in the population corresponds to a unique arrangement from 1 to n , and each individual corresponds to a distribution scheme.

Decoding: arrange the vehicles according to the sequence of customers in the distribution scheme

and the constraint of maximum loading capacity. That is, it is sequentially included in the path of the first car according to the customer's coding order, and the second car's path is arranged until the maximum load of the first car is exceeded.

(2) Evaluation criteria

The total length of the vehicle path is used to evaluate all individuals and the merits of each distribution scheme.

(3) Implementation process

According to the abovementioned coding method, each population is generated. Each individual (distribution scheme) is evaluated according to the evaluation criteria. Update each individual with formula (3) to generate a new distribution scheme. According to the constraint conditions, the actual vehicle path is generated and evaluated. In the process of updating with formula (3), integer

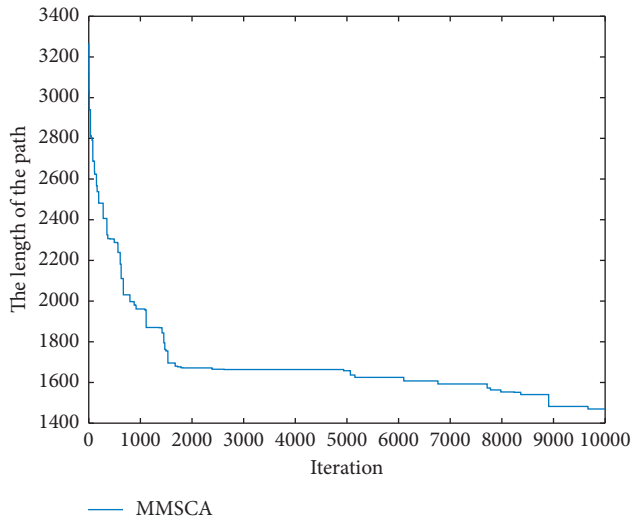


FIGURE 3: Convergence curve (A-n60-k9).

specification constraints are required to avoid generating infeasible solutions.

5.3. Simulation and Results Analysis. In this paper, five standard VRPLIB test samples are selected for testing, and the test results of the MMSCA algorithm are compared with the PSO algorithm [55] and the GA algorithm. Table 4 shows the experimental results. In the experiment, the population number of MMSCA algorithm is divided two groups, the number of individuals in each group is 50, the solution is thirty times, and the number of iterations is 10,000. During the solution process, the best solution result is recorded.

Where the Best column represents the known optimal results, the Dev column indicates the deviation between the optimal solutions obtained by the algorithm and the best known results. As can be seen from the abovementioned table, the MMSCA algorithm has achieved better results and has certain advantages compared with the PSO algorithm and the genetic algorithm. But, there is still a certain gap with the currently known optimal solutions, indicating that the algorithm still has room for improvement. Figure 3 shows the convergence curve when the MMSCA algorithm solves the CVRP instance A-n60-k9.

6. Conclusion

Firstly, in view of the problems existing in the SCA algorithm update process, this paper proposes multigroup multistrategy SCA algorithm (MMSCA). The algorithm contains multiple populations with the same number of individuals, and each population performs a different update strategy. Multiple strategies include rand strategy and best strategy. The rand strategy can improve the global optimization ability of the algorithm, and the algorithm can still be disturbed at the later stage to avoid algorithm stagnation. The best strategy is responsible for keeping the algorithm stable and speeds up the convergence speed of the algorithm. Through numerical experiments on the test function, the

MMSCA algorithm has a certain improvement in the accuracy and convergence speed of the original SCA algorithm. Finally, this paper successfully applies the MMSCA algorithm to solve CVRP (Capacitated Vehicle Routing Problem) and obtains better results, which further confirms the practicability and effectiveness of the algorithm. The next research direction will continue to improve the algorithm and apply it to more practical engineering problems.

Data Availability

All data are included within the tables of this article.

Conflicts of Interest

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

References

- [1] F. Ahmed and K. Kilic, "Fuzzy Analytic Hierarchy Process: a performance analysis of various algorithms," *Fuzzy Sets and System*, vol. 362, pp. 110–128, 2019.
- [2] S.-M. Chen, G. M. T. Manalu, J.-S. Pan, and H.-C. Liu, "Fuzzy forecasting based on two-factors second-order fuzzy-trend logical relationship groups and particle swarm optimization techniques," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 1102–1117, 2013.
- [3] S.-M. Chen, Y.-C. Chang, and J.-S. Pan, "Fuzzy rules interpolation for sparse fuzzy rule-based systems based on interval type-2 Gaussian fuzzy sets and genetic algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 3, pp. 412–425, 2013.
- [4] R. R. Sarukkai, "Link prediction and path analysis using Markov chains," *Computer Networks*, vol. 33, no. 1–6, pp. 377–386, 2000.
- [5] 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.
- [6] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 606–626, 2016.
- [7] J. H. Holland and J. S. Reitman, "Cognitive systems based on adaptive algorithms," *ACM SIGART Bulletin*, vol. 63, pp. 49–49, 1977.
- [8] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95—International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, Australia, 1995.
- [9] C. Sun, J. Zeng, J. Pan, S. Xue, and Y. Jin, "A new fitness estimation strategy for particle swarm optimization," *Information Sciences*, vol. 221, pp. 355–370, 2013.
- [10] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, and M. Ventresca, "Enhancing particle swarm optimization using generalized opposition-based learning," *Information Sciences*, vol. 181, no. 20, pp. 4699–4714, 2011.
- [11] M. Dorigo, V. Maniezzo, and A. Coloni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.
- [12] S. C. Chu, J. F. Roddick, and J. S. Pan, "Ant colony system with communication strategies," *Information Sciences*, vol. 167, no. 1–4, pp. 63–76, 2004.

- [13] S. C. Chu, P. W. Tsai, and J. S. Pan, "Cat swarm optimization," in *Proceedings of the 9th Pacific Rim International Conference on Artificial Intelligence*, pp. 854–858, Guilin, China, August 2006.
- [14] 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*, vol. 6, pp. 3328–3333, Kunming, China, 2008.
- [15] 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.
- [16] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [17] P. W. Tsai, J. S. Pan, B. Y. Liao et al., "Enhanced artificial bee colony optimization," *International Journal of Innovative Computing, Information & Control: IJICIC*, vol. 5, no. 12, pp. 5081–5092, 2009.
- [18] X. M. Hu, J. Zhang, and H. H. Chen, "Optimal vaccine distribution strategy for different age groups of population: a differential evolution algorithm approach," *Mathematical Problems in Engineering*, vol. 2014, Article ID 702973, 7 pages, 2014.
- [19] Z. Meng, J.-S. Pan, and K.-K. Tseng, "PaDE: an enhanced Differential Evolution algorithm with novel control parameter adaptation schemes for numerical optimization," *Knowledge-Based Systems*, vol. 168, pp. 80–99, 2019.
- [20] J.-S. Pan, N. Liu, and S.-C. Chu, "A hybrid differential evolution algorithm and its application in unmanned combat aerial vehicle path planning," *IEEE Access*, vol. 8, pp. 17691–17712, 2020.
- [21] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.
- [22] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, no. 2, pp. 495–513, 2016.
- [23] X. P. Wang, J.-S. Pan, and S.-C. Chu, "A parallel multi-verse optimizer for application in multilevel image segmentation," *IEEE Access*, vol. 8, pp. 32018–32030, 2020.
- [24] E. Nabil, "A modified flower pollination algorithm for global optimization," *Expert Systems with Applications*, vol. 57, pp. 192–203, 2016.
- [25] T.-T. Nguyen, J.-S. Pan, and T.-K. Dao, "An improved flower pollination algorithm for optimizing layouts of nodes in wireless sensor network," *IEEE Access*, vol. 7, pp. 75985–75998, 2019.
- [26] E. Emary, W. Yamany, A. E. Hassaniien, and V. Snasel, "Multi-objective gray-wolf optimization for attribute reduction," *Procedia Computer Science*, vol. 65, pp. 623–632, 2015.
- [27] P. Hu, J.-S. Pan, S.-C. Chu, Q.-W. Chai, T. Liu, and Z.-C. Li, "New hybrid algorithms for prediction of daily load of power network," *Applied Sciences*, vol. 9, no. 21, p. 4514, 2019.
- [28] J.-S. Pan, P. Hu, and S.-C. Chu, "Novel parallel heterogeneous meta-heuristic and its communication strategies for the prediction of wind power," *Processes*, vol. 7, no. 11, p. 845, 2019.
- [29] H. Duan and P. Qiao, "Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning," *International Journal of Intelligent Computing and Cybernetics*, vol. 7, no. 1, pp. 24–37, 2014.
- [30] A. Q. Tian, S. C. Chu, J. S. Pan et al., "A compact pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station," *Sustainability*, vol. 12, no. 3, 2020.
- [31] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, vol. 284, pp. 65–74, 2010.
- [32] P. W. Tsai, J. S. Pan, B. Y. Liao et al., "Bat algorithm inspired algorithm for solving numerical optimization problems," *Applied Mechanics and Materials*, vol. 148, pp. 134–137, 2012.
- [33] A. E. Ezugwu and D. Prayogo, "Symbiotic organisms search algorithm: theory, recent advances and applications," *Expert Systems with Applications*, vol. 119, pp. 184–209, 2019.
- [34] S.-C. Chu, Z.-G. Du, and J.-S. Pan, "Symbiotic organism search algorithm with multi-group quantum-behavior communication scheme applied in wireless sensor networks," *Applied Sciences*, vol. 10, no. 3, p. 930, 2020.
- [35] Z. Meng, J.-S. Pan, and H. Xu, "QUasi-Affine TRansformation Evolution (QUATRE) Algorithm: a cooperative swarm based algorithm for global optimization," *Knowledge-Based Systems*, vol. 109, pp. 104–121, 2016.
- [36] Z. Y. Meng and J. S. Pan, "QUasi-Affine TRansformation Evolution (QUATRE) Algorithm: a parameter-reduced differential evolution algorithm for optimization problems," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 4082–4089, Vancouver, Canada, July 2016.
- [37] J. S. Pan, Z. Y. Meng, H. R. Xu et al., "QUasi-Affine TRansformation Evolution (QUATRE) Algorithm: a new simple and accurate structure for global optimization," in *Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp. 657–667, Morioka, Japan, August 2016.
- [38] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
- [39] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 809–818, 2016.
- [40] 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, no. 4, pp. 809–818, 2005.
- [41] S.-C. Chu, J. F. Roddick, J.-S. Pan, and C.-J. Su, "Parallel ant colony systems," in *Lecture Notes in Computer Science*, vol. 2871, pp. 279–284, Springer, Berlin, Germany, 2003.
- [42] 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.
- [43] C.-I. Wu, H.-Y. Kung, C.-H. Chen, and L.-C. Kuo, "An intelligent slope disaster prediction and monitoring system based on WSN and ANP," *Expert Systems with Applications*, vol. 41, no. 10, pp. 4554–4562, 2014.
- [44] H.-W. Wang, C.-H. Chen, D.-Y. Cheng, C.-H. Lin, and C.-C. Lo, "A real-time pothole detection approach for intelligent transportation system," *Mathematical Problems in Engineering*, vol. 2015, Article ID 869627, 7 pages, 2015.
- [45] J. S. Pan, L. P. Kong, T. W. Sung et al., " α -fraction first strategy for hierarchical model in wireless sensor networks," *Journal of Internet Technology*, vol. 19, no. 6, pp. 1717–1726, 2018.

- [46] J. S. Pan, L. P. Kong, T. W. Sung et al., "A clustering scheme for wireless sensor networks based on genetic algorithm and dominating set," *Journal of Internet Technology*, vol. 19, no. 4, pp. 1111–1118, 2018.
- [47] J. Wang, Y. Cao, B. Li, H.-j. Kim, and S. Lee, "Particle swarm optimization based clustering algorithm with mobile sink for WSNs," *Future Generation Computer Systems*, vol. 76, pp. 452–457, 2017.
- [48] J. Wang, Y. Gao, X. Yin, F. Li, and H.-J. Kim, "An Enhanced PEGASIS Algorithm with Mobile Sink Support for Wireless Sensor Networks," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 9472075, 9 pages, 2018.
- [49] W. Long, T. Wu, X. Liang, and S. Xu, "Solving high-dimensional global optimization problems using an improved sine cosine algorithm," *Expert Systems with Applications*, vol. 123, pp. 108–126, 2019.
- [50] C. W. Qu, Z. L. Zeng, J. Dai, and W. He, "A modified sine-cosine algorithm based on neighborhood search and Greedy Levy mutation," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 4231267, 19 pages, 2018.
- [51] S. N. Chegini, A. Bagheri, and F. Najafi, "PSOSCALF: a new hybrid PSO based on Sine Cosine Algorithm and Levy flight for solving optimization problems," *Applied Soft Computing*, vol. 73, pp. 697–726, 2018.
- [52] 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.
- [53] G. B. Dantzing and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.
- [54] J.-Y. Potvin, "State-of-the art review-evolutionary algorithms for vehicle routing," *InCforms Journal on Computing*, vol. 21, no. 4, pp. 518–548, 2009.
- [55] Q. Shan and J. C. Wang, "Solve Capacitated Vehicle Routing Problem Using Hybrid Chaotic Particle Swarm Optimization," in *Proceedings of the 2013 Sixth International Symposium on Computational Intelligence and Design Hangzhou*, pp. 422–427, Hangzhou, China, 2013.