Parallel Sine Cosine Algorithm for the Dynamic Deployment in Wireless Sensor Networks

Fang Fan[1,2], Shu-Chuan Chu[1], Jeng-Shyang Pan[1], Qingyong Yang[1], Huiqi Zhao[2]
[1] College of Computer Science and Engineering, Shandong University of Science and Technology, China
[2] Department of Information Engineering, Shandong University of Science and Technology, China
fangfansdust@hotmail.com, scchu0803@gmail.com, jengshyangpan@gmail.com, qyzide@163.com, zhqskd@163.com

Abstract

All along, people have a high enthusiasm for the research of optimization algorithm. A large number of new algorithms and methods have emerged. The sine cosine algorithm (SCA) is an excellent algorithm that has appeared in recent years. It is a stochastic optimization algorithm based on population. Compared with the existing algorithms, SCA is a suitable solution to different optimization problems, especially the optimization of unimodal functions. It is qualified to optimize real-world problems with unknown and limited search space. But sometimes it does not perform satisfactorily when dealing with some specific problems, such as optimization of multimodal functions or composite functions. This paper presents a parallel version of the sine cosine algorithm (PSCA) with three communication strategies. Different strategies can be selected according to the type of optimization function to achieve better results. We have repeatedly tested different types of functions, and the results show that the proposed PSCA can solve the optimization problem more specifically. In the simulation of wireless sensor network (WSN) dynamic deployment optimization, it is found that using this method can get the ideal sensor node distribution, which makes PSCA's performance in solving other practical problems worth looking forward to.

Keywords: Sine cosine algorithm (SCA), Parallel sine cosine algorithm (PSCA), Communication strategies, Wireless sensor networks (WSN), Dynamic deployment

1 Introduction

Optimization is a technique that studies how to determine the optimal value of unknown parameters of the target system under certain constraints. With optimization, you can find the best solution from a set of available solutions by reaching the extreme value of the system's objective function. Obviously, optimization problems are widespread in various fields. Due to the needs of practical applications and advances in computing technology, research on optimization methods has developed rapidly. More and more optimization algorithms are extensively used in function optimization, scientific research, engineering application, etc. Moreover, researchers have been actively pursuing better optimization results.

The traditional optimization method is "method-oriented", that is, it can only solve the problem of meeting the applicable conditions of the method. So many times, we have to simplify or change the original problem in order to use a certain method. This makes the traditional optimization methods have many limitations such as low calculation efficiency, easy to fall into local optimum, and restricted application range. Due to the variety of optimization problems and the increasing requirements for the performance, traditional optimization methods have failed often. Aiming at the shortcomings of traditional optimization methods, people put forward some new requirements for optimization. It cannot be limited to the solution of a certain type of problem, and should be changed from "method-oriented" to "problem-oriented". Since the 1960s and 1970s, people have introduced artificial intelligence technology and biological evolution mechanisms into optimization methods, and progressively formed a group of refreshing modern optimization methods that are completely different from traditional optimization methods, for example genetic algorithms (GA) [1], particle swarm optimization (PSO) [2], differential evolution (DE) [3], bat algorithm (BA) [4], ant colony optimization (ACO) [5], etc.

Different types of optimization problems can use different optimization methods, even the same type of problems can adopt multiple optimization methods. In contrast, some optimization methods can be used to solve multiple types of problem models, and they may outperform other algorithms for specific problems. This can be explained by No Free Lunch (NFL) theorem [6]. This has greatly encouraged the majority of researchers to conduct extensive and in-depth research on optimization algorithms. The theoretical

*Corresponding Author: Jeng-Shyang Pan; E-mail: jengshyangpan@gmail.com
DOI: 10.3966/160792642021052203001

research surrounding optimization involves three main aspects: improving existing technologies, combining different algorithms, and proposing original algorithms.

In 2016, the Australian scholar Mirjalili proposed a new algorithm, the sine cosine algorithm (SCA) [7], which is not only novel in thought but also refined in structure. As a stochastic algorithm based on population, it has very loose requirements for objective function. Like other stochastic algorithms, optimization problem is regarded as a black box [8]. In addition, the algorithm has simple structure, few controlled parameters and is easy to realize. The global optimization problems can be solved only by using the iterations of sine cosine function. The superior optimization performance of SCA and its high adaptability to various complex optimization problems have attracted researchers' attention. SCA has been successfully applied in many application areas.

Along with the sustained development of optimization theory and the continuous progress of computing technology, parallel processing is generally considered as an effective method for function optimization, which can not only increase the efficiency of optimization, but also improve the effect [9]. The parallel processing of the optimization algorithm can not only allocate the computation to multiple processors, but also expand the global search capability and improve the precision compared with the original algorithm. The parallel processing has been applied for many existing algorithms, for example parallel GA [10], parallel PSO [11], parallel CSO [12], parallel BA [13], etc.

This article introduces the concept of parallel processing into SCA and designs three communication strategies for this purpose. Three kinds of communication strategies are designed according to different types of functions, which can solve the optimization problems of unimodal, multimodal, composite functions and even unknown types. The benchmark function test confirms that PSCA has better optimization performance than SCA both in accuracy and convergence. In order to further demonstrate the practical value of PSCA, application simulation was carried out at the end of the work. We try to optimize the dynamic deployment problem in wireless sensor networks (WSN). Comparing the simulation results, the distribution of sensor nodes obtained by PSCA is the most uniform. This further proves the effectiveness and potential of the PSCA in practical applications.

Briefly introduce the organizational structure of the rest of the article:

Section 2 reviews SCA and related research work around it. In section 3, the design of the PSCA with three communication strategies is introduced in detail. Then We show relevant experimental results and make an analysis comparison between SCA and PSCA in section 4. Section 5 is about the practical application of PSCA. We use it to solve the dynamic deployment problem of WSN. The final section 6 is summary and outlook.

2 Related Woks

2.1 Sine Cosine Algorithm

The design of SCA is very ingenious. Finding the best solution just uses the oscillation characteristics of sine and cosine functions. First, the algorithm is initialized to generate a set of random solutions. Then, through the cooperation of the two phases of exploration and exploitation, the continuous optimization of the search space is achieved. The optimization trend is dominated by the sine cosine function. After several iterations, the optimization process ends and the optimal solution is output or approximated. The following is the specific optimization process of SCA. At the beginning, N search individuals $X_1, X_2, X_3, \ldots, X_N$ are randomly generated in d-dimensional search space. $X_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{id})$ is the position of the ith individual, where $i = 1, 2, ..., N$. Next, substitute each individual X_i in the population into the evaluation function and calculate its fitness value $f(X_i)$. And record the current optimal individual $P_i = (P_{i1}, P_{i2}, \ldots, P_{id})$, whose fitness value is the best. The update equations of search individual position in each iteration are as follows:

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

In Eqs. (1), t represents the current number of iterations. X_i^t is the value of the solution at the current number of iterations in the ith dimension. It will become X_i^{t+1} in the next iteration. P_i^t is the value of the ith dimension of the current destination point. And $|\cdot|$ is an absolute value symbol. The Eqs.(1) shows that there are four primary parameters, r_1, r_2, r_3, and r_4, all of which are random numbers. The parameter r_1 is a linear decreasing function which decreases gradually from a to 0 with the increase of iterations. Its calculation method is as follows:

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

In Eq. (2), a is a constant and $a > 0$. T is the upper limit of the number of iterations of the algorithm. The parameter r_1 mainly acts as a guide for the ith individual in the next iteration. Specifically, when $r_1 < 1$, the next search space is between the current solution and the dimension, i.e. local exploitation; when $r_1 > 1$, the search space is outside it, i.e. global exploration. The parameter $r_2 \in [0, 2\pi]$, which is subject to uniform distribution, is used to control the movement distance in the next iteration. The parameter $r_3 \in [0, 2]$, as a random weight subject to uniform distribution, can control the influence of the current optimal solution P_i^t on the update of individual position. The parameter $r_4 \in [0, 1]$ is responsible for switching the position update strategies: when $r_4 < 0.5$, chooses the sine component in Eqs. (1); otherwise, chooses the cosine component.

SCA is efficient, fully distributed and with low complexity. It can switch between exploration and exploitation, which is conducive to finding more potential solutions and has strong global optimization capabilities. In addition, SCA also realized a gradual transition from early emphasis to exploration to late focus on exploitation, which accelerated the convergence of the function. Figure 1 shows the principle of SCA optimization.

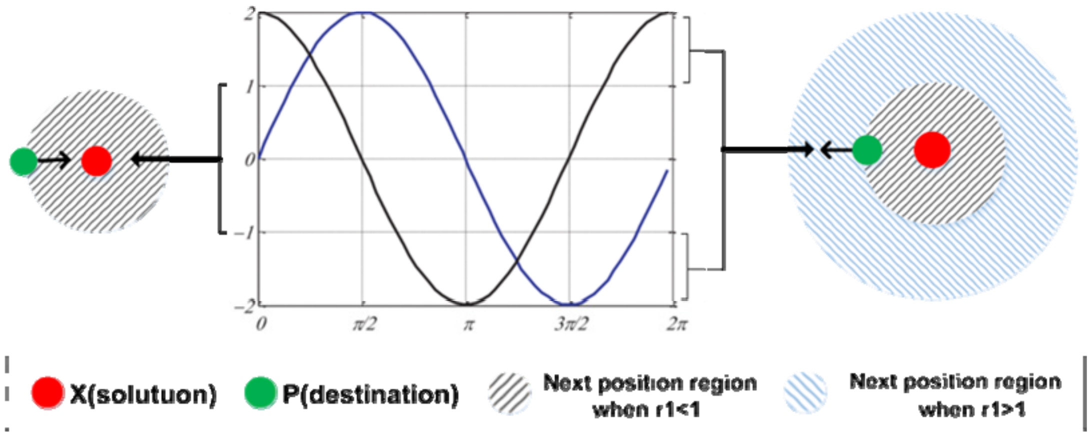


Figure 1. Principle of SCA algorithm

2.2 The Improvement of Sine Cosine Algorithm

It turns out that SCA has better optimization performance than many other algorithms (e.g. PSO, GA, BA, firefly algorithm (FA) [14], flower pollination algorithm (FPA) [15], etc.). It has faster convergence speed and higher convergence accuracy. However, through the analysis of the SCA population update model, it is found that during the entire optimization iteration process, the dependence of the search individual's position update on the position of the search individual itself is always the same, which causes the global search capability to be weakened in the early stages. In addition, since the solution always updates its position near the current optimal solution, the search space is gradually shifting towards the optimal region during the optimization process. However, the negative effect of this operation is that the individual diversity of solution space is greatly reduced. And this will increase the risk of premature convergence or only local optimization of the algorithm. What's more, SCA optimization is slower and takes longer than most other stochastic algorithms when the search space is relatively larger. Some scholars put forward many improved algorithms based on the existing problems of SCA, and the research on SCA improvement strategies is mainly divided into two categories. One is to integrate with other algorithms to develop the optimization performance. For instance, by combining grey wolf optimization (GWO) [16] with SCA, a novel hybrid GWO-SCA method [17] is proposed, which significantly improved the accuracy of the algorithm. The hybrid SCA-DE algorithm [18] which integrates SCA with differential evolution can reduce the local optimization to a certain extent and make the convergence speed faster. The other is to use the search strategy of other optimization algorithms to improve SCA's search ability. For example, a new sine and cosine algorithm improves the traditional SCA by increasing two coefficients of exploration rate and exploitation rate, which greatly improves the convergence speed [19]. The opposition-based sine cosine global optimization algorithm (OBSCA) [20] can obtain higher accuracy of the optimization process. In order to improve the ability of SCA for global exploration and local exploitation, some researchers have also tried to re-establish the change law of existing parameters or introduce other new methods, and achieved certain results [21].

Although the current improvement of SCA has achieved some results, it still has various defects. And the hybridization and fusion of multiple algorithms will make the algorithm more and more complicated. To this end, we propose a parallel processing of SCA (PSCA). While improving the overall performance of SCA, it also combines the advantages of parallel processing. The next section goes into detail.

3 Parallel Sine Cosine Algorithm (PSCA)

In this part, the main idea and implementation scheme of PSCA are discussed in detail. For the purpose of effectively making up for the deficiencies of the original SCA, we introduced the concept of multi-population. This is conducive to maintaining the diversity of population, so as to ensure that the best solution can be found in the process of optimization as much as possible. Because the search process is from one-point set (population) to another point set (population) in space, it is actually a kind of parallel search. In this way, it can not only help to jump out of local optimization, but also realize large-scale parallel computing. The specific scheme is as follows: first construct a parallel processing structure by grouping the entire population to get several sub-populations. Then each sub-population evolves independently according to the iterative rules. The evaluation of the solution is based on the fitness value. After triggering a certain inter-population communication scheme, replace inappropriate individuals in the population with corresponding strategies, and exploit or explore the promising area. Based on this idea, this paper designs three kinds of PSCA communication strategies adapted to different function types. Strategy 1 is mainly applicable to simple unimodal functions. Strategy 2 is suitable for multimodal functions, which usually present multi-peak and multi-valley. Strategy 3 is available for complex functions, or when the type of the function is unknown. The following is a detailed description of the three communication strategies:

Strategy 1 is a communication method based on global optimal replacement. First, the search population X obtained by SCA initialization is divided into N groups g_i, $g_i \in X (i = 1, 2, 3, \ldots, N)$. Each group g_i is independently optimized by SCA. Suppose the total number of iterations is M, including K exchanges $(K = k, 2k, 3k, \ldots, M)$, k is the predefined number of iterations. When the communication condition is triggered (assuming iteration t), migration occurs between the groups. The worst n individuals in each group are replaced by the global optimal solution P^t with the highest fitness the population X. M, N, k, t are predefined constants. Figure 2 shows the first communication strategy in the form of a flowchart.

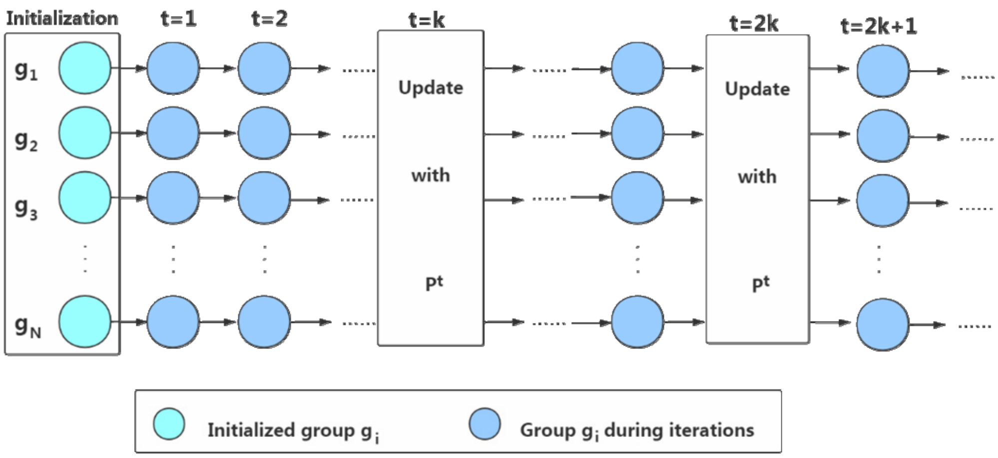


Figure 2. Communication strategy 1 for unimodal function optimization

When the objective function is complex, such as multimodal function, the optimization algorithm is prone to situations that cannot rely on previous iterative rules to get rid of local optimum local optimum. Generally speaking, the local optimum is caused by the lack of communication between populations. So we propose the second strategy, as shown in Figure 3. Strategy 2 adds a method of genome replacement based on strategy 1. In each group, the individual with the best fitness value is recorded as $g_{i,b}$, and the individual with the sub-best fitness is recorded as $g_{i.sb}$, $i = 1, 2, 3, ..., N$. The best fitness individual and the sub-best fitness individual of each group form a dominant genome $Gen(g_{1.b}, g_{1.sb}, g_{2.b}, g_{1.sb}, ..., g_{N.b}, g_{N.sb})$. When the communication condition is triggered, one individual candidate ($candidate \in Gen$) in the genome is randomly selected to replace the individual with the worst fitness in each group. This method can effectively protect the diversity of the population and maintain evolutionary vitality. Therefore, strategy 2 contains two alternatives: global optimal replacement and genome replacement. While communicating between groups, choose one randomly.

When we know the type of the objective function, a targeted selection of strategy 1 or strategy 2 can achieve better optimization effect. However, if the objective function is complex or unknown, improper application may lead to poor performance. In this case, the third strategy which is inspired by Tabu search [22-24], can be considered, as shown in Figure 4. Tabu search is a heuristic algorithm which can avoid circuit search as much as possible. It is a deterministic local optimal jumping strategy and has great potential in global optimization of functions. The most important idea is to mark some objects corresponding to the searched local optimal solution, and try to avoid these objects in further iterations, so as to ensure to explore

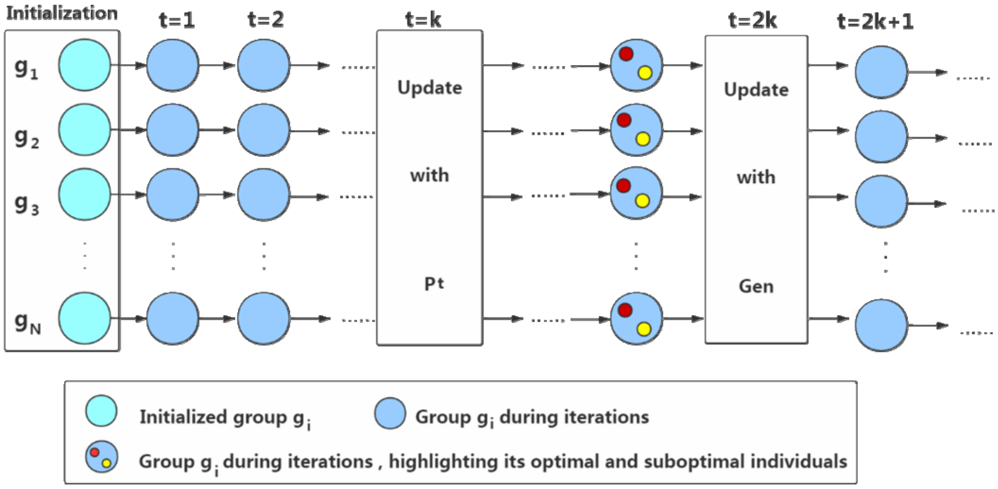


Figure 3. Communication strategy 2 for multimodal function optimization

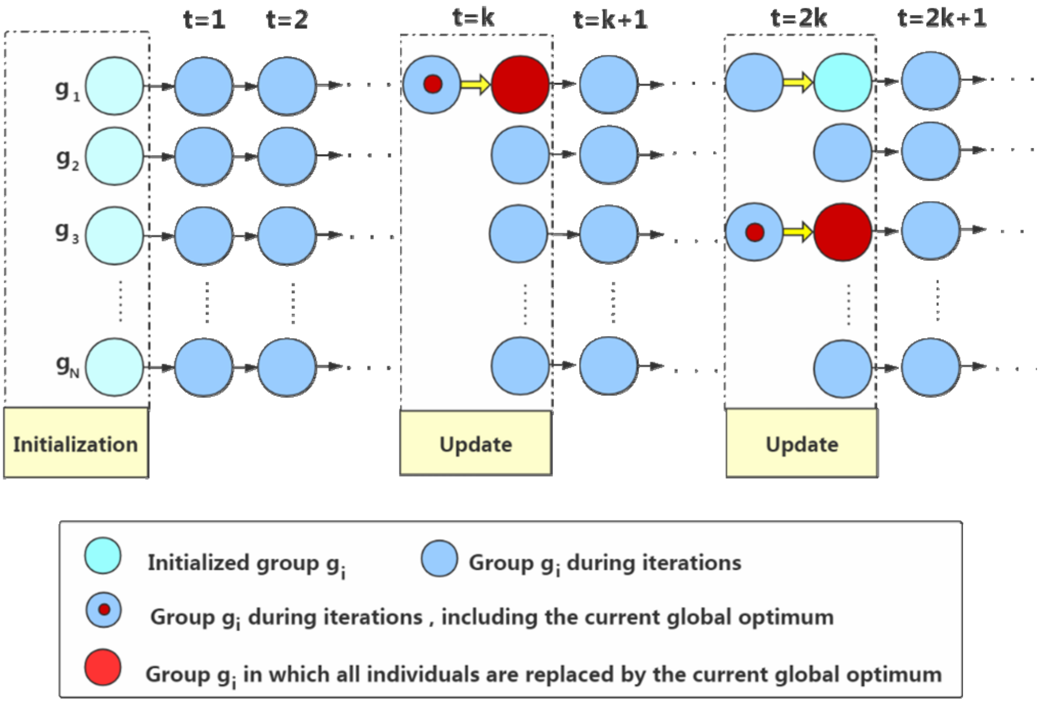


Figure 4. Communication strategy 3 for composite or unknown function optimization

with different effective search paths. Inspired by this, strategy 3 is proposed. The initialization and grouping of the population are the same as strategy 1 and strategy 2. The difference is that when running to the tth iteration for communication, the current global optimal solution P^t is used to replace all the individuals of its group which is marked as g_{best} while the other groups remain unchanged. When the t'th iteration triggers the next communication, it continues to replace all individuals in the group it belongs to with the current global optimal $P^{t'}$. If $P^{t'}$ appears outside the g_{best} group marked last time, the previous g_{best} group will be reinitialized and the g_{best} group's mark will be updated. Repeat until all iterations are completed.

Here are the complete steps for the proposed PSCA:

1. Initialization:

Generate S search individuals randomly and divide them into N groups. The jth group has N_j individuals, so $S=\sum_{j=0}^{N-1} N_j$. Each individual is denoted as x_{ij}^t, which represents the value of the ith individual in the jth group when the tth iteration is performed, where $i=0,1,2,3,\dots,N_j-1$.

2. Evaluation:

Evaluate each individual's fitness value $f(x_{ij}^t)$ in all the groups.

3. Update:

Use Eqs. (1) to update the position of each individual.

4. Communication:

Choose one of the three communication strategies:

Strategy 1: Migrate the optimal individual P^t in the population X to each group, replace the worst individual in each group with P^t, and update P_j^t of each group in every iteration cycle.

Strategy 2: Randomly choose global optimal replacement or genome replacement. The global optimal replacement method is the same as strategy 1. Genome replacement is as follows: The optimal individual $g_{i.b}^t$ and the sub-optimal individual $g_{i.sb}^t$ in each group constitute the genome *Gen*. An individual candidate is randomly selected and migrated to each group, replacing the worst searching individual in each group and updating the P_j^t in every iteration cycle.

Strategy 3: Replace the other members of the group with the best individual P^t in the whole population *X*, which is marked as g_{best} group; the other groups are unchanged, and only P_j^t of each group is updated. In the next communication, repeat the above operations, and check whether the group to which the optimal solution belongs is the last marked g_{best} group. If not, reinitialize the last marked group and re-mark the g_{best} group.

5. Termination:

Repeat steps 2 to 4. If a predefined function value has been obtained or all iterations have been completed, record the global optimal solution P^t and its best fitness value (P^t), and the optimization process ends here.

4 Experiment Results and Analysis

Whether the PSCA including the three communication strategies meets the theoretical expectations requires relevant experiments to verify. This section introduces the experimental scheme in detail and conducts a comprehensive analysis of the experimental results. In order to make an objective comparison with SCA, all the benchmark functions selected are from the literature that proposed the original SCA. The experimental results show that strategy 1 is the best for unimodal function optimization, while strategy 2 is excellent when solving multimodal function. Finally, the comprehensive processing capacity of strategy 3 is tested when facing unimodal, multimodal and composite benchmark functions.

In the test of strategy 1, the performance of SCA and PSCA in the optimization of unimodal functions $(F_1(x)-F_4(x))$ was compared. In order to achieve fair competition, the populations of SCA and PSCA are set to the same size. There are 500 iterations in total and the structures of the solutions are all 10-dimensional real-valued vectors. To verify the effectiveness of strategy 1, the number of exchanges was set to 20 and the replacement ratio was set to 25% in the first experiment. The experimental results show that strategy 1 performs well in solving optimization problems of unimodal functions, and its convergence speed and optimization accuracy are significantly improved compared with SCA. Please see Table 1 for detailed experimental results. The optimal solutions have been highlighted in the table. And Figure 5 shows the optimization result of the benchmark function F_1.

Table 1. Performance comparison of SCA and PSCA's strategy 1 for the unimodal functions $(F_1(x)-F_4(x))$

Functions	Functions values	
	SCA	PSCA (strategy 1)
$F_1(x)$	1.50E-13	1.76E-15
$F_2(x)$	2.44E-08	3.89E-11
$F_3(x)$	1.74E-05	1.24E-05
$F_4(x)$	1.48E-03	2.30E-06

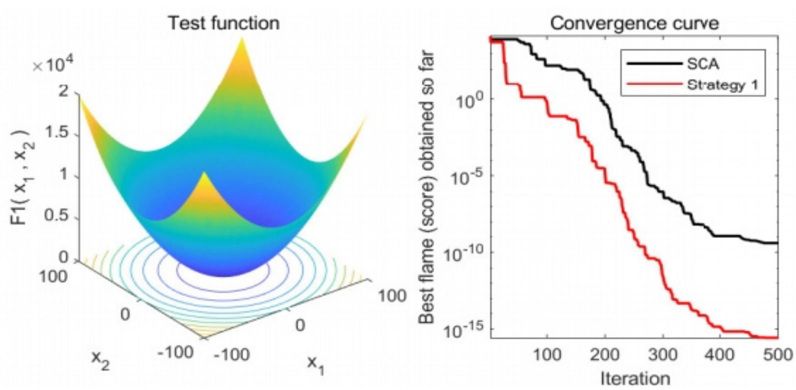


Figure 5. The experimental result of benchmark function F_1

Next experiment compares the performance of SCA and PSCA based on strategy 1 and strategy 2 in multimodal function optimization. From the experimental results, strategy 2 has the best effect, and the average convergence accuracy is 35% higher than SCA. In addition, the performance of strategy 1 became unstable. Please see Table 2 for detailed experimental results. The optimal solutions have been highlighted in the table. And Figure 6 shows the optimization result of the benchmark function F_9.

Table 2. Performance comparison of SCA and PSCA (including strategy 1 and strategy 2) for the multimodal functions $(F_9(x) - F_{12}(x))$

Functions	Functions values		
	SCA	PSCA (strategy 1)	PSCA (strategy 2)
$F_8(x)$	5.77E-01	2.67E-11	0.00E+00
$F_9(x)$	1.85E-07	2.18E-08	1.50E-09
$F_{10}(x)$	3.08E-12	1.44E-15	0.00E+00
$F_{11}(x)$	1.07E-01	6.23E-02	3.65E-02
$F_{12}(x)$	4.70E-01	2.24E-01	1.96E-01

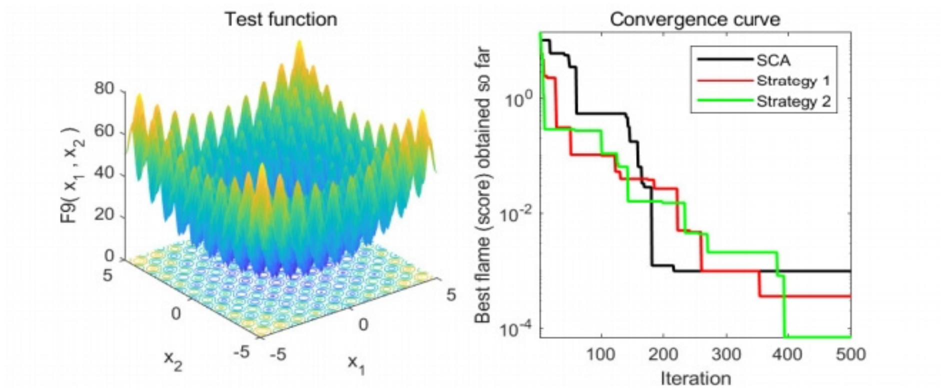


Figure 6. The experimental result of benchmark function F_9

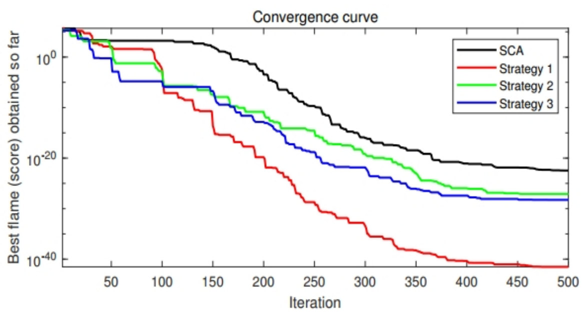
The third experiment tests the optimization performance of SCA and PSCA when the objective function contains multiple types. There are unimodal, multimodal, and composite functions. Through repeated experiments, the results show that the overall performance of PSCA's strategy 3 is better than others. Compared with SCA, strategy 3 generally has higher convergence accuracy and speed, increased by 25% and 19% respectively. Most of the time, strategy 1 and strategy 2 performed better than SCA, but still lagged behind Strategy 3. So strategy 3 is recommended when faced with complex optimization problems or unclear problems. Table 3 shows the experimental results. The optimal solutions are highlighted. And Figure 7 displays the optimization results of five benchmark functions: $F_4, F_{11}, F_{12}, F_{13}, F_{18}$.

Table 3. Performance comparison of SCA and PSCA (including strategy 1, strategy 2 and strategy 3) for all the benchmark functions

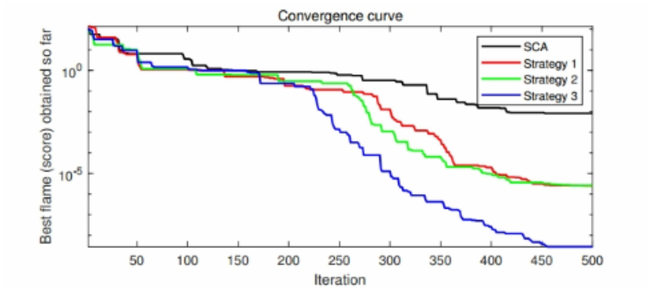
Functions	Functions values			
	SCA	PSCA (strategy 1)	PSCA (strategy 2)	PSCA (strategy 3)
$F_1(x)$	3.81E-17	1.57E-12	4.92E-10	2.43E-18
$F_2(x)$	5.75E-12	1.77E-12	1.87E-10	7.76E-10
$F_3(x)$	7.48E-15	2.98E-15	1.88E-13	2.99E-17
$F_4(x)$	2.20E-25	4.59E-22	8.59E-25	1.94E-32
$F_5(x)$	2.45E-12	4.67E-12	7.15E-10	2.04E-10
$F_6(x)$	4.45E-04	6.67E-05	8.49E-07	3.84E-04
$F_7(x)$	0.00E+00	0.00E+00	0.00E+00	0.00E+00
$F_8(x)$	3.18E-27	1.77E-31	6.27E-26	5.47E-25

Table 3. Performance comparison of SCA and PSCA (including strategy 1, strategy 2 and strategy 3) for all the benchmark functions (continue)

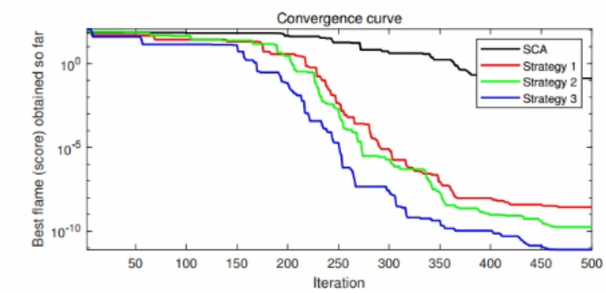
Functions	Functions values			
	SCA	PSCA (strategy 1)	PSCA (strategy 2)	PSCA (strategy 3)
$F_9(x)$	6.64E-03	2.85E-03	4.31E-04	4.66E-04
$F_{10}(x)$	7.26E+00	7.33E+00	7.22E+00	7.65E+00
$F_{11}(x)$	3.77E-13	3.51E-10	2.03E-04	1.75E-10
$F_{12}(x)$	8.54E-01	2.47E+01	4.54E+00	2.61E-06
$F_{13}(x)$	1.46E-08	4.47E-12	1.24E-01	2.65E-14
$F_{14}(x)$	2.11E+03	2.11E+03	2.36E+03	2.09E+03
$F_{15}(x)$	2.35E-07	4.22E-08	1.28E-06	3.62E-08
$F_{16}(x)$	7.82E-02	8.04E-02	4.83E-02	7.26E-02
$F_{17}(x)$	3.80E-01	1.83E-01	4.02E-01	2.23E-01
$F_{18}(x)$	1.90E-08	2.78E-08	1.02E-07	4.34E-11
$F_{19}(x)$	2.74E+00	2.36E+00	4.10E+00	1.45E+00
$F_{20}(x)$	-5.00E+00	-5.00E+00	-5.00E+00	-5.00E+00
$F_{21}(x)$	9.78E-03	9.72E-03	9.72E-03	9.72E-03
Win	4	6	7	12



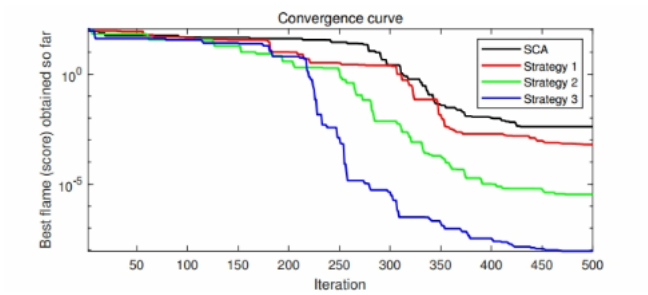
(e) F_4



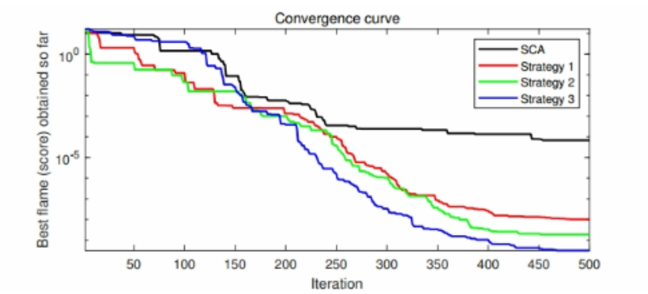
(a) F_{11}



(b) F_{12}



(c) F_{13}



(d) F_{18}

Figure 7. Convergence curves for the benchmark functions: $F_4, F_{11}, F_{12}, F_{13}, F_{18}$

5 Application of PSCA in Dynamic Deployment of Wireless Sensor Networks

The value of the proposed PSCA also needs to be reflected in the solution of actual optimization problems. To this end, we selected the dynamic deployment problem of WSN for application simulation. WSN is composed of a large number of sensor nodes deployed in the monitoring area. The nodes exchange messages through wireless communication, thus forming a self-organizing network. This technology can not only reduce the mutual interference between different nodes, but also cuts the cost [25-27], and the node position can be set flexibly [28]. WSN can realize real-time sensing and monitoring of various objects of interest in the target area [29-33]. Today in the age of the Internet of Things, WSN has become a key technology which has a broad application prospect and has been successfully applied in military, disaster rescue, environment, medical, industrial, commercial and other fields [34-36]. Node deployment is a key issue related to the overall performance of WSN. It will have an important impact on coverage, energy consumption, reliability, security, quality of service (QoS) and other aspects [37-42]. The deployment problem of WSN can be simply understood as how to maximize coverage rate with as few sensors as possible, while maintaining excellent network connectivity and consuming the least energy [43-44]. Depending on whether the node location changes in the application scenario, two schemes for deploying nodes have been generated: static deployment and dynamic deployment. The so-called static deployment means that it has been formulated in advance before the network starts. The topology of the sensor network is predefined, and the specific location of each node has been designed. After setting the node position, it is considered that the node will not move and the network structure will not change dynamically. This method is mainly applicable to the situation of friendly environment and stable structure in the target area, such as indoor monitoring, industrial control, etc. Obviously, if the network topology needs to be dynamically adjusted, or the environment is harsh (e.g. large-scale unmanned areas, nuclear radiation areas, battlefields, etc.), static deployment is no longer suitable and dynamic deployment is required in these cases [45]. In dynamic deployment, sensor node locations are randomly set during initialization. Obviously, this node deployment method is very prone to uneven distribution in the target area, which will lead to poor network connections, unbalanced workload and low coverage quality. So subsequent efforts need to be made to adjust the position of the nodes to cover as much of the target area as possible. How to arrange sensor positions reasonably to achieve high coverage rate has always been the focus of researchers. An improved algorithm ACO-Greedy [46], which is produced by introducing the greedy migration mechanism into the ant colony optimization, can quickly achieve high network coverage and reduce power consumption, but it will increase the cost because more sensor nodes are needed for the area with large traffic. Although the node deployment strategy using the modified ABC algorithm [47] can sometimes achieve higher coverage and faster deployment speed, it still faces the high risk of falling into a local optimum. The glowworm swarm optimization (GSO) [48] also be applied to sensor deployment [49], which can expand the network coverage scale without increasing the number of nodes, but the impact of errors on the optimal solution will accumulate as the number of iterations increases.

5.1 Coverage Model

The node sensing model we chose is the disk sensing model, and its other name is Boolean sensing model. In a two-dimensional coordinate system, draw a circle with the node position as the center and *r* as the radius. The resulting closed circular area is the coverage of a sensor node. *r* is called the node's sensing radius. Any point located inside the circle can be fully monitored, while points outside the circle are considered to be undetectable. So there are only two possibilities in this model, the target is either absolutely detected or not. The outstanding advantage of this model is that it can simplify the solution of the coverage strategy, allowing us to explore the optimization problem in more depth.

Suppose the target area is a two-dimensional square with $A = w \times w$. In order to simplify the calculation, this area is discretized into $m \times n$ pixel points $L_j (j = 1, 2, ..., m \times n)$. The pixel position $L_j(x_j, .y_j)$ is the optimized destination for node deployment. N sensor nodes S_i are randomly scattered, where $i = 1, 2, ..., N$, and $S_i(x_i, y_i)$ represents the position coordinate of the *i*th node. $d(S_i, L_j)$ is the Euclidean distance between the sensor node S_i and the pixel L_j. In order to determine whether the pixel L_j is covered by the sensor node S_i, the Euclidean distance $d(S_i, L_j)$ between the two needs to be obtained first, and then the distance is compared with the sensing radius *r*. The calculation method of $d(S_i, L_j)$ is shown in Eq.(3).

$$d(S_i, L_j) = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad \textbf{(3)}$$

$P(S_i, L_j)$ is the probability that the pixel L_j is perceived by the node S_i, and can be calculated using Eq.(4).

$$P(S_i, L_j) = \begin{cases} 1, & d(S_i, L_j) \geq r \\ 0, & otherwise \end{cases} \quad \textbf{(4)}$$

There may be overlap between the sensing areas of the sensor nodes. That is, there may be a case where multiple nodes S_i simultaneously cover the position of the same pixel L_j in the monitoring area. So $P(S, L_j)$ is defined as the joint sensing probability of the pixel L_j, as shown in Eq. (5).

$$P(S, L_j) = 1 - \prod_{i=1}^{N}(1 - P(S_i, L_j)) \quad \textbf{(5)}$$

Here, S represents the collection of all sensor nodes that simultaneously sense the target point L_j in the area.

The definition of node coverage is as follows:

$$P = \frac{\sum_{i=1}^{N} A_i}{A} \quad \textbf{(6)}$$

As can be seen from Eq.(6), two items need to be calculated separately: the union of the coverage of all nodes $\sum_{i=1}^{N} A_i = \{A_1 \cup A_2 \cup A_3 \cdots \cup A_n\}$ and the size of the entire monitoring area A. The ratio of the two is the total node coverage rate P. But it is not convenient to calculate by definition. In order to simplify the operation, it can be replaced with another ratio, which is the ratio of the number of pixels in the sensing range of all sensor nodes to the total number of pixels in the target area. The calculation method is shown in Eq.(7).

$$P = \frac{\sum_{j=1}^{m*n} (S, L_j)}{m * n} \quad \textbf{(7)}$$

5.2 PSCA in the Dynamic Deployment of WSN

The goal of WSN node deployment is to hope to obtain the best sensor distribution and maximize the coverage. This requires a reasonable setting of the location of each sensor node. In the optimization algorithm, the optimal solution of the target problem can be found through some optimization mechanism. So we can consider using the PSCA to establish the optimal sensor node distribution. This paper focuses on the issue of maximizing the coverage rate. Assume that the monitoring area is a regular plane, and the sensor nodes have the following three properties: all nodes are the same, the nodes can be moved to a specified position and get the locations of other nodes in the sensing range in real time. Now establish the following correspondence between the problem domain and the solution space: the position coordinates of the sensor are the values of the dimensions in the solution, and the node coverage rate is the fitness value of the solution. Obviously, the best distribution of sensor nodes is the optimal solution of the optimization algorithm.

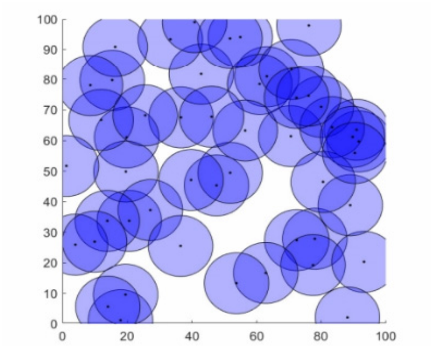
5.3 Simulation Results

To test the effectiveness of PSCA in solving WSN dynamic deployment problems, simulation experiments are performed for the SCA and PSCA's three strategies. In the process of testing the coverage optimization results of different algorithms, the same monitoring area is used, and the structure and number of sensor nodes deployed are also the same. Assume that the monitoring area is a square two-dimensional plane A with a side length of $100m$, i.e. $A = 100m \times 100m$. For processing convenience, discretize A into 100×100 pixels. In area A, 50 sensor nodes are randomly distributed, and their sensing radius are all $r = 10m$. The population size of PSCA is consistent with the number of sensors, so the number of individuals is set to $N = 50$. The algorithm iterates a total of 1000 times. The simulation experiments of each algorithm were performed 10 times independently. Table 4 shows the average coverage rate of the 10 experiments and Figure 8 shows the node distribution of the 10th optimization.

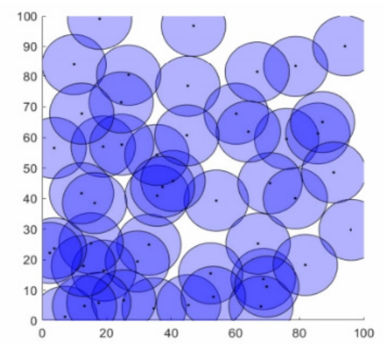
In the simulation process of dynamic deployment, the premature tendency of SCA and the shortcomings of difficulty in getting rid of local optimum are exposed. All this leads to poor node distribution. Although PSCA's strategy 1 increases the coverage rate to a certain extent, the improvement is limited. Strategy 2's optimization result is significantly improved compared to strategy 1 because it introduces a dominant genome which will effectively provide a motive force for the sustainable evolution of the population, thereby avoiding excessive population singularity and loss of evolutionary vitality. Strategy 3 has the best performance, achieves the ideal coverage results, and the nodes are evenly distributed, which benefits from its excellent global exploration and local exploitation capabilities.

6 Conclusion

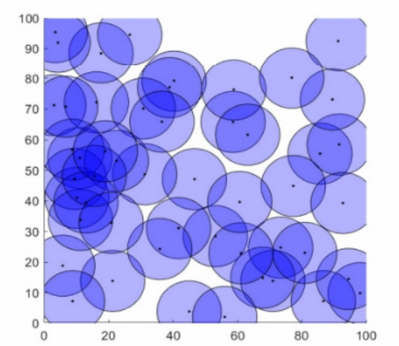
This paper first proposed parallel sine cosine algorithm (PSCA), involving the optimization of three types of objective functions: unimodal, multimodal, and composite functions. According to the characteristics of different types of functions, three communication strategies are proposed. Strategy 1 is a simple and effective choice in the optimization of unimodal functions. Strategy 2 based on dominant genome replacement achieves the optimization of multimodal functions well. Strategy 3 is inspired by Tabu search which can deal with the optimization of complex functions and it can also be selected when the



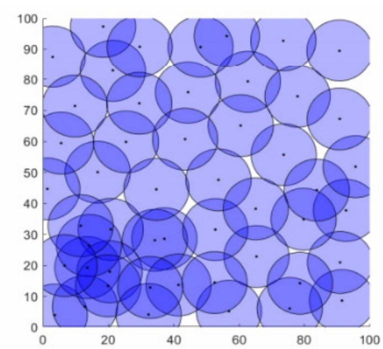
(a) Initialization



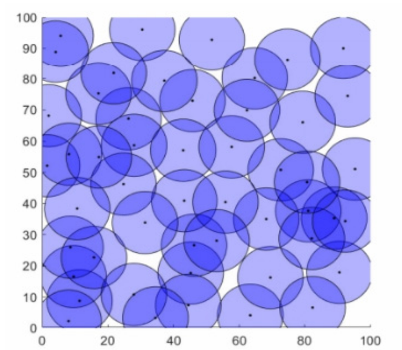
(b) SCA



(c) PSCA-Strategy 1



(d) PSCA-Strategy 2



(e) PSCA-Strategy 3

Figure 8. Sensor distribution

Table 4. Comparison of competition algorithms in coverage rate

Algorithm	Initialization	SCA	PSCA-Strategy 1	PSCA-Strategy 2	PSCA-Strategy 3
Coverage Rate	81.82%	84%	88.70%	94.11%	97.30%

function type is unknown. The effectiveness of the PSCA strategies have been tested on the benchmark functions. Experiment results indicated that all three strategies are superior to the original SCA and display excellent performance in the optimization process of corresponding types of functions. We have achieved the successful application of the proposed PSCA in WSN dynamic deployment, which further proves that it has high practical application value. In the next step, we will make more improvements on the basis of SCA parallelization by combining some promising methods [50-55].

References

[1] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, 1975.

[2] R. Eberhart, J. Kennedy, A New Optimizer Using Particle Swarm Theory, *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995, pp. 39-43.

[3] R. Storn, K. Price, Differential Evolution-A Simple and Efficient Heuristic for global Optimization Over Continuous Spaces, *Journal of Global Optimization*, Vol. 11, No. 4, pp. 341-359, December, 1997.

[4] X. S. Yang, A New Metaheuristic Bat-Inspired Algorithm, in: J. R. Gonzalez, D. A. Pelta, C. Cruz, G. Terrazas, N. Krasnogor (Eds.), *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, Vol. 284, Springer, Berlin, Heidelberg, 2010, pp. 65-74.

[5] M. Dorigo, M. Birattari, *Ant Colony Optimization*, Springer US, 2010.

[6] D. H. Wolpert, W. G. Macready, No Free Lunch Theorems for Optimization, *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp. 67-82, April, 1997.

[7] S. Mirjalili, SCA: A Sine Cosine Algorithm for Solving Optimization Problems, *Knowledge-Based Systems*, Vol. 96, pp. 120-133, March, 2016.

[8] S. Droste, T. Jansen, I. Wegener, Upper and Lower Bounds for Randomized Search Heuristics in Black-box Optimization, *Theory of Computing Systems*, Vol. 39, No. 4, pp. 525-544, July, 2006.

[9] B. Chopard, O. Pictet, M. Tomassinp, Parallel and Distributed Evolutionary Computation for Financial Applications, *Parallel Algorithms and Applications*, Vol. 15, No. 1-2, pp. 15-36, June, 2000.

[10] D. Abramson, J. Abela, A Parallel Genetic Algorithm for Solving the School Timetabling Problem, *15th Australian Computer Science Conference*, Hobart, Australia, 1992, pp.

1-11.

[11] J. F. Chang, S. C. Chu, J. F. Roddick, J. S. Pan, A Parallel Particle Swarm Optimization Algorithm with Communication Strategies, *Journal of Information Science and Engineering*, vol. 21, No. 4, pp. 809-818, July, 2005.

[12] P. W. Tsai, J. S. Pan, S. M. Chen, B. Y. Liao, S. P. Hao, Parallel Cat Swarm Optimization, *2008 International Conference on Machine Learning and Cybernetics*, Kunming, China, 2008, pp. 3328-3333.

[13] T. K. Dao, T. S. Pan, T. T. Nguyen, J. S. Pan, Parallel Bat Algorithm for Optimizing Makespan in Job Shop Scheduling Problems, *Journal of Intelligent Manufacturing*, Vol. 29, No. 2, pp. 451-462, February, 2018.

[14] X. S. Yang, Firefly Algorithm, Stochastic Test Functions and Design Optimisation, *International Journal of Bio-Inspired Computation*, Vol. 2, No. 2, pp. 78-84, March, 2010.

[15] X. S. Yang, M. Karamanoglu, X. He, Flower pollination Algorithm: A Novel approach for Multiobjective Optimization, *Engineering Optimization*, Vol. 46, No. 9, pp. 1222-1237, 2014.

[16] S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey Wolf Optimizer, *Advances in Engineering Software*, Vol. 69, pp. 46-61, March, 2014.

[17] N. Singh, S. B. Singh, A novel hybrid GWO-SCA Approach for Optimization Problems, *Engineering Science and Technology, an International Journal*, Vol. 20, No. 6, pp. 1586-1601, December, 2017.

[18] H. Nenavath, R. K. Jatoth, Hybridizing Sine Cosine Algorithm with Differential Evolution for Global Optimization and Object Tracking, *Applied Soft Computing*, Vol. 62, pp. 1019-1043, January, 2018.

[19] M. Meshkat, M. Parhizgar, A Novel Sine and Cosine Algorithm for Global Optimization, *2017 7th International Conference on Computer and Knowledge Engineering (ICCKE)*, Mashhad, Iran, 2017, pp. 60-65.

[20] M. A. Elaziz, D. Oliva, S. Xiong, An Improved Opposition-Based Sine Cosine Algorithm for Global Optimization, *Expert Systems with Applications*, Vol. 90, pp. 484-500, December, 2017.

[21] C. Qu, Z. Zeng, J. Dai, Z. Yi, W. He, A Modified Sine-cosine Algorithm Based on Neighborhood Search and Greedy Levy Mutation, *Computational intelligence and neuroscience*, Vol. 2018, pp. 1-19, July, 2018.

[22] F. Glover, Future Paths for Integer Programming and Links to Artificial Intelligence, *Computers & operations research*, Vol. 13, No. 5, pp. 533-549, 1986.

[23] F. Glover, Tabu Search—Part I, *ORSA Journal on Computing*, Vol. 1, No. 3, pp. 190-206, Summer, 1989.

[24] F. Glover, Tabu Search—Part II, *ORSA Journal on Computing*, Vol. 2, No. 1, pp. 4-32, Winter, 1990.

[25] J. Wang, C. Ju, Y. Gao, A. K. Sangaiah, G. J. Kim, A PSO Based Energy Efficient Coverage Control Algorithm for Wireless Sensor Networks, *Computers, Materials and Continua*, Vol. 56, No. 3, pp. 433-446, 2018.

[26] J. Wang, Y. Gao, W. Liu, A. K. Sangaiah, H. J. Kim, Energy Efficient Routing Algorithm with Mobile Sink Support for Wireless Sensor Networks, *Sensors*, Vol. 19, No. 7, Article No. 1494, April, 2019.

[27] P. Hu, J. S. Pan, S. C. Chu, Q. W. Chai, T. Liu, Z. C. Li, New Hybrid Algorithms for Prediction of Daily Load of Power Network, *Applied Sciences*, Vol. 9, No. 21, Article No. 4514, November, 2019.

[28] Z. G. Du, J. S. Pan, S. C. Chu, H. J. Luo, 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, January, 2020.

[29] J. Wang, X. Gu, W. Liu, A. K. Sangaiah, H. J. Kim, An Empower Hamilton Loop Based Data Collection Algorithm with Mobile Agent for WSNs, *Human-centric Computing and Information Sciences*, Vol. 9, pp. 1-14, May, 2019.

[30] W. Li, Z. Chen, X. Gao, W. Liu, J. Wang, Multimodel Framework for Indoor Localization Under Mobile Edge Computing Environment, *IEEE Internet of Things Journal*, Vol. 6, No. 3, pp. 4844-4853, June, 2019.

[31] C. Yin, S. Ding, J. Wang, Mobile Marketing Recommendation Method Based on User Location Feedback, *Human-centric Computing and Information Sciences*, Vol. 9, pp. 1-17, May, 2019.

[32] J. Wang, Y. Gao, X. Yin, F. Li, H. J. Kim, An Enhanced PEGASIS Algorithm with Mobile Sink Support for Wireless Sensor Networks, *Wireless Communications and Mobile Computing*, Vol. 2018, pp. 1-9, December, 2018.

[33] J. Wang, Y. Gao, W. Liu, A. K. Sangaiah, H. J. Kim, An Intelligent Data Gathering Schema with Data Fusion Supported for Mobile Sink in Wireless Sensor Networks, *International Journal of Distributed Sensor Networks*, Vol. 15, No. 3, pp. 1-9, March, 2019.

[34] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless Sensor Networks: A Survey, *Computer networks*, Vol. 38, No. 4, pp. 393-422, March, 2002.

[35] C. H. Chen, C. A. Lee, C. C. Lo, Vehicle Localization and Velocity EStimation Based on Mobile Phone Sensing, *IEEE Access*, Vol. 4, pp. 803-817, February, 2016.

[36] C. I. Wu, H. Y. Kung, C. H. Chen, 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, August, 2014.

[37] G. Fan, S. Jin, Coverage Problem in Wireless Sensor Network: A Survey, *Journal of networks*, Vol. 5, No. 9, pp. 1033-1040, September, 2010.

[38] C. Zhu, C. Zheng, L. Shu, G. Han, A Survey on Coverage and Connectivity Issues in Wireless Sensor Networks, *Journal of Network and Computer Applications*, Vol. 35, No. 2, pp. 619-632, March, 2012.

[39] L. Li, H. Li, T. L. Zhang, B. R. Tao, W. X. Zhang, Strategy of WSN Coverage Optimization by Improved Artificial Fish Swarm Algorithm, *Microelectronics & Computer*, Vol. 30, No. 2, pp. 83-86, February, 2013.

[40] C. W. Tsai, P. W. Tsai, J. S. Pan, H. C. Chao, Metaheuristics for the Deployment Problem of WSN: A Review, *Microprocessors and Microsystems*, Vol. 39, No. 8, pp. 1305-

1317, November, 2015.

[41] J. S. Pan, L. P. Kong, T. W. Sung, P. W. Tsai, V. Snášel, α-Fraction First Strategy for Hierarchical Model in Wireless Sensor Networks, *Journal of Internet Technology*, Vol. 19, No. 6, pp. 1717-1726, November, 2018.

[42] T. T. Nguyen, J. S. Pan, T. K. Dao, An Improved Flower Pollination Algorithm for Optimizing Layouts of Nodes in Wireless Sensor Network, *IEEE Access*, Vol. 7, pp. 75985-75998, June, 2019.

[43] T. T. Nguyen, J. S. Pan, T. Y. Wu, T. K. Dao, T. D. Nguyen, Node Coverage Optimization Strategy Based on Ions Motion Optimization, *Journal of Network Intelligence*, Vol. 4, No. 1, pp. 1-9, February, 2019.

[44] J. Chen, E. Shen, Y. Sun, The Deployment Algorithms in Wireless Sensor Net Works: A Survey, *Information Technology Journal*, Vol. 8, No. 3, pp. 293-301, 2009.

[45] N. X. Liu, J. S. Pan, T. T. Nguyen, A Bi-population QUasi-Affine TRansformation Evolution Algorithm for Global Optimization and Its Application to Dynamic Deployment in Wireless Sensor Networks, *EURASIP Journal on Wireless Communications and Networking*, Vol. 2019, Article No. 175, July, 2019.

[46] X. Liu, D. He, Ant Colony Optimization with Greedy Migration Mechanism for Node Deployment in Wireless Sensor Networks, *Journal of Network and Computer Applications*, Vol. 39, pp. 310-318, March, 2014.

[47] X. Yu, J. Zhang, J. Fan, T. Zhang, A Faster Convergence Artificial Bee Colony Algorithm in Sensor Deployment for Wireless Sensor Networks, *International Journal of Distributed Sensor Networks*, Vol. 9, No. 10, pp. 1-9, October, 2013.

[48] K. N. Krishnanand, D. Ghose, Glowworm Swarm Optimization for Simultaneous Capture of Multiple Local Optima of Multimodal Functions, *Swarm intelligence*, Vol. 3, No. 2, pp. 87-124, June, 2009.

[49] W. H. Liao, Y. Kao, Y. S. Li, A Sensor Deployment Approach Using Glowworm Swarm Optimization Algorithm in Wireless Sensor Networks, *Expert Systems with Applications*, Vol. 38, No. 10, pp. 12180-12188, September, 2011.

[50] C. Sun, Y. Jin, R. Cheng, J. Ding, J. Zeng, Surrogate-assisted Cooperative Swarm Optimization of High-dimensional Expensive Problems, *IEEE Transactions on Evolutionary Computation*, Vol. 21, No. 4, pp. 644-660, August, 2017.

[51] J. S. Pan, P. Hu, S. C. Chu, Novel Parallel Heterogeneous Meta-Heuristic and Its Communication Strategies for the Prediction of Wind Power, *Processes*, Vol. 7, No. 11, Article No. 845, November, 2019.

[52] J. S. Pan, L. P. Kong, T. W. Sung, P. W. Tsai, V. Snášel, 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, July, 2018.

[53] J. S. Pan, C. Y. Lee, A. Sghaier, M. Zeghid, J. Xie, Novel Systolization of Subquadratic Space Complexity Multipliers Based on Toeplitz Matrix-Vector Product Approach, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 27, No.7, pp. 1614-1622, July, 2019.

[54] S. C. Chu, X. S. Xue, J. S. Pan, X. J. Wu, Optimizing Ontology Alignment in Vector Space, *Journal of Internet Technology*, Vol. 21, No. 1, pp. 15-22, January, 2020.

[55] Z. Meng, J. S. Pan, 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, March, 2019.

Biographies

Fang Fan received the master's degree from Tiangong University, Tianjin, in 2006. She is a lecturer in Shandong University of Science and Technology. She is currently working toward her Ph.D degree in Shandong University of Science and Technology. Her research interests include intelligent computing, wireless sensor networks, and network security.

Shu-Chuan Chu received the Ph.D. degree in 2004 from the Flinders University of South Australia. Currently, She is the Research Fellow with PhD advisor in the College of Computer Science and Engineering of Shandong University of Science and Technology from September 2019. Her research interests are mainly in Swarm Intelligence, Intelligent Computing and Data Mining.

Jeng-Shyang Pan received the Ph.D. degree in electrical engineering from the University of Edinburgh, U.K. in 1996. He is currently the Professor in the College of Computer Science and Engineering, Shandong University of Science and Technology. His current research interest includes the information hiding, artificial intelligence and wireless sensor networks.

Qingyong Yang is an undergraduate student in Shandong University of Science and Technology, Qingdao, Shandong, China. His research interests are intelligent computation and wireless sensor networks.

Huiqi Zhao received his Ph.D. degree from Shandong University of Science and Technology, Qingdao, Shandong, China. He is a lecturer in department of information engineering, Shandong University of Science and Technology, Taian, Shandong. His research

interests are computer networks, wireless body area networks, healthcare applications and data privacy protection.