Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 8820907, 10 pages
https://doi.org/10.1155/2020/8820907

Research Article

Coverage Optimization of Sensors under Multiple Constraints Using the Improved PSO Algorithm

Haifeng Ling,[1] Tao Zhu ,[1] Weixiong He,[2] Hongchuan Luo ,[1] Qing Wang,[1] and Yi Jiang[1]

[1]*Field Engineering College, Army Engineering University of PLA, Nanjing, China*
[2]*Armed Police Academy, Beijing, China*

Correspondence should be addressed to Tao Zhu; lgdxiaozhu1987@outlook.com

Received 3 September 2020; Revised 8 September 2020; Accepted 10 September 2020; Published 22 September 2020

Academic Editor: Xingling Shao

Sensor deployment is an important issue in wireless sensor network (WSN), which is a typical nonlinear system. Conditions of both area coverage and point coverage should be considered in research studies on sensor coverage. It is generally necessary to ensure high coverage ratio of area when controlling sensor locations, and covering specific point targets to ensure long lifetime is also important sometimes. In current studies, swarm intelligence algorithms such as particle swarm optimization (PSO) are widely used to solve the sensor deployment problem in WSN. In this paper, coverage rate and network life indicators are analyzed comprehensively with establishment of a more general K-coverage model. In related calculation examples with different coverage requirements including target coverage, area coverage, and boundary coverage, several improved algorithms based on PSO are applied to solve the problem in the paper. Simulation results show that the improved algorithms can achieve a good performance and deployment effect.

1. Introduction

Wireless sensor network (WSN) is generally composed of a large number of sensor nodes distributed in a specific area whose main task is to jointly monitor the region [1]. WSN can be used in many scenarios such as battlefield surveillance using UAV [2, 3], security management with early warning, big data of IoT [4], and so on, which plays significant roles in both military and civil fields. However, sensor nodes have their own limitations such as small sense range and weak processing power [5, 6]. When evaluating the deployment of sensor nodes in WSN, the following two performance indicators should be generally considered: one is the coverage ability of the sensor and the other is the life cycle of the whole network [7].

The coverage problem of the sensor node deployment can be divided into two types: area coverage and target coverage. The area coverage problem focuses on monitoring the entire space, while the target coverage problem focuses on some specific points in the monitoring area. The main index of area coverage is coverage rate, which can measure the service quality of wireless sensor networks. Sometimes, it is necessary to monitor several specific points in the study of the sensor coverage problem, which is called the target coverage problem [8] that usually consists of simple coverage and Q-coverage. Each target only needs one sensor to cover itself in the simple coverage problem. As to Q-coverage, target i is set to be monitored by at least q_i sensor nodes, and the q_i of each target can be the same or different.

Due to the energy limitation of WSN, the entire network has a life cycle. The lifetime of WSN, which refers to the duration until it is unable to monitor the required area or point, is an important criterion for determining the efficiency of WSN [9]. Sensor nodes are generally powered by batteries, which cannot be recharged or replaced due to the condition of coverage area. Therefore, the sensor placement should also be optimized to extend the network lifetime.

The problem of determining the location of sensor nodes so that the network has good properties is called optimal deployment of sensors, which is a typical NP-hard problem [10]. So far, many kinds of methods are proposed to solve it, such as particle swarm optimization (PSO) [11], constrained control policy [12, 13], and neuroadaptive dynamic theory [14, 15]. In fact, there is no precise polynomial-time algorithm to solve NP-hard problem, and the only hope is to find a better feasible solution. In this paper, several improved PSO algorithms are proposed to solve the sensor deployment problem. The main features and contributions of this paper are highlighted as follows:

Based on the comprehensive analysis of both coverage ratio and network lifetime, a more general K-coverage problem is sum up, and a sensor optimal deployment model considering point target and regional coverage is established.

A new PSO-based improvement method is proposed, and simulation experiments are carried out for three conditions with other two typical improved PSO algorithms to verify the effectiveness of the algorithm.

The remaining part of this paper is arranged as follows: related work is introduced in Section 2. K-coverage problem is analyzed and the corresponding sensor coverage model is established in Section 3. In Section 4, a new improved PSO algorithm and solution to the K-coverage problem is described. Simulation experiments and comparison results are given in different cases in Section 5. Finally, there are conclusion and prospects in Section 6.

2. Related Works

A variety of approaches have been proposed to address the sensor deployment problem in WSN so far, but the PSO algorithm has become one of the most widely used methods due to its high search efficiency [16, 17]. Mini et al. [18] have comprehensively studied the coverage problem in WSN by comparing a variety of improved PSO algorithms; however, the optimization goal is only to maximize the network lifetime. As to the target coverage problem, Yarinezhad and Hashemi [19] have transformed network lifetime condition into requirements of target points, which must be covered by multiple sensor nodes, using an improved PSO algorithm using fuzzy logic [20] to obtain a long network lifetime without taking regional coverage rate likewise.

Yu et al. [21] uses centralized and distributed protocols, which take the remaining energy into consideration, but do not consider how to determine the location of sensor nodes to solve the coverage problem in WSN. Deploying a large number of sensor nodes in a random manner will cause a lot of sensing overlap areas; so, it is necessary to optimize the location of nodes to improve the coverage ratio. Zhang et al. [22] uses standard particle swarm optimization (SPSO) to adjust the position of nodes so that node distribution is more even, the perception blind area is significantly reduced, and the coverage of the entire network is greatly improved after optimization. Aiming at the shortcomings of basic PSO, such as slow convergence speed and easy local optimality, Wang et al. [23] regard improving coverage ratio as the main optimization goal and introduced logistic chaos into a quantum-behaved particle swarm optimization (QPSO) algorithm in solving the coverage problem in WSN.

Due to the characteristics of less control parameters, simple implementation, and good robustness, the PSO algorithm can often obtain the approximate optimal solution when solving search problems, which has become a popular swarm intelligence optimization method [24]. To have a better control over the search scope, standard particle swarm optimization (SPSO) with an inertia weight factor w is put forward by Shi and Eberhart, which can achieve better results in the application of some certain issues [25]. In SPSO framework, each particle, represented as a potential solution for the problem, is evaluated by a function called fitness function. To achieve the best solution, every particle moves to search in a D-dimension space, and the velocity $\mathbf{v_i}$ and the position $\mathbf{x_i}$ are updated as follows:

$$\begin{aligned} \mathbf{v_i}(t+1) &= w \cdot \mathbf{v_i}(t) + c_1 r_1 (\mathbf{p_i} - \mathbf{x_i}) + c_2 r_2 (\mathbf{g} - \mathbf{x_i}), \\ \mathbf{x_i}(t+1) &= \mathbf{x_i}(t) + \mathbf{v_i}(t+1), \end{aligned} \tag{1}$$

where $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ is the position of i^{th} particle, $v_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$ is the velocity of particle i, t represents the generation number, c_1 and c_2 are local and global learning factors, respectively, r_1, r_2 are random numbers between 0 and 1, $\mathbf{p_i} = (p_{i1}, p_{i2}, \ldots, p_{iD})$, or called **pbest**, stands as the previous best position for particle i, and $\mathbf{g} = (g_1, g_2, \ldots, g_D)$, also named **gbest**, is the global best position found so far in the entire swarm. The updates will be iterated until an acceptable amount for **gbest** is reached or the number of iterations reach a fixed amount called $t_{\max}$.

Theoretical research shows that the inertia weight factor w can balance the "exploration" and "exploitation" of particles. When w is selected as some specific values, such as $w = 0.7298$, $c_1 = c_2 = 1.494$, SPSO can ensure convergence and get good results in many benchmark functions [26, 27]. However, SPSO is easy to fall into local optimal state prematurely in many optimization problems. Therefore, adaptive inertia weight SPSO with variable inertia weight factor appears. Many scholars believe that global search ability is strong when w is large and local search ability is strong when w is small. The common w adaptation mechanism usually decreases with the iteration time [28] as shown in the following:

$$w(t) = \frac{t_{\max} - t}{t_{\max}} (w_0 - w_1) + w_1. \tag{2}$$

The particle motion state in the SPSO algorithm needs to be expressed by both velocity and position. In quantum space, motion state of the particle with momentum and energy can be directly expressed by its wave function, and the properties of particle are completely different from Newton space. Sun et al. [29] proposed quantum behaved particle swarm optimization (QPSO) from the perspective of quantum mechanics. Based on the δ potential well, the model considers that particles have quantum behavior and can search in the whole space of feasible solutions, with high swarm intelligence and strong collaborative ability. According to the uncertainty principle proposed by Heisenberg, particle position and velocity cannot be accurately determined at the same time, but the statistical significance of the wave function can be used to determine the occurrence of a particle at a certain position and a certain moment by the probability density function. Results of the standard test function show that the global search performance of the QPSO algorithm is better than that of the SPSO algorithm.

In order to ensure its convergence, each particle i in the QPSO algorithm must converge to its own $\mathbf{Q_i}$ point, which is expressed as follows:

$$\mathrm{Q}_i = \left(\frac{(\varphi_1 \times p_i + \varphi_2 \times g)}{(\varphi_1 + \varphi_2)} \right), \quad (3)$$

where $\varphi_1 = r\,\mathrm{and}\,(0,1)$, and $\varphi_2 = r\,\mathrm{and}\,(0,1)$. At the same time, a global point **mbest** is introduced to calculate the next generation of particles, which is defined as an average value of the best local positions of all particles:

$$\mathbf{mbest} = \sum_{i=1}^{\mathrm{ps}} \mathbf{p_i}, \quad (4)$$

where ps means the particle number, and iterative equation of particle i is as follows:

$$\mathbf{x_i}(t+1) = \mathbf{Q_i} \pm \beta \cdot \left|\mathbf{mbest} - \mathbf{x_i}(t)\right| \cdot \ln\left(\frac{1}{u}\right), \quad (5)$$

where $u = r\,\mathrm{and}\,(0,1)$, and β is called shrinkage-expansion coefficient, which controls the convergence speed. In many cases, value of β is often given as follows:

$$\beta(t) = 0.5 + (1.0 - 0.5) \times \frac{(t_{\max} - t)}{t_{\max}}. \quad (6)$$

The initial distribution of particles also affects the convergence speed and solution quality of algorithm. If the initial particles are evenly distributed in the solution space, the speed of optimization and the quality of the solution will be naturally improved. The traditional logistic chaos has good distribution characteristics [30], and its iterative equation is as follows:

$$x_{n+1} = \mu x_n (1 - x_n), \quad (7)$$

where μ is the control parameter and $\mu \in [0,4]$; x_n is the iterative state value and $x_n \in (0,1)$; and system is in a chaotic state when $\mu = 4$. Using the logistic chaotic map in the initialization of the algorithm, it makes the particle distribution cover a wide area and helps to prevent premature convergence of the population.

In order to study the sensor placement problem under complex conditions, it is necessary to comprehensively analyze network lifetime and coverage ratio index so as to establish a general optimization model. At the same time, several algorithms are needed to carry out many times of simulations to achieve a better deployment effect. A new improved PSO algorithm is put forward in this paper, and two typical algorithms of SPSO and QPSO are seen as a benchmark for their good generality.

3. Problem Description

Complex sensor coverage problems usually have resource conditions of sensing range, network lifetime, node quantity, and so on whose practical application includes different scene conditions such as target coverage, area coverage, and boundary coverage. However, it is generally required to maximize the coverage ratio. In the following part, coverage problem considering network lifetime is analyzed, and the corresponding sensor model is given.

3.1. K-Coverage Problem. Suppose there are M sensor nodes $\{s_1, s_2, \ldots, s_M\}$ and N targets $\{T_1, T_2, \ldots, T_N\}$ in the monitoring area. If the distance between the sensor node s_i and target point T_j is shorter than R_i, s_i will cover T_j. Thus, the following covering matrix can be defined:

$$C_{ij} = \begin{cases} 1, & \text{if } s_i \text{ covers } T_j, \\ 0, & \text{otherwise}, \end{cases} \quad (8)$$

where $i = 1, \ldots, M$, and $j = 1, \ldots, N$.

The network lifetime of WSN is mainly related to the location and energy of sensor nodes. We define $b_i' = (b_i/e_i)$ to present the lifetime of the battery, where b_i is the initial battery power, and e_i is the energy consumption rate of the node s_i. The upper bound of the lifetime can be calculated as follows [31]:

$$U = \min_j \frac{\sum_i C_{\mathrm{ij}}^* b_i'}{q_j}, \quad (9)$$

where q_j is a positive integer, and it can be set according to the importance of the target T_j, $j = 1, 2, \ldots, N$.

Suppose that the region is deployed with homogeneous sensor nodes, which means that R_i and b_i' have a fixed value for every sensor node. If each target is equally important (i.e., q_j is also the same) and network lifetime U is given, it will be equivalent that C_{ij} must satisfy the corresponding K-coverage constraint where K is a positive integer calculated through lifetime condition.

3.2. Sensor Model. In order to optimize sensor deployment problems under different conditions, we assume that the monitoring area is a two-dimensional plane. The wireless sensor nodes adopt the Boolean sensing model, i.e., probability of the target point in the sensing range is 1; otherwise, it is 0, to simplify the coverage problem. When the number of sensors is large, the total coverage rate of all nodes to the monitoring area is difficult to be solved by formula. Therefore, the region is divided into grid points of equal size, which can be further equivalent to pixels with a discrete accuracy of 1 [32].

The common sensing range is like a disk centered on the location of the sensor node. The Boolean sensing model can be described mathematically as that the coordinates of the sensor node is $s_i = (x_i, y_i)$, and the sensing radius is R_i in a configured Euclid space at this time. Then, the probability of pixel point a with coordinates (x, y), which can be perceived, is as follows:

$$p(a, s_i) = \begin{cases} 1, & d(a, s_i) \le R_i, \\ 0, & \text{otherwise}, \end{cases} \tag{10}$$

where the Euclid distance $d(a, s_i) = \|a - s_i\|_2$. Note that the sensor set $S = \{s_i\}$, $i = 1, \ldots, M$. Pixel point a may be perceived by different sensors, so its comprehensive perception probability is as follows:

$$p(a, S) = 1 - \prod_{s_i \in S} [1 - p(a, s_i)]. \tag{11}$$

Assuming that the monitoring area is equivalent to $u \times v$ pixel points, the coverage ratio f of the sensor deployment area can be defined as follows:

$$f = \frac{\sum_{x=1}^{m} \sum_{y=1}^{n} p(a, S)}{u \times v}. \tag{12}$$

The above definition of the coverage rate is the main general objective function for optimal deployment of sensors. For K-coverage problems with different requirements, only specific constraints need to be added. When $K = 0$, it is an unconstrained coverage optimization problem; when $K = 1$, it indicates that some key targets must be covered; and when $K \ge 2$, it may have K-coverage constraint with specific network lifetime conditions, to ensure that each important point must be within the coverage range of K sensor nodes.

4. Proposed Method

In this paper, we consider that the sensor nodes can be determined before deployed. In this section, we propose another improved PSO method to solve the sensor deployment approach for the K-coverage problem, and the task is mainly to choose location points of the sensor nodes in the area.

4.1. CLPSO Algorithm and Its Improvement. Original PSO algorithm is easy to fall into local extremum when solving complex multimodal problems, leading to premature convergence. Each particle learns from its own optimal value and global optimal value at the same time, and all particles will be affected if the global optimal value falls into a local extreme value. In addition, each particle learns from all dimensions of its own optimal value in the speed formula of the original PSO algorithm, but the own optimal value is usually not optimal in all dimensions. Liang et al. [33] put forward the comprehensive learning particle swarm optimization (CLPSO) algorithm, in which the social part of particles learning from the global optimal value is not used. It introduces a comprehensive learning strategy, using **pbest** of all particles to construct learning samples, which can effectively promote the information exchange between particles in different dimensions and avoid trapping in the local extremum. The velocity update formula of particle i is as follows:

$$v_i^d = \mathrm{wv}_i^d + \mathrm{cr}1_i^d\left(\mathrm{pbest}_{\mathrm{fi}(d)}^d - x_i^d\right), \tag{13}$$

where $\mathrm{pbest}_{\mathrm{fi}(d)}^d$ represents the optimal value of particle i in the d^{th} dimension of the learning sample. $\mathbf{f_i} = [f_i(1), f_i(2), \ldots, f_i(D)]$, which is a set of learning samples constructed by particles randomly selected based on the learning probability parameter Pc_i. Pc_i can be calculated by the following formula:

$$\mathrm{Pc}_i = 0.05 + 0.45 \cdot \frac{(\exp(10(i-1)/\mathrm{ps} - 1) - 1)}{(\exp(10) - 1)}, \tag{14}$$

where $i \in (1, \mathrm{ps})$, and ps is the number of particles. Suppose that $\mathbf{p_i}$ of particle i has not been updated after m generation evolution, learning samples are constructed according to the following steps:

Step 1: As to particle i, generate a random number r and with uniform distribution within (0, 1) in a dimension and compare it with Pc_i.

Step 2: If r and $> \mathrm{Pc}_i$, particle i will learn from its historical optimal value. Otherwise, it learns from the historical optimal value of other particles.

Step 3: When learning from the historical optimal value of other particles, a selection operator of tournament mechanism is added. Choose two particles whose speed has not been updated in the current dimension randomly, and then select the particle with better fitness value as the learning sample.

Step 4: Choose another dimension and repeat Step 1–Step 3 until all dimensions obtained learning samples.

With each particle's potential search space increased, the diversity increases. As each particle is possibly a good area, the search of CLPSO is neither blind nor random. Compared to the original PSO, CLPSO searches more promising regions to find the global optimum [33].

However, global and local exploration ability of the swarm intelligence algorithm should be balanced in different ways. The other ability will usually decrease when one ability is improved. For example, the SPSO algorithm has fast convergence speed and strong local search ability, while the CLPSO algorithm promotes the full exchange of information among particles and enhances the global exploration ability, with the local search ability decreasing and the convergence speed slowing down [34]. To solve this dilemma, we further improve the CLPSO algorithm and call it the ICLPSO algorithm.

First, the global optimal value is also added into population evolution, which can help to improve the convergence speed of particles. The new speed update formula is as follows:

$$v_i^d = \omega \mathrm{v}_i^d + c_1 r 1_i^d\left(\mathrm{pbest}_{\mathrm{fi}(d)}^d - x_i^d\right) + c_2 r 2_i^d\left(\mathrm{gbest}^d - x_i^d\right). \tag{15}$$

Furthermore, introduction of particle mutation strategy usually helps particles escape from the local optimal value and improves the global exploration ability. Yao et al. [35] have found that the offspring of particles after Cauchy mutation are farther away from their parents than Gaussian mutation, which can effectively help particles escape from the local optimal value. Therefore, Cauchy mutation is introduced into the CLPSO algorithm in this paper. In this regard, the standard Cauchy mutation strategy of $t = 1$ is adopted, and it applies disturbance mutation to particles whose evolution has stagnated for more than m generations. The specific mutation formula is as follows [36]:

$$\mathbf{x}_i^{\text{new}} = \mathbf{x}_i + \text{Cauchy}(0, 1), \tag{16}$$

where Cauchy (0, 1) is the standard Cauchy distribution. Its generating function of random variable is $y = \tan(\pi \cdot (N - 0.5))$, and N is the random variable obeying the standard normal distribution within (0, 1).

The complete flowchart of ICLPSO is given in Figure 1.

4.2. Solving K-Coverage Problem of Sensor Deployment. We use the PSO-based sensor deployment method in this paper. It mainly consists of particle encoding scheme, fitness function of optimization objective, and update and constraint followed by termination criteria.

4.2.1. Particle Encoding Scheme. To solve the sensor deployment problem by using different PSO algorithms, we suppose that each particle is a feasible solution. Assume we have a WSN with M nodes and a sensor deployment area of size $u \times v$. Two dimensions are required to deploy each sensor node. Hence, each feasible solution for the problem is a particle, which is equal to M pairs as $\{(x_1, y_1), (x_2, y_2), \ldots, (x_M, y_M)\}$, where $0 \le x_i \le u$ and $0 \le y_i \le v$ for each $i \in \{1, 2, \cdots, M\}$, and (x_i, y_i) is the location of the i^{th} sensor node. Every swarm is a collection of particles. In other words, when solving the sensor deployment problem by using the above algorithms, a swarm of size l is indeed a collection of l different deployments of the sensor nodes in the area.

4.2.2. Fitness Function of Optimization Objective. Optimization objective here can be divided into two parts. One is constraint of K-coverage, which includes target coverage requirement and network lifetime condition. The other is coverage rate of the monitoring area because it can be optimized by selecting different node locations.

In order to achieve a good performance of WSN, the main goal in this paper is to determine the location of the sensor nodes, so that the required conditions are satisfied, and the coverage area is maximized. Therefore, the fitness function is f in equation (12), which presents the coverage ratio.

4.2.3. Update and Constraint. The position and velocity of each particle are updated in each iteration (QPSO only has position updating). However, the algebraic steps of addition and subtraction operations in these equations may cause the new position of the particles to be out of range (not in $u \times v$). Therefore, the proposed algorithm handles the positions of the particles in such a way that they fall in the desired range. Three following rules handle these scenarios for a particle (x_i, y_i):

(1) If x_i or y_i are negative, replace it with a new random number

(2) If $x_i > u$, then $x_i = u$

(3) If $y_i > v$, then $y_i = v$

It must be noted that the process of updating **gbest** need to consider its specific condition. In fact, it should update so that it becomes a feasible solution for the problem. In other words, **gbest** must be updated such that it satisfies the K-coverage constraint caused by WSN lifetime. In order to achieve this goal, in each step where a particle is updating, the required condition is checked; if it is not met, then another particle satisfying the condition will be updated to the fitness function.

4.2.4. Termination Criteria. The iterations keep running until the termination criteria are met. A predefined iteration number is the termination criterion in our algorithm, and the convergence results can be analyzed after the algorithm is terminated.

5. Simulation and Analysis

In order to further evaluate the performance of different algorithms, a series of simulations are carried out in this section.

5.1. Target Coverage. Consider the simple coverage problem of target points which means $K = 1$. Assume that there is an area of size $32 \times 32\,\text{m}^2$ with 100 targets whose locations are set randomly, and there are 4 sensor nodes with a same sensing range of 12 m.

Parameters set in four different algorithms are as follows: ps = 30 and $t_{\max} = 300$. As to CLPSO and ICLPSO, $m = 5$. After many times of simulations, optimization results of sensor deployment given by these four algorithms are nearly the same, as shown in Figure 2. However, SPSO sometimes may fall into local optimum. To compare performance of convergence, a convergence curve of each algorithm is selected, as shown in Figure 3, where cover index is defined as the coverage rate of target points.

5.2. Area Coverage. We assume that the deployment area size is $800\,\text{m} \times 800\,\text{m}$, and there is a condition with several important points ($M = 5$), which must be covered. The number of the sensor nodes is 25, and sensing range of each node is fixed and equals to 88 m. Relevant parameters in this part are given in Table 1, and the four algorithms mentioned above are all executed.

After 20 runs in the same configuration, we find different sensor deployment schemes using different algorithms. And the average results of coverage ratio given by every algorithm are shown in Table 2.

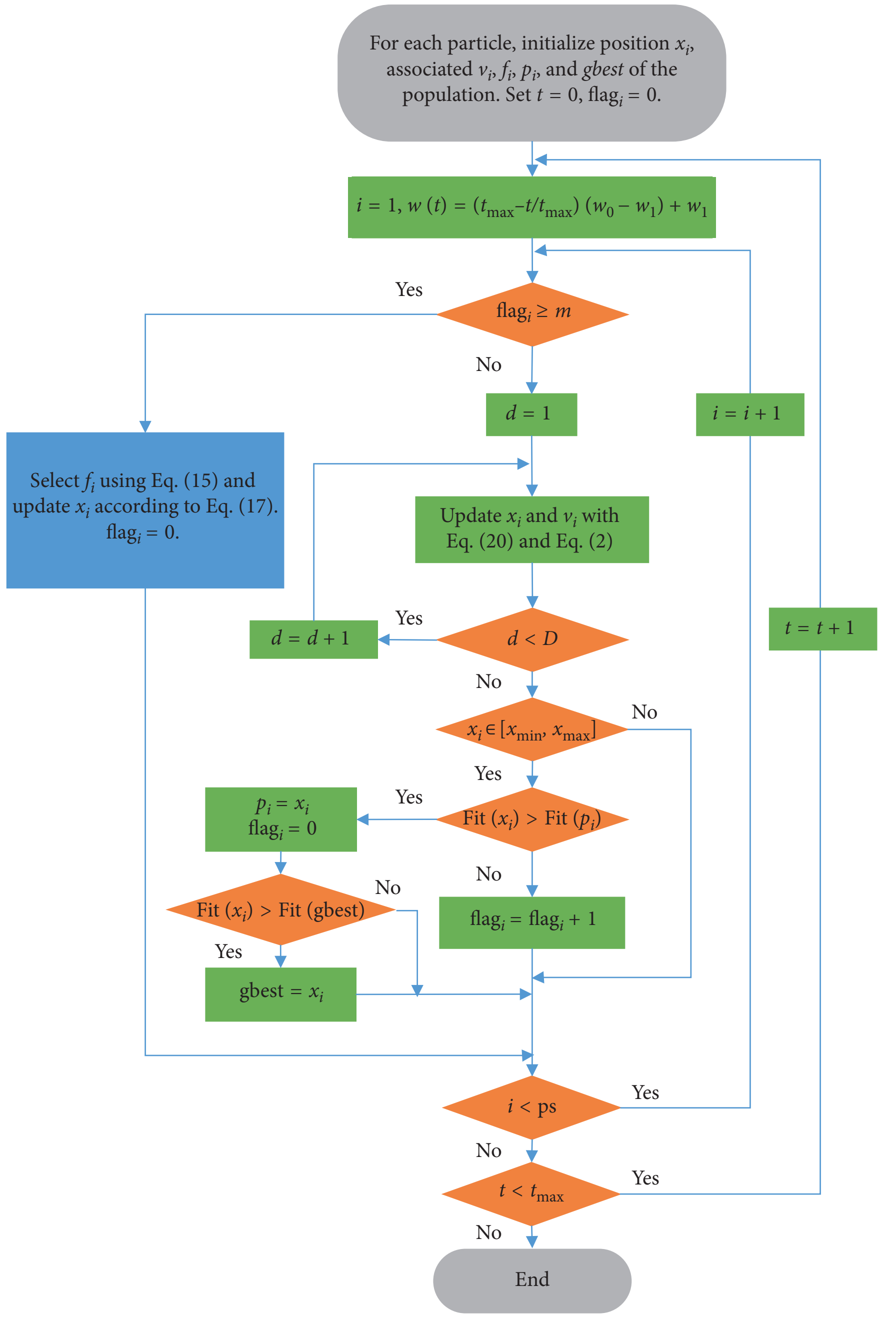


Figure 1: Flowchart of ICLPSO.

According to the results, the ICLPSO algorithm can get a better performance, while the optimization result of CLPSO is the poorest. In fact, CLPSO is found difficult to meet the coverage condition in some experiments resulting from its slow convergence. For further comparison, best locations of sensor nodes calculated by the four algorithms are shown in Figure 4 with coverage ratio number in brackets.

5.3. Boundary Coverage. In this part, boundary covering condition is considered and converted to K-coverage problem. Sense nodes sometimes are required to cover a certain boundary, and we can simplify the problem into coverage of a series of targets. As shown in Figure 5, there are 16 target points, which form a square boundary, and the circles with dotted line show that these targets can communicate with each other. In this way, it is reasonable that these 16 points can approximately represent the whole boundary.

As to the boundary points above, they may be important targets, which require a certain condition of WSN lifetime, and we can set different kinds of lifetime requirements according to equation (9). Therefore, this kind of problem can also be solved by the K-coverage model built in this paper. Assume that the deployment area is also 800 m × 800 m, while the boundary is a square with side

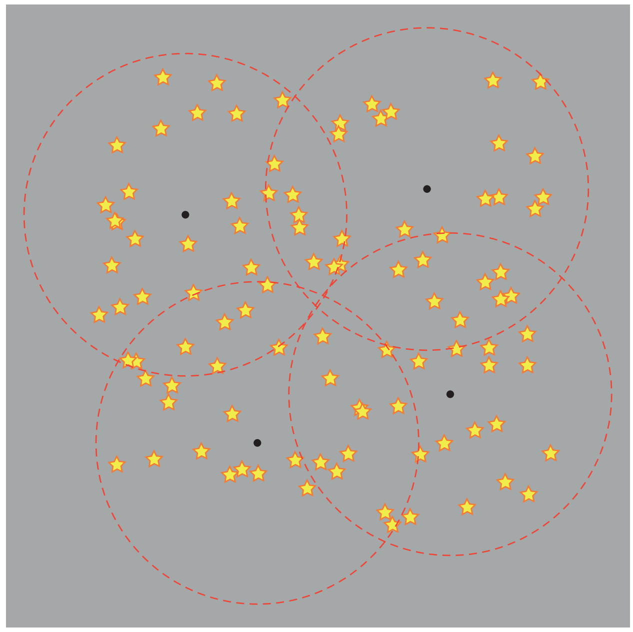


Figure 2: Optimization results of sensor deployment.

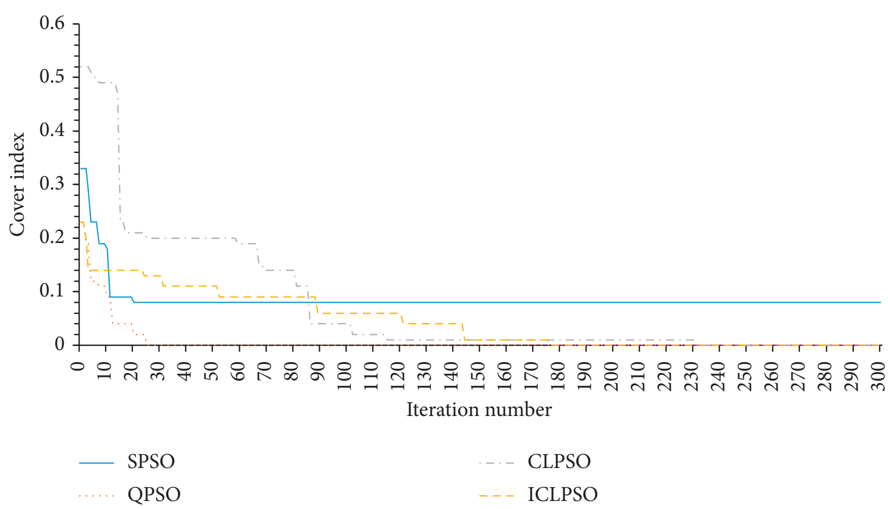


Figure 3: Selected convergence curve of each algorithm.

Table 1: Parameters of wide area coverage.

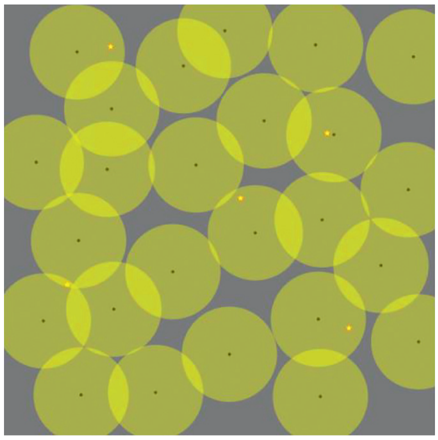
Parameter	Value
Region size	800 m × 800 m
Sensor nodes number N	25
Sensing range	88 m
ps	10
m	5
$t_{\max}$	1000

length 320 m. Specific simulation parameters are summarized in Table 3, and algorithms of SPSO, QPSO, and ICLPSO are conducted to have a comparison due to the difficult convergence of CLPSO.

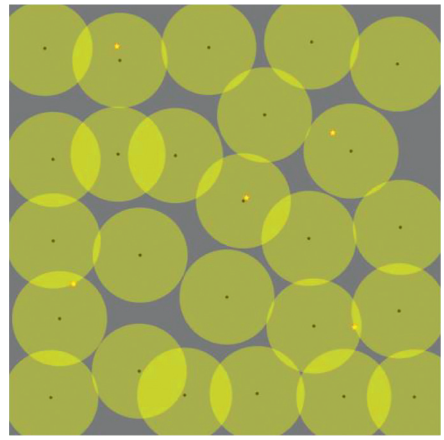
Table 2: Average results of coverage ratio.

Condition	SPSO	QPSO	CLPSO	ICLPSO
$K = 1$	0.8027	0.8013	0.7904	**0.8477**

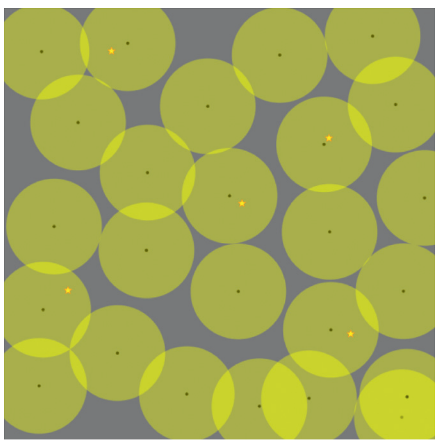
In order to compare the three algorithms in different K-coverage conditions, we suppose different positive integer values of K, which are 2, 3, and 4 in this paper, and the corresponding sensor nodes number N are, respectively, 32, 48, and 64. The coverage range of each sensor node is fixed and equals to 88 m. To execute these improved PSO algorithms, we considered an initial population of 10 particles, and the value of the parameter m in ICLPSO is chosen to be 10. The termination criterion of every run of each algorithm is the maximum number of iterations, which is equal to 5000



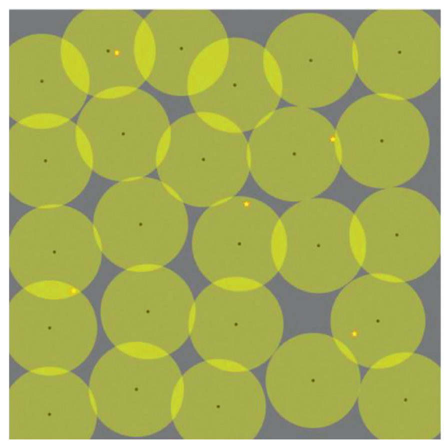
(a)



(b)



(c)



(d)

Figure 4: Locations of sensor nodes optimized by different algorithms. (a) SPSO (0.8366). (b) QPSO (0.8456). (c) CLPSO (0.8240). (d) ICLPSO (0.8660).

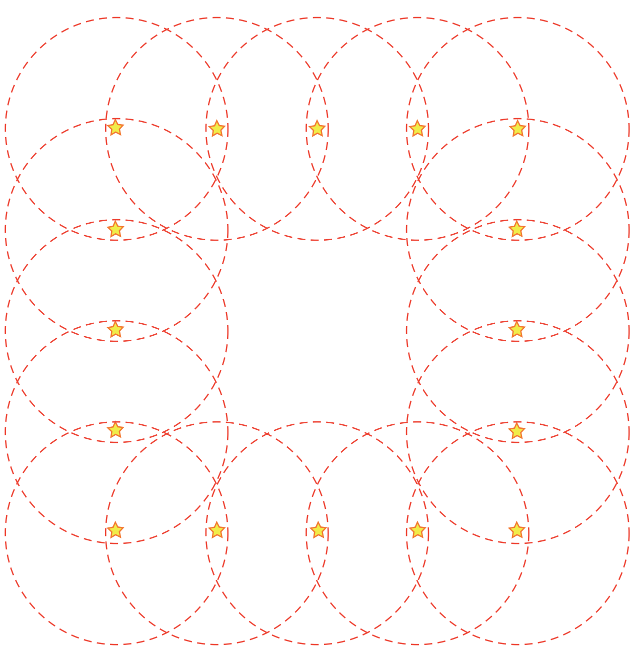


Figure 5: Target points representing the boundary.

Table 3: Simulation parameters.

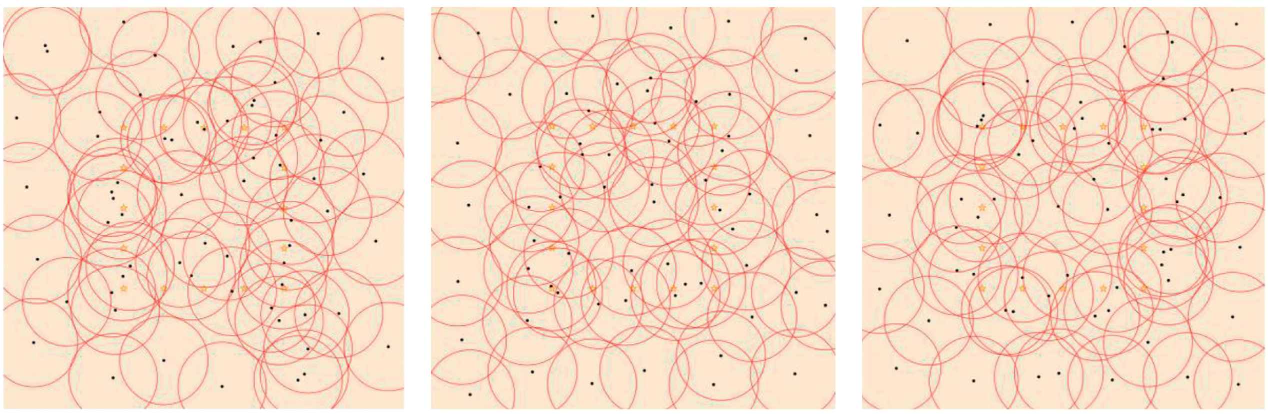
Parameter	Value
Region size	800 m × 800 m
Target points number M	16
Sensing range	88 m
ps	30
c_1	1.494
c_2	1.494
w_0	0.9
w_1	0.4
m	10
$t_{\max}$	5000

in this section. Among the simulations in this part, it is assumed that K-coverage conditions must be fully met for each case. In fact, K-coverage condition is set by the requirement of network lifetime, and evaluation of coverage ratio is meaningful only when all the target points are covered.

Results are reported from an average of 20 runs in each configuration of every algorithm, and they are shown in

Table 4: Statistical results of simulations.

Condition	SPSO		QPSO		ICLPSO	
	C	S	C	S	C	S
$K = 2$	0.8245	0.025	0.8453	0.028	**0.8521**	**0.020**
$K = 3$	0.8891	0.026	**0.9321**	0.030	0.9319	**0.017**
$K = 4$	0.9212	0.023	**0.9687**	**0.016**	0.9457	0.018



(a) (b) (c)

Figure 6: Best locations of sensors calculated by each algorithm ($K = 4$). (a) SPSO (0.9553). (b) QPSO (0.9910). (c) ICLPSO (0.9706).

Table 4. In this table, C stands for the mean value of coverage ratio, and S means the corresponding standard deviation. For the K-coverage problem in this section, ICLPSO has a better performance than SPSO, and it still works best when $K = 2$. However, QPSO conducts best in values of both C and S when $K = 4$. Figure 6 shows sensor deployment of the best result calculated by each algorithm when $K = 4$. It can be seen that QPSO really get a better optimization than the other two.

The success of ICLPSO may be due to the way it creates the global best position. To be concrete, the global best particle is replaced with a particle, in which the value of each dimension is selected based on a tournament selection mechanism, and it adds variation to improve the exploration in solution space. As to QPSO, only the position vector is needed in the evolution equation, which makes the form of the iteration equation easier to control. Therefore, QPSO may behave well in both convergence and accuracy in some situations.

6. Conclusion and Future Works

Sensor deployment problem considering both WSN lifetime constraint and coverage ratio index is studied in this paper. A general sensor model of K-coverage condition is established, and several improved PSO algorithms have been applied in order to get a better optimization effect of deployment. Different cases of sensor deployment problem such as target coverage, area coverage, and boundary coverage are simulated by different algorithms. Many experiments show that ICLPSO proposed in this paper is effective and conducts well in all these conditions, and QPSO also has a better performance than SPSO.

As a future work, we would like to suggest developing these algorithms for solving the K-coverage problem with cost optimization. That is to say, the number of sensor nodes should also be reduced, so that it can achieve better utilization. Moreover, we wish to use these algorithms for mobile targets, which can be an interesting line of research.

Data Availability

This paper designed and tested improved PSO algorithms for sensor deployment without using any data, and there is no data availability need to state.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Wang, Y. Gao, C. Zhou, R. S. Sherratt et al., “Optimal coverage multi-path scheduling scheme with multiple mobile sinks for wsns,” *Computers, Materials and Continua*, vol. 61, no. 3, pp. 695–711, 2020.

[2] X. Shao, L. Wang, J. Li, and J. Liu, “High-order ESO based output feedback dynamic surface control for quadrotors under position constraints and uncertainties,” *Aerospace Science and Technology*, vol. 89, pp. 288–298, 2019.

[3] X. Shao, J. Liu, H. Cao, C. Shen, and H. Wang, “Robust dynamic surface trajectory tracking control for a quadrotor UAV via extended state observer,” *International Journal of Robust and Nonlinear Control*, vol. 28, no. 7, pp. 2700–2719, 2018.

[4] J. Wang, Y. Yang, T. Wang, R. S. Sherratt et al., "Big data service architecture: a survey," *Journal of Internet Technology*, vol. 21, no. 2, pp. 393–405, 2020.

[5] C. Duan, J. Feng, H. Chang, J. Pan, and L. Duan, "Research on sensor network coverage enhancement based on non-cooperative games," *Computers, Materials & Continua*, vol. 60, no. 3, pp. 989–1002, 2019.

[6] 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.

[7] R. M. Curry and J. C. Smith, "A survey of optimization algorithms for wireless sensor network lifetime maximization," *Computers & Industrial Engineering*, vol. 101, pp. 145–166, 2016.

[8] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," *Mobile Networks and Applications*, vol. 10, no. 4, pp. 519–528, 2005.

[9] R. Yarinezhad and S. N. Hashemi, "Increasing the lifetime of sensor networks by a data dissemination model based on a new approximation algorithm," *Ad Hoc Networks*, vol. 100, no. 100, Article ID 102084, 2020.

[10] Y. Gu, H. Liu, and B. Zhao, "Target coverage with QoS requirements in wireless sensor networks," in *Proceedings of the 2007 International Conference on Intelligent Pervasive Computing (IPC 2007)*, pp. 35–38, Jeju Island, Korea, October 2007.

[11] J. Wang, C. Ju, Y. Gao et al., "A pso based energy efficient coverage control algorithm for wireless sensor networks," *Computers, Materials and Continua*, vol. 56, no. 3, pp. 433–446, 2018.

[12] X. Shao and Yi Shi, "Neural adaptive control for MEMS gyroscope with full-state constraints and quantized input," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6444–6454, 2020.

[13] X. Shao, Y. Shi, and W. Zhang, "Input-and-Measurement event-triggered control for flexible air-breathing hypersonic vehicles with asymmetric partial-state constraints," *Nonlinear Dynamics*, 2020.

[14] X. Shao, B. Tian, and W. Yang, "Estimator-based MLP neuroadaptive dynamic surface containment control with prescribed performance for multiple quadrotors," *Aerospace Science and Technology*, vol. 97, Article ID 105620, 2020.

[15] X. Shao, Si Haonan, H. Li et al., "Neurodynamic formation maneuvering control with modified prescribed performances for networked uncertain quadrotors," *IEEE System Journal*, 2020.

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

[17] Y. Zhang, S. Wang, and G. Ji, "A comprehensive survey on particle swarm optimization algorithm and its applications," *Mathematical Problems in Engineering*, vol. 2015, no. 1, 38 pages, Article ID 31256, 2015.

[18] S. Mini, S. K. Udgata, and S. L. Sabat, "Sensor deployment and scheduling for target coverage problem in wireless sensor networks," *IEEE Sensor Journal*, vol. 14, no. 3, pp. 636–644, 2013.

[19] R. Yarinezhad and S. N. Hashemi, "A sensor deployment approach for target coverage problem in wireless sensor networks," *Journal of Ambient Intelligence and Humanized Computing*, 2020.

[20] X. Shao, Si Haonan, and W. Zhang, "Fuzzy wavelet neural control with improved prescribed performance for mems gyroscope subject to input quantization," *Fuzzy Sets Systems*, 2020.

[21] J. Yu, S. Wan, X. Cheng, and D. Yu, "Coverage contribution area based k -coverage for wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 8510–8523, 2017.

[22] L. Zhang, Z. H. Jiao, M. Xu et al., "Coverage enhancement method for 3D wireless sensor network based on particle swarm optimization algorithm," *Journal of Yangtze University*, vol. 17, no. 2, pp. 98–103, 2020.

[23] W. Wang, J. J. Zhu, J. S. Wan et al., "Coverage optimization of wireless sensor networks based on chaotic quantum-behaved particle swarm algorithm," *Chinese Journal of Sensors and Actuators*, vol. 29, no. 2, pp. 290–296, 2016.

[24] M. Mavrovouniotis, C. Li, and S. Yang, "A survey of swarm intelligence for dynamic optimization: algorithms and applications," *Swarm and Evolutionary Computation*, vol. 33, pp. 1–17, 2017.

[25] Y. Shi and R. Eberhart, "A Modified Particle Swarm Optimizer," in *Proceedings of the 1998 IEEE World Congress on Computational Intelligence*, pp. 69–73, New York, NY, USA, June 1998.

[26] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Proceedings of the International Conference on Evolutionary Programming*, pp. 591–600, New York, NY, USA, May1998.

[27] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[28] Y. H. Shi and R. C. Eberhart, "Experimental study of particle swarm optimization," in *Proceedings of the SCI2000 Conference*, Salt Lake City, UT, USA, July 2000.

[29] J. Sun, B. Feng, and W. B. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proceedings of the 2004 Congress on Evolution Computation*, IEEE Press, Piscataway, NJ, USA, pp. 325–331, June 2004.

[30] O. S. Qasim and Z. Y. Algamal, "Feature selection using particle swarm optimization-based logistic regression model," *Chemometrics and Intelligent Laboratory Systems*, vol. 182, pp. 41–46, 2018.

[31] M. Chaudhary and A. K. Pujari, "Q-coverage problem in wireless sensor networks," in *Proceedings of the International Conference on Distributed Computing and Networking*, pp. 325–330, Bangalore, India, January 2009.

[32] Y. L. Wu, Q. He, and T. W. Xu, "Application of improved adaptive particle swarm optimization algorithm in WSN coverage optimization," *Chinese Journal of Sensors and Actuators*, vol. 29, no. 4, pp. 559–565, 2016.

[33] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[34] Y. W. Wang and S. C. Ma, "Multi-swarm comprehensive learning particle swarm optimizer," *Mathematics in Practice and Theory*, vol. 49, no. 10, pp. 273–285, 2019.

[35] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 2002.

[36] W. W. Yu and C. W. Xie, "Hybrid particle optimization with multiply strategies," *Computer Science*, vol. 45, no. 6A, pp. 120–123, 2018.