# Particle Swarm Optimization Algorithm Based on Different Inertia Weights for Solving the P-Hub Allocation Problem

Yu-Xuan Xing, Jie-Sheng Wang *, Yue Zheng, Yu-Cai Wang

*Abstract*—**The p-Hub allocation problem is a classical problem in the location allocation problem, where an optimal network of node allocation paths is formed by placing Hub device locations and assigning each demand node to the corresponding Hub. In this paper, a solution strategy is proposed to solve the p-Hub allocation problem based on the particle swarm optimization (PSO) algorithm with different inertia weights. Firstly, a mathematical model of the p-Hub allocation problem is established. The PSO algorithm is improved by introducing the idea of mutation operator in genetic algorithm (GA) and Metropolis criterion in simulated annealing (SI) algorithm. Then five different inertia weight adjustment strategies (linear decreasing, parabolic, sinusoidal, stochastic and adaptive inertia weights) are adopted in the PSO algorithm. The proposed improved PSO algorithm is used to solve three p-Hub allocation problems, and different inertia weight strategies are compared. Finally, the optimal improved PSO algorithm, cat swarm optimization (CSO) and harmony search (HS) algorithm are selected to optimize the P-Hub allocation problems. Simulation results verify the effectiveness of the improved PSO algorithm.**

*Index Terms*—**Particle swarm optimization algorithm ; p-Hub allocation problem; Inertia weights; Metropolis criterion**

## I. Introduction

The location allocation problem is an important topic in the field of operations research, and it has been widely studied and applied in urban planning, production and military fields. The basic location allocation problems can be divided into three main categories, namely the p-median problem, the p-centre problem and the p-cover problem. The p-median problems explore the selection of the optimal P locations in the set of candidate locations to obtain the objective function with the smallest value, that is, find the P locations with the smallest sum of the product of demand and transmission distance between the demand node and the service node. The difference between the p-median problem and the p-centre problem is that the objective of the p-centre problem is to minimize the maximum distance from each demand node to the nearest service node. The p-cover problems are to explore how to build the minimum number of service stations or spend less under the condition that all demand points are covered [1]. The p-Hub allocation problem is one of the location allocation problems. The p-Hub allocation problem can be divided into two categories according to the objective function: median objective class and center objective class. The Median target class is similar to the p-median problem. Its principle is to select P Hub points and assign all the demand points to the corresponding Hub nodes so as to minimize the overall cost. This target class is generally applicable to the network data communication system [2]. The Center target class model is similar to the p-center problem, which aims to minimize the maximum distance between the demand point and the Hub point. The target class applies to areas, such as transportation of items with shorter shelf life and accelerated mail delivery [3], accelerated mail delivery and other fields. This paper studies a single-allocation p-Hub median problem without capacity constraints, that is, the p-Hub median problem of facilities without capacity constraints and each demand point only connects to one Hub point.

For the p-Hub allocation problem, there are generally the following three constraints. Each Hub node is connected to each other; The discount factor is used to represent the economic cost generated by the connection between Hubs; Except Hub node, there is no direct service between any two nodes [4]. Based on these three prerequisites, scholars have made many in-depth studies in each type of p-Hub allocation problem. In Ref. [5], a heuristic algorithm based on Tabu search and greedy random adaptive search process (GRASP) is proposed for discrete p-hub location problem. Simulation results show that the proposed method can find better solutions in general. Ref. [6] considered the location problem of single distribution hub under demand uncertainty, proposed an alternate convex mixed integer nonlinear programming for variable allocation and a customized solution method based on the cutting plane to solve this computing problem. It solved the capacity constraint and single distribution P-median problem. Ref. [7] carried out the research on a Hub location allocation problem of distribution network. First of all, the mixed integer linear programming (MILP) method is applied, and then four heuristic algorithms

Manuscript received July 23, 2022; revised November 8, 2022. This work was supported by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province (Grant No. LJKZ0293), and the Project by Liaoning Provincial Natural Science Foundation of China (Grant No. 2019-ZD-0031).

Yu-Xuan Xing is a postgraduate student in the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: 805790141@qq.com).

Jie-Sheng Wang is a professor of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (Corresponding author, phone: 86-0412-2538246; fax: 86-0412-2538244; e-mail: wang_jiesheng@126.com).

Yue Zheng is a postgraduate student in the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: 1558867105@qq.com).

Yu-Cai Wang is a doctoral candidate in the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: 1275934857@qq.com).

(genetic algorithm, simulated annealing algorithm, particle swarm optimization algorithm and anti-vibration optimization algorithm) were used to obtain the optimal solution with reasonable execution time, overcome the limit of accurate solution method. Ref. [8] proposed a new mixed integer linear programming (MILP) method for the multiple-allocation P-Hub median value problem based on linear programming and an explicit enumeration algorithm to obtain the exact solution. Ref. [9] put forward a new general variable neighborhood search method, which is used to solve the network capacity limits of the single distribution of p-Hub median problem by using three neighborhood and efficiently updating the data to calculate the new traffic of the network structure. The new nested variable neighborhood descent method was used to solve a very large test instance and show the best performance. Ref. [10] studies the single-allocation p-Hub median problem (USApHMP) with no capacity limitation, and proposes two GAs to solve this NP-hard problem. Numerical experiments on standard ORLIB Hub datasets show robust and efficient solutions for USApHMP with up to 200 nodes and 20 hubs. Ref. [11] proposed a hybrid method to solve the p-hub center problem by locating P hubs from a group of candidate hub locations and assigning demand and supply nodes to hubs and minimizing the maximum travel time in the network, and proposed a hybrid solution to the fuzzy VIKOR modeling hub location problem. Genetic algorithm is used to solve several problem examples successfully. Particle swarm optimization (PSO) algorithm is a random search algorithm based on swarm collaboration developed by simulating the foraging behavior of birds. Based on the observation of animals and swarm activities, the PSO algorithm makes use of the information sharing of individuals to make the movement of the whole swarm produce an evolution process from disorder to order in the problem solving space, so as to obtain the optimal solution. It has been widely used in soft sensor modeling [12], knapsack problem [13], path planning [14], adaptive control [15], multi-objective optimization [16], optimal power flow calculation [17] and other fields because of its easy implementation, high precision and fast convergence.

This paper proposes a method for solving p-Hub allocation problem based on the PSO algorithm with different inertia weights. The structure of the paper is arranged as follows. Section 2 introduces the mathematical model of the p-Hub allocation problem. The third section introduces the working principle and workflow of the basic PSO algorithm. Then the PSO algorithm is improved by introducing other intelligent algorithms and five inertia weight adjustment strategies. Finally, the algorithm coding and parameter setting are expounded. The fourth section is the experimental simulation and results analysis. Finally is the conclusion of the paper.

## II. Mathematical Model of P-Hub Allocation Problem

The p-Hub median problem has the single allocation without capacity constraints, that is to say that the facility is not limited in capacity, and each demand point is only connected to a Hub point. The total cost of this problem is composed of the fixed cost of establishing a Hub and the transportation cost. This kind of mathematical model was first proposed by Kelly in 1987 in the study of route hub location planning of aerospace system, that is, the number of nodes is given, and a certain nodes are selected from them as the Hub point and the cost generated by connection and transmission between Hubs is represented by a discount factor $\alpha$. At the same time, the product of the distance and unit distance between two nodes is given so that the total cost generated between nodes and Hubs and between Hubs and Hubs is the least.

According to the description of p-Hub allocation problem, the following mathematical models can be obtained.

$$\sum_i \sum_j w_{ij} \left( \sum_k x_{ik} c_{ik} + \sum_m x_{jm} c_{jm} + \alpha \sum_k \sum_m x_{ik} x_{jm} c_{km} \right) \tag{1}$$

$$x_{ij} < x_{jj} \text{ for all the } i, j \tag{2}$$

$$\sum_k x_{ik} = 1 \text{ for all the } i \tag{3}$$

$$\sum_N f_i x_i \tag{4}$$

$$x_{ik} \in \{0,1\} \text{ for all the } i, k \tag{5}$$

where, $w_{ij}$ represents the total flow between node $i$ and node $j$, and $c_{ij}$ represents the transportation cost per unit distance between node $i$ and node $j$. When node $i$ is assigned to Hub at node $k$, the value is 1, otherwise 0. $x_{jm}$ is the same. Eq. (1) is used to calculate the sum of the total transportation costs of the model. The purpose of Eq. (2) is to ensure that each demand node is connected to a Hub node. Eq, (3) and (5) ensure that each node can only connect to a Hub node at most. Eq. (4) is the cost of establishing a Hub node. When point $i$ is a Hub node, it is 1, otherwise it is 0, and $f_i$ is the cost of establishing a Hub at different nodes.

## III. Solving p-Hub Allocation Problem Based on PSO Algorithm with Different Inertia Weights

### A. Basic Principle of PSO Algorithm

The PSO algorithm was proposed by Kennedy and Eberhart by studying the behavior pattern of bird collective predation in nature [19]. The basic PSO algorithm is described as follows.

If there are $N$ particles in a $D$ -dimensional space, the $i$ -th particle is represented as a $D$ -dimensional vector, and the position of the $i$ -th particle is represented as:

$$X_i = (X_{i1}, X_{i2} \ldots,\ X_{iD}), i = 1,2 \ldots,\ N \tag{6}$$

The moving velocity of the $i$ -th particle is also expressed as a $D$ -dimensional vector.

$$V_i = (V_{i1}, V_{i2} \ldots, V_{iD}), i = 1,2 \ldots, N \tag{7}$$

The individual optimal value of the $i$ -th particle is named as the individual extremum, which is denoted as:

$$pbest = (p_{i1}, p_{i2} \ldots, p_{iD}), i = 1,2 \ldots, N \tag{8}$$

The optimal value of all particles in the particle swarm is named as the global extremum, which is denoted as:

$$gbest = (p_{g1}, p_{g2} \ldots, p_{gD}) \tag{9}$$

The particle starts updating its speed and position according to $p_{best}$ and $g_{best}$.

$$X_{id}^{k+1} = X_{id}^{k} + V_{id}^{k+1} \quad (11)$$

where, $V_{id}^{k+1}$ represents the $D$ -dimensional velocity of the $i$ -th particle at the ( $k+1$ )-th iteration; The maximum value of $V_{id}^{k+1}$ is $V_{\max}$ $(V_{\max} > 0)$; If $V_{id}^{k+1} > V_{\max}$, $V_{id}^{k+1} = V_{\max}$; $X_{id}^{k+1}$ represents the position of the $i$ -th particle at the ( $k+1$ )-th iteration; $i = 1,2\ldots,N$ represents that there are $N$ particles in the particle swarm; $\omega$ is the inertia weight which controls the particle to adjust its weight according to the current velocity; $rand_1$ and $rand_2$ represent two random numbers between 0 and 1; $c_1$ and $c_2$ are learning factors, and their values determine the weight of particles self-adjustment according to their own experience and social information; $pbest_{id}^{k}$ represents the individual extreme value of the $i$ -th particle found after $k$ iterations; $gbest_{id}^{k}$ represents global extreme of population after $k$ iterations. The procedure of PSO algorithm is shown in Fig. 1.

### *B. Coding and Parameter Design of p-Hub Allocation Problem*

In this paper, the PSO algorithm is used to solve the p-Hub allocation problem based on MATLAB simulation software. The optimal allocation model is obtained by constructing the p-Hub model, coding the algorithm optimization index, setting the particle swarm parameters, debugging and running the program.

(1) Code Strategy

In the spatial domain, there are $N$ nodes waiting for Hub location allocation. Given the $N$ point coordinates, the transfer fees $C$ between any two points, the throughput $r$ between two points, the discount factor $\alpha$ for the connection cost between Hubs, the weights $W_1$ and $W_2$ of circulation cost and establishment cost between hubs, respectively. The Hub in different nodes is set as the cost $f$, and the particle is coded as the matrix of $N \times N$. In the initialization phase, all particles in the particle swarm randomly generate the position matrix of $N \times N$ between 0 and 1, which represents the zero matrix of $N \times N$ of particle velocity. In each particle position matrix diagonal element greater than 0.5, $P_{\max}$ elements with the largest numerical value are selected as Hub nodes. If the number of $P_{\max}$ elements with numerical value greater than 0.5 is less than , all elements with numerical value greater than 0.5 are taken as Hub points. If there is no element with numerical value greater than 0.5 in the diagonal element, the largest element in the diagonal element is taken as Hub point, and the remaining element is taken as the demand node. Determine which Hub node is assigned to each requirement node by comparing the element values of the column where the requirement node is and the row where the Hub node is. The transmission cost between nodes is defined as:

The constraint of Eq. (12) is $x_{ij} < x_{jj}$, $x_{ik}, x_{jl} \in \{0,1\}$ and $(N-P+1)x_{kk} - \sum x_{ik} > 0$. If point $i$ is assigned to Hub at point $k$, $x_{ik}$ is 1, otherwise it is 0. If point $j$ is assigned to Hub at point $l$, $x_{jl}$ is 1, otherwise it is 0. If $k$ is selected as Hub point, $x_{kk}$ is 1, otherwise 0, so as to ensure that each demand node has and only one Hub is connected to it. The total cost of establishing Hub is calculated by:

$$SumXF = \sum_{m} x_m f_m \quad (13)$$

When $m$ is selected as Hub point, $x_m$ is 1, otherwise it is zero. The total cost is calculated based on Eq. (12) and Eq. (13).

$$TotalCost = W_1 \cdot SumOCR + W_2 \cdot SumXF \quad (14)$$

The global optimum is obtained by comparing the $TotalCost$ of all particles. The new $TotalCost$ is obtained by adjusting the position and velocity of the particles at each iteration and the global optimum is updated. The final solution of the PSO algorithm is obtained at the end of the iteration.

$$V_{id}^{k+1} = \omega * V_{id}^{k} + c_1 * rand_1 * \left(pbest_{id}^{k} - X_{id}^{k}\right) + c_2 * rand_2 * \left(gbest_{id}^{k} - X_{id}^{k}\right) \quad (10)$$

$$SumOCR = \sum_{i}\sum_{j} r_{ij} \left( \sum_{k} C_{ik} x_{ik} + \alpha \sum_{k}\sum_{l} C_{kl} x_{ik} x_{jl} + \sum_{l} C_{jl} x_{jl} \right) \quad (12)$$

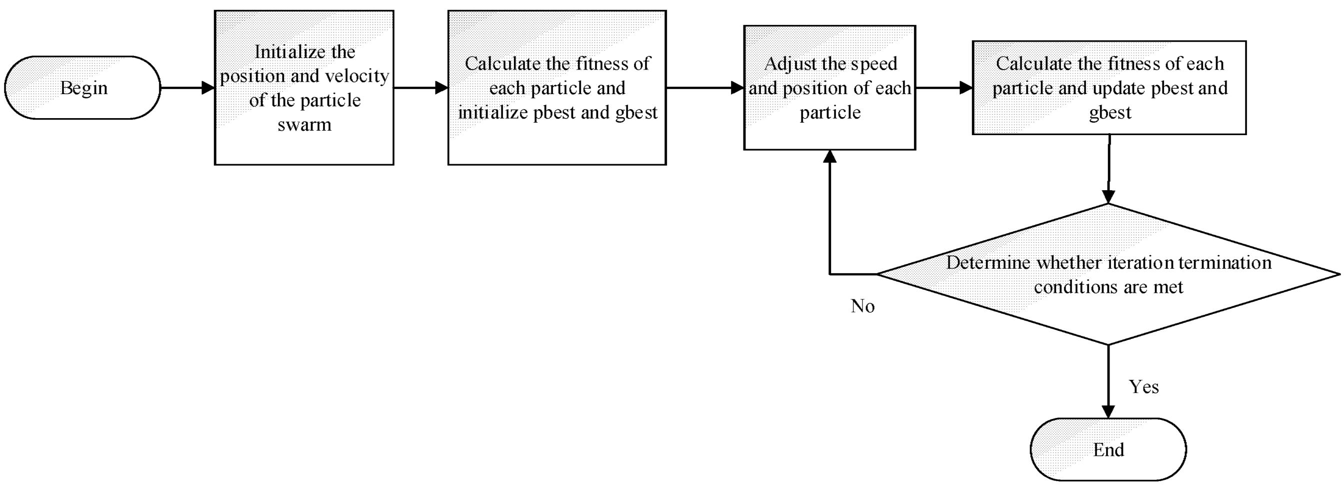



Fig. 1 Procedure of PSO algorithm.

(2) Parameter Setting

1) Population size

The population size is the number of particles in the PSO algorithm. In general, the smaller population size will lead to the algorithm easier to fall into local optimum, and the larger population size can improve the convergence of the algorithm and make the optimization ability of the algorithm stronger, but it will increase the amount of calculation required by the PSO algorithm and reduce its efficiency. This paper selects population size Npop=150.

2) Number of iterations

As the number of iterations increases, the solution of the algorithm becomes more and more stable, and the amount of calculation increases. According to the specific situation studied in this paper, the number of iterations $G_k$ is selected as 250.

3) Inertia weight

The inertia weight $\omega$ determines the proportion of particle flight velocity retention. In this paper, five dynamic adjustment inertia weights are used to optimize the algorithm.

4) Learning factor

$c_1$ and $c_2$ are the learning factors of the PSO algorithm. $c_1$ is the self-learning factor, which is the weight parameter of the learning adjustment part through the individual historical optimization experience. $c_2$ is the social learning factor, which adjusts the weight parameters by learning from the historical experience of the entire particle swarm. $c_1 + c_2 < 4$ and $c_1 = 1.5, c_2 = 2$ in this paper.

*C. Improved Particle Swarm Optimization Algorithm*

(1) Improved PSO Algorithm Based on Metropolis Criterion and Mutation Operator

Based on the basic PSO algorithm, this paper introduces the Metropolis criterion in simulated annealing and the mutation operator in genetic algorithm. Metropolis criterion is an important sampling method proposed by Metropolis. The main idea of Metropolis criterion is to accept the new non-optimal state of the algorithm with certain probability. The mutation operator of GA is to change the gene value of individuals in the population to accelerate the algorithm to converge to the global optimal solution. In this paper, the Metropolis criterion is combined with mutation operator, and the position of each particle after adjustment is mutated in each iteration. Then, according to the Metropolis criterion in the SA algorithm, whether the fitness of the mutated particle is better than that of the particle before mutation is judged. If the mutated particle is better, the particle inherits the position velocity and fitness after mutation. If the particle fitness before mutation is better than that after mutation, the new solution after mutation is accepted with a probability of 10%. At each iteration, the global optimal solution is obtained by comparing the individual historical optimal found by all particles, and then the global optimal is mutated three times again. By comparing the fitness before and after the mutation, the optimal inheritance is selected, and the poor solution obtained by the algorithm is no longer probability accepted when the global optimal is mutated. In this paper, seven different variants are introduced, and one of them is randomly selected at each mutation.

1) Mutation one: exchange two random rows of the matrix.

2) Mutation two: exchange two random columns of the matrix.

3) Mutation three: exchange any two elements of the matrix.

4) Mutation four: exchange any two elements on the diagonal of the matrix.

5) Mutation five: randomly select two elements on the diagonal of the matrix and reverse the diagonal intercepted by the two elements.

6) Mutation six: randomly select an element whose value is greater than 0.5 in the diagonal of the matrix and halve it.

7) Mutation seven: randomly select an element whose value is less than 0.5 in the diagonal of a matrix and double its value.

The above seven methods can change the distribution of the selected Hub location node and the demand node in different ways. By combining Metropolis criterion with mutation operator, it can jump out when the PSO algorithm falls into local optimum and continue to find global optimum.

(2) Improved PSO Algorithm Based on Different Inertia Weights

Although the PSO algorithm has the characteristics of less parameters, simple implementation and high operation efficiency, it is easy to fall into local optimum and cannot find the global optimum due to the premature convergence. Therefore, many scholars have conducted in-depth discussions on how to balance the local and global optimization performance of the algorithm. It is found that the strategy change of inertia weight can effectively solve this problem. In the PSO algorithm, the inertia weight is one of the key factors affecting the particle flight speed maintenance. When $\omega$ increases, the global search ability of particles will increase, and when $\omega$ decreases, the local search ability of particles will increase. This paper adopts the five strategies of inertia weight.

1) Linear decreasing strategy

The decreasing inertia weight with the increase of iteration times can enhance the self-learning ability of the particles in the early iteration. While the iteration enters the later stage, $\omega$ becomes smaller and smaller, and the social learning ability of the particles will gradually be enhanced, which makes the local search optimization ability and global search optimization ability of PSO algorithm be balanced to a certain extent. In a linear decreasing inertia weight, set its initial $\omega_{\max} = 1$, $\omega_{\min} = 0.2$ at the end of iteration, $k$ is the number of current iterations, and the maximum number of iterations is $G_k$.

$$\omega^k = \omega_{\max} - \frac{k(\omega_{\max} - \omega_{\min})}{G_k} \tag{15}$$

2) Parabolic strategy

The linear decreasing inertia weight is relatively easy to realize. Some scholars have proposed a nonlinear decreasing strategy based on the linear decreasing strategy. Chatterjee and Siarry have proposed a $\omega$ adjusting strategy to solve practical problems, and this inertia weight is a form of curve[21]. Its mathematical expression can be described as:

$$\omega_k = (\omega_{\max} - \omega_{\min})\left(1 - \frac{k}{G_k}\right)^n + \omega_{\min} \quad (16)$$

where, $n$ is set to 2, $\omega_{\max}$ is set to 1, and $\omega_{\min}$ is set to 0.2. When $n$ is set to 2, $\omega$ decreases with iteration in the form of parabola.

3) Sinusoidal function strategy

In Ref. [22], the inertia weight by using sine function is introduced. When this strategy is adopted, the $\omega$ value increases first and then decreases. The particle first searches locally based on its own experience, and then searches globally according to social learning, and finally searches locally. In this paper, the initial inertia weight $\omega_{\text{beg}}$ is set to 0.3.

$$\omega^k = \omega_{beg} + 0.6 * \sin\left(\frac{\pi * k}{G_k}\right) \quad (17)$$

4) Random inertia weight

The random inertia weight is based on a certain range of random values at the end of each iteration. This strategy can avoid the weak local search ability to a certain extent in the early iteration, and can make up for the poor global search ability in the later iteration. A random inertia weight for tracking dynamic systems proposed in Ref. [23] is shown in Eq. (18).

$$\omega = 0.5 + \frac{rand()}{2} \quad (18)$$

5) Adaptive inertia weight

The adaptive inertia weight connects the inertia weight with the performance of the algorithm, and adjusts the inertia weight in real time through the feedback information provided by the PSO algorithm. Due to the characteristics of its intelligent control, this strategy can be well applied in complex environments. At the same time, it can avoid the premature convergence of the algorithm and accelerate the late optimization efficiency. An adaptive inertia weight was proposed with the parameter optimization of adaptive fuzzy inference system. On this basis, the mathematical expression adjusted according to the actual situation is defined as follows.

$$\omega = \begin{cases} \omega_{\max} - \frac{(\omega_{\max} - \omega_{\min})(f - f_{\min})}{(f_{avg} - f_{\min})} & f < f_{avg} \\ \omega_{\max} & f \geq f_{avg} \end{cases} \quad (19)$$

where, $\omega_{\max} = 0.8$ , $\omega_{\min} = 0.3$ , $f$ is the objective function value found by particles, $f_{\min}$ is the optimal value of the objective function found by all particles in the iteration, $f_{avg}$ is the average objective function value of all particles in the previous iteration, and $f_{avg}$ is initialized before the first iteration. This strategy strengthens the global search ability when the objective function value of particles is higher than all averages. When the target value of particles is lower than the average value of all particles, the local optimization ability is adjusted according to this value. The smaller the target value, the greater the improvement of the local search ability.

## IV. Simulation Experiments and Result Analysis

### A. *Description of Simulation Cases*

In the first three sections of this paper, the research background and current situation of p-Hub allocation problem are expounded, and the mathematical model is established. The basic PSO algorithm is improved and their basic parameters are given. Based on the previous three sections, this section uses MATLAB simulation software for program debugging, optimizes three different P-Hub cases, and selects the best improved PSO algorithm to be compared with the simulation results of other classical algorithms to verify the feasibility of the improved PSO algorithm. This paper selects the following three p-Hub allocation models for carrying out the simulation experiments.

(1) Model one. The location problem with 20 nodes is selected as shown in Fig. 2, and the up to 3 points are selected as Hub points, and the non-Hub points are assigned to Hub points. Give the unit transmission cost $C$ between any two points, the transmission amount $r$ between any two points, the discount factor $\alpha$ of the cost generated by the connection between Hubs, and the cost $f$ required for the establishment of Hubs in different nodes.

(2) Model two. The location problem with 30 nodes is selected as shown in Fig. 3, and the up to 6 points are selected as Hub points, and the non-Hub points are assigned to Hub points. Give the unit transmission cost $C$ between any two points, the transmission amount $r$ between any two points, the discount factor $\alpha$ of the cost generated by the connection between Hubs, and the cost $f$ required for the establishment of Hubs in different nodes.

(3) Model three. The location problem with 40 nodes is selected as shown in Fig. 4, and the up to 7 points are selected as Hub points, and the non-Hub points are assigned to Hub points. Give the unit transmission cost $C$ between any two points, the transmission amount $r$ between any two points, the discount factor $\alpha$ of the cost generated by the connection between Hubs, and the cost $f$ required for the establishment of Hubs in different nodes.

In view of the above three models, according to the algorithm parameters and the five inertia weight strategies, the basic PSO algorithm are simulated ten times respectively. The optimal values are recorded, and the mean and variance are calculated. The distribution diagram of the optimal solution, the curve of different inertia weight with iteration and the curve of global optimal solution with different inertia weight with iteration times are obtained.

### B. *Simulation Results and Analysis*

(1) Simulation Results and Analysis (Model One)

The simulation results of model one is shown in Table 1. The convergence curves under different PSO algorithms of model one are shown in Fig. 5. The variable curves of different inertia weights of model one with the number of iterations are shown in Fig. 6. For model one, the optimal solution is obtained after ten simulations with the each inertia weight and the corresponding optimal Hub location allocation is shown in Fig. 7. Based on the analysis of the simulation results listed in Table 1, the parabolic strategy, adaptive strategy and random strategy all have smaller optimal solutions than other strategies, and the mean value of these three strategies is significantly less than that of other strategies. It can be seen from the standard deviation of each strategy that the stability of the adaptive inertia weight strategy is the strongest. The comprehensive analysis of the data shows that other strategies except sin-type strategy have achieved better results than the basic PSO algorithm. Analysis on the curves shown in Fig. 5 shows that the early convergence speed of basic PSO algorithm is too fast and the late optimization ability is insufficient, while the early convergence speed of other strategies is lower than that of basic PSO, and it still has good optimization effect in the middle of iteration.

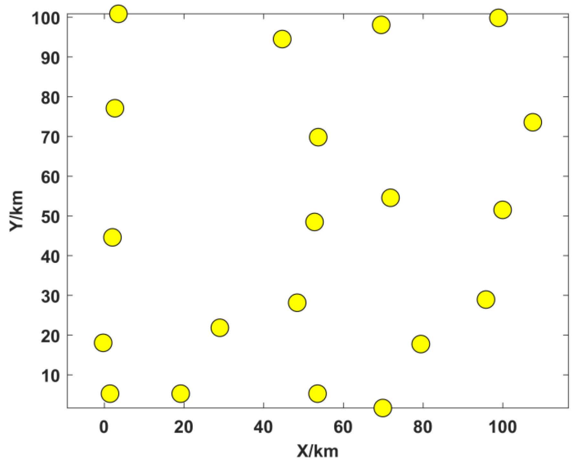



Fig. 2 Distribution map of nodes to be located (model one).

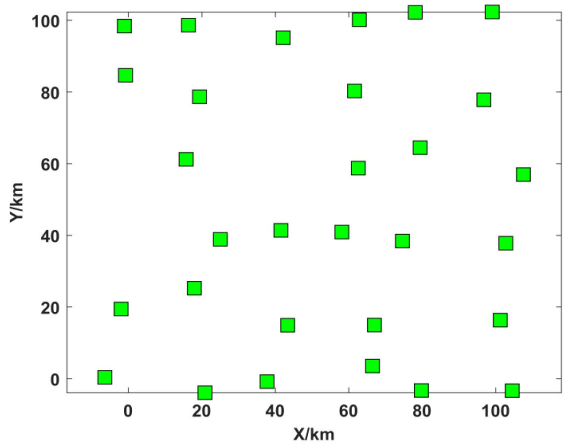



Fig. 3 Distribution map of nodes to be located (model two).

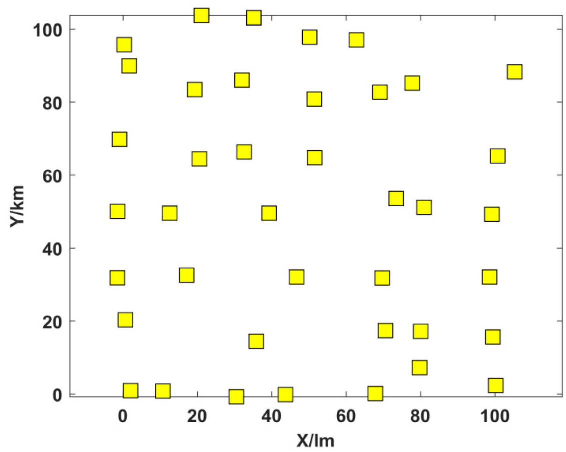



Fig. 4 Distribution map of nodes to be located (model three).

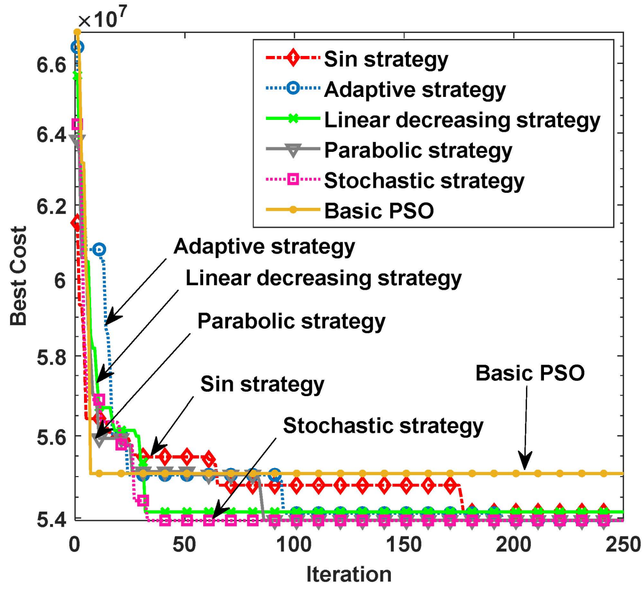
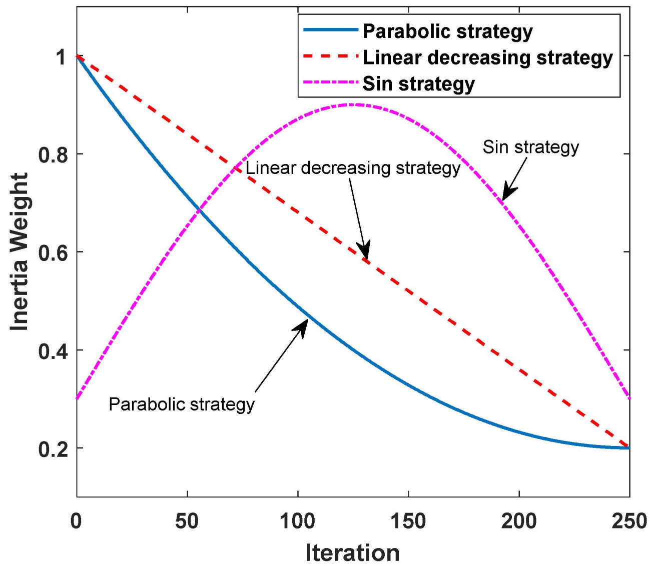
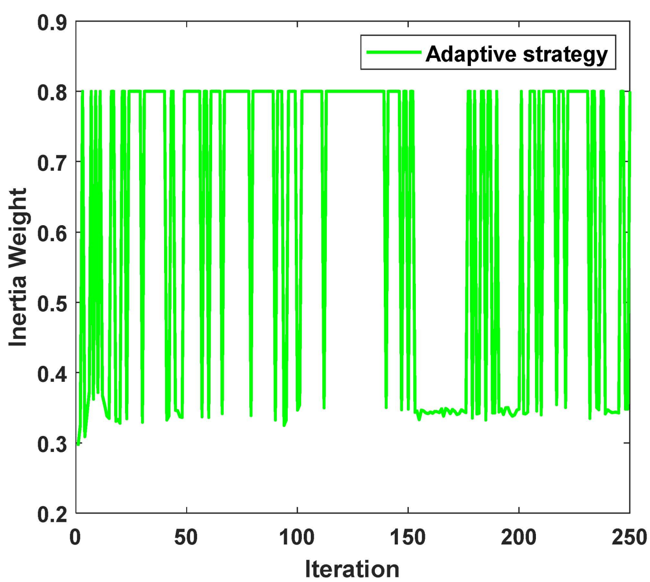


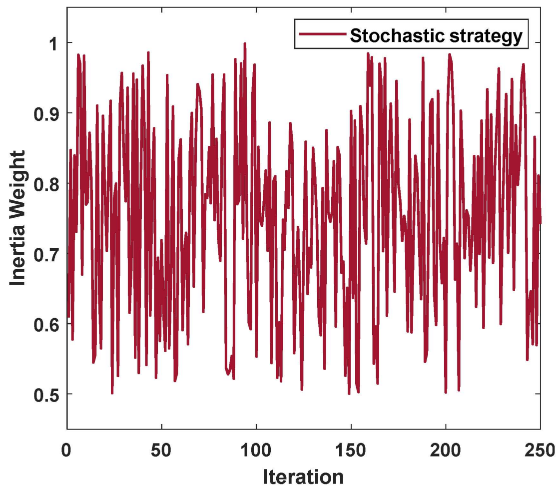

Fig. 5 Convergence curves under different algorithms (model one).




(a)

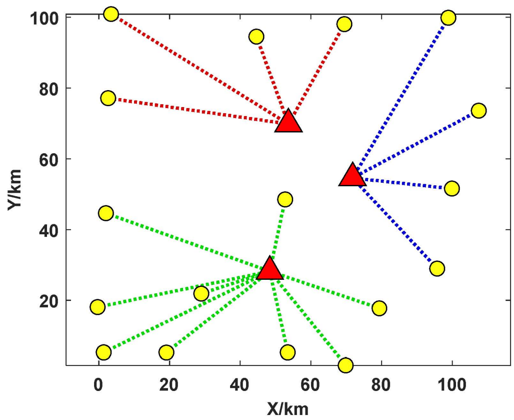



(b)

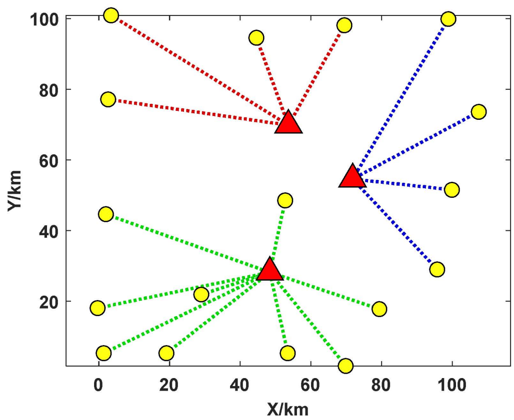



(c)

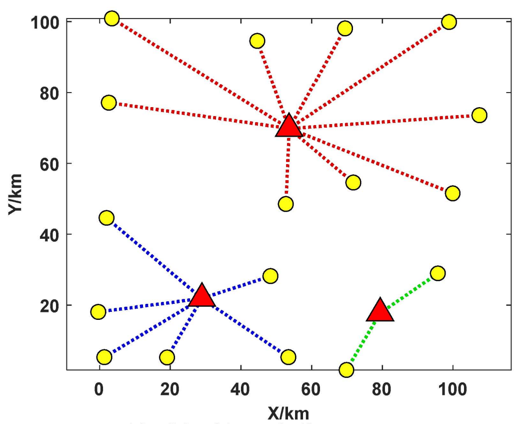
Fig. 6 Variable curves of different inertia weights (model one).



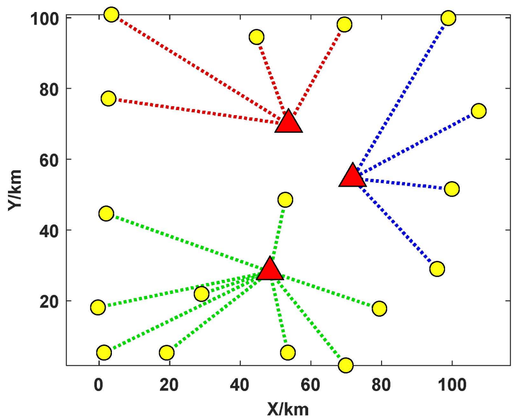

(a) Basic PSO



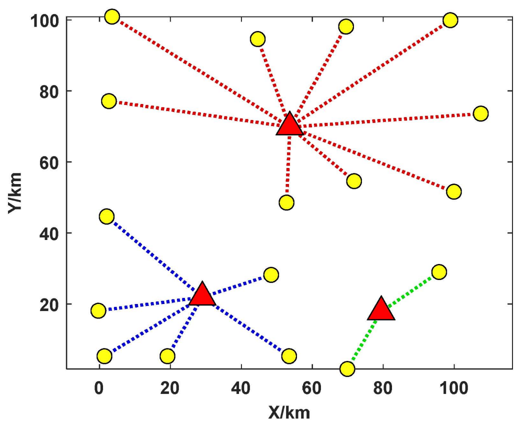

(b) PSO with linear decreasing strategy



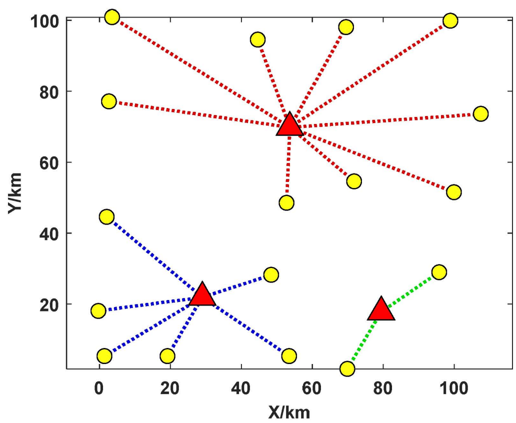

(c) PSO with parabolic strategy




(d) PSO with sin strategy




(e) PSO with stochastic strategy




(f) PSO with adaptive strategy

Fig. 7 Distribution map (model one).

TABLE 1. SIMULATION RESULTS UNDER DIFFERENT METHODS (MODEL ONE)

| Algorithm | Parameters | Optimal Solution | Mean value | Standard deviation |
|---|---|---|---|---|
| Basic PSO | $N = 20, P_{\max} = 3$ | 54143520 | 54766472.9 | 443167.94 |
| PSO with linear decreasing $\omega$ | $N = 20, P_{\max} = 3$ | 54143520 | 54678937.15 | 435383.58 |
| PSO with parabolic $\omega$ | $N = 20, P_{\max} = 3$ | 53942415 | 54552884.35 | 485848.46 |
| PSO with sin $\omega$ | $N = 20, P_{\max} = 3$ | 54143520 | 54823762.15 | 410201.02 |
| PSO with stochastic $\omega$ | $N = 20, P_{\max} = 3$ | 53942415 | 54514794.4 | 489315.41 |
| PSO with adaptive $\omega$ | $N = 20, P_{\max} = 3$ | 53942415 | 54584234.9 | 409722.99 |

(2) Simulation Results and Analysis (Model Two)

The simulation results of model two are shown in Table 2. The convergence curves under different PSO algorithms of model two are shown in Fig. 8. The variable curves of different inertia weights of model two with the number of iterations are shown in Fig. 9. For model two, the optimal solution is obtained after ten simulations with the each inertia weight and the corresponding optimal Hub location allocation is shown in Fig. 10.

Through the analysis of the ten simulation results of model two shown in Table 2, it can be seen that the parabolic inertia weight, the linear decreasing inertia weight and the adaptive inertia weight have smaller optimal solutions than other strategies. In addition, on the data of the mean value of simulation solutions, these three strategies are also significantly lower than other strategies. It can be seen that the use of these three strategies for model two has good results.

By observing the standard deviation of the simulation solution, it can be seen that the adaptive inertia weight still has strong stability for the solution of model two, but the parabolic strategy has a large standard deviation. By analyzing the change rule of this strategy and the curves in Fig. 8, the linear graph with the increase of the number of iterations shows the shape of a downward interval of an open upward parabola, and the decrease speed is relatively fast at the beginning of the iteration.

This situation may lead to the insufficient search of the particles in the spatial domain, resulting in a large standard deviation of the solution. By comparing the data, it can be concluded that the simulation results of these five strategies for model two are better than those of the basic PSO algorithm, but the effect of sin-type inertia weight strategy is not obvious.

(3) Simulation Results and Analysis (Model Three)

The simulation results of model three is shown in Table 3. The convergence curves under different PSO algorithms of model three are shown in Fig. 11. The variable curves of different inertia weights of model three with the number of iterations are shown in Fig. 12.

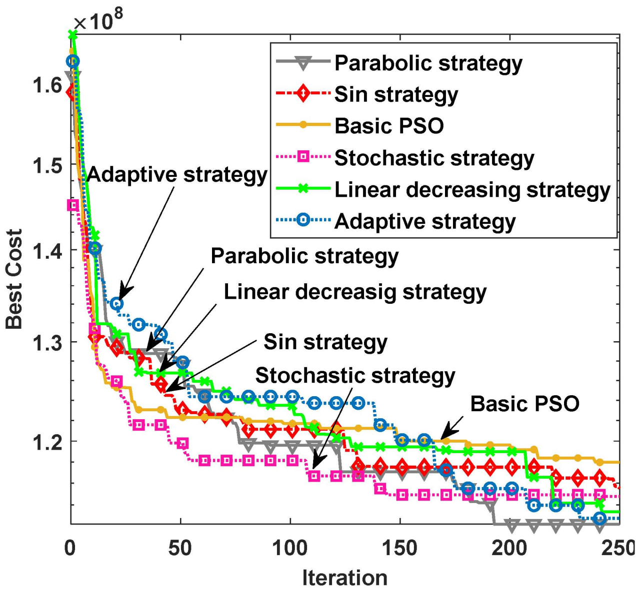



Fig. 8 Convergence curves under different algorithms (model two).

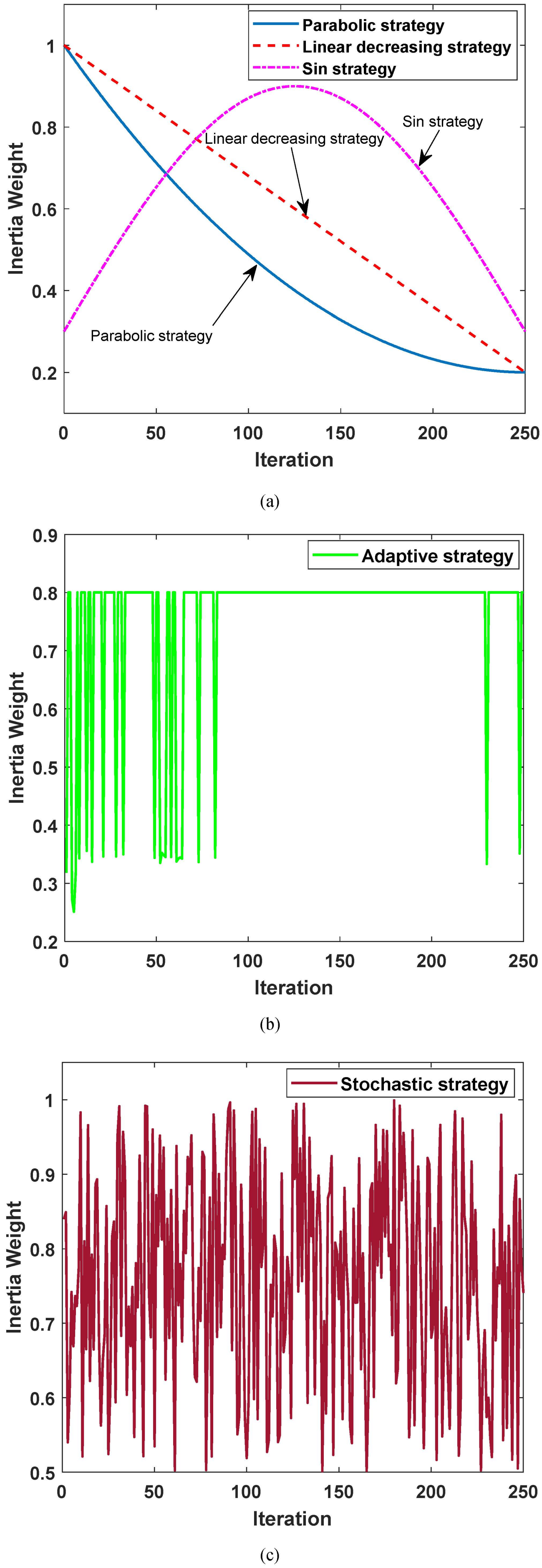
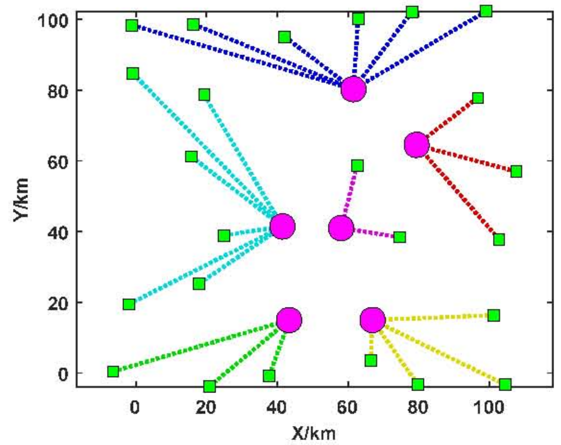
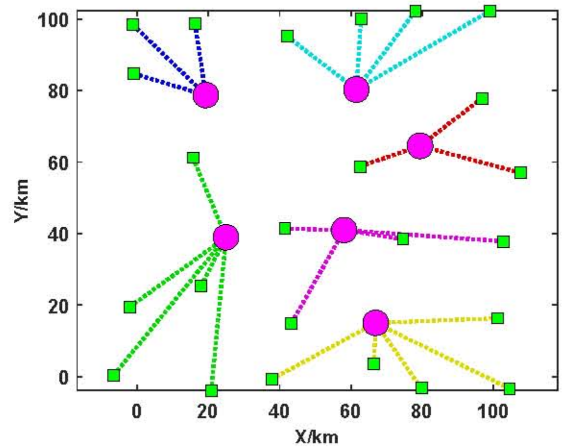


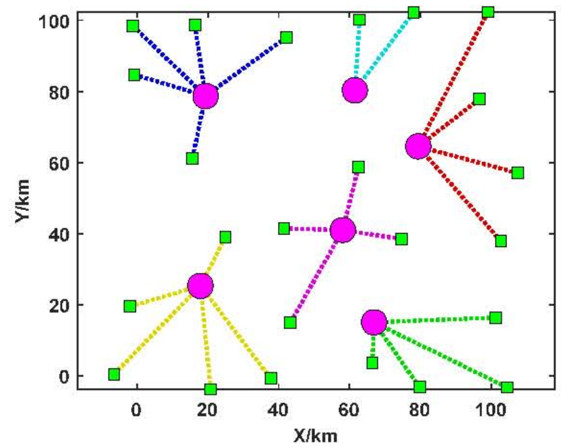

Fig. 9 Variable curves of different inertia weights (model two).



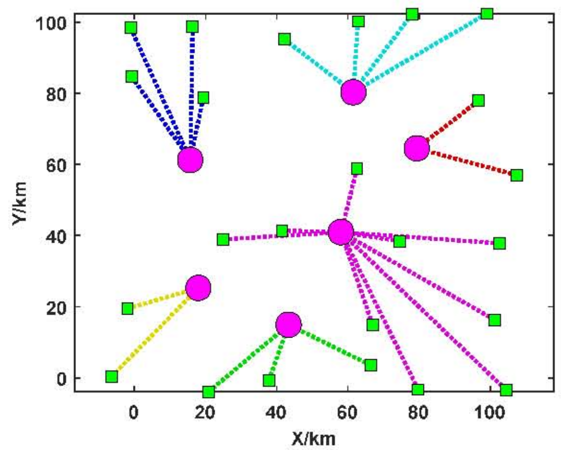
(a) Basic PSO



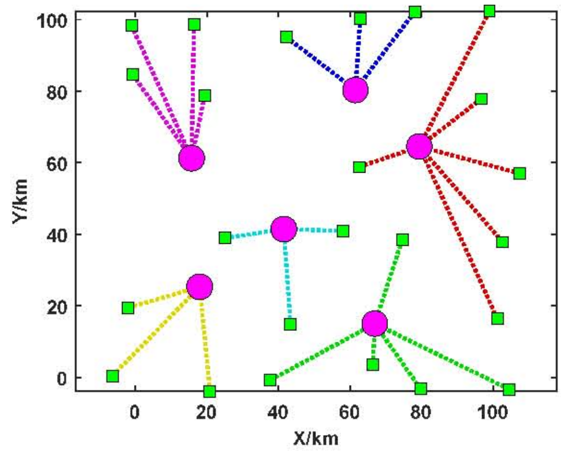
(b) PSO with linear decreasing strategy



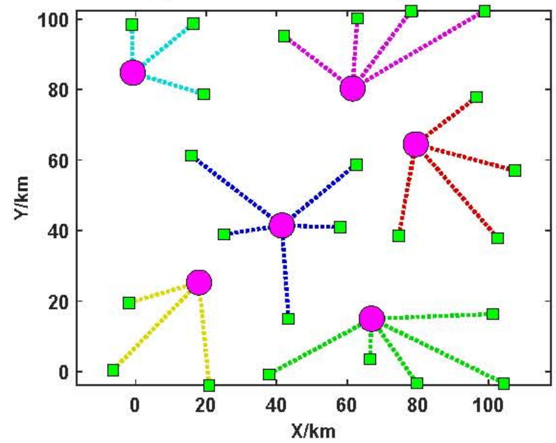
(c) PSO with parabolic strategy



(d) PSO with sin strategy



(e) PSO with stochastic strategy



(f) PSO with adaptive strategy

Fig. 10 Distribution map (model two).

TABLE 2. SIMULATION RESULTS UNDER DIFFERENT METHODS (MODEL TWO)

| Algorithm | Parameters | Optimal Solution | Mean value | Standard deviation |
|---|---|---|---|---|
| Basic PSO | $N = 30, P_{max} = 6$ | 1179631805 | 120150154.2 | 1713922.63 |
| PSO with linear decreasing $\omega$ | $N = 30, P_{max} = 6$ | 113360101 | 116592065.8 | 2129943.31 |
| PSO with parabolic $\omega$ | $N = 30, P_{max} = 6$ | 112233504 | 117999123.9 | 3967385.85 |
| PSO with sin $\omega$ | $N = 30, P_{max} = 6$ | 115519314 | 119710333.2 | 3298107.94 |
| PSO with stochastic $\omega$ | $N = 30, P_{max} = 6$ | 114767260 | 118053344.5 | 2900706.17 |
| PSO with adaptive $\omega$ | $N = 30, P_{max} = 6$ | 112761326.5 | 116779565.4 | 2690613.52 |

For model three, the optimal solution is obtained after ten simulations with the each inertia weight and the corresponding optimal Hub location allocation is shown in Fig. 13. Through the analysis of the ten simulation results of the model three shown in Table 3, it can be seen that the parabolic inertia weight obtains the minimum optimal value. In the mean value of simulation solution, the parabolic inertia weight and adaptive inertia weight are significantly better than other strategies, and the other three strategies are also better than the solution of the basic PSO algorithm.

Compared with the standard deviation of the simulation data, it can be seen that the adaptive strategy still has strong stability. It can be seen from Fig. 11 that with the increase of the solution model, the particle convergence rate with each strategy decreases in the early stage, and the late optimization ability of the basic PSO algorithm is not strong. The optimization ability of the stochastic inertia weight for model three is lower than that of model one and model two. The analysis of the change rule of this strategy may be the result of the increase of the solving model, which cannot make the particles fully search in the early iteration and cannot make all the particles accelerate the convergence in the late iteration.

### *C. Analysis and Summary of Inertia Weight Strategy*

(1) Linear decreasing strategy

In the early iteration, the linear decreasing inertia weight strategy has good global search performance, and also has good local optimization performance in the late iteration. Observing the line graph of the simulation, it can also be found that the particle swarm can make full global search in the early stage so as to fully search the spatial domain, and improve the convergence speed in the later stage to search for the optimal solution near the global optimum. The simulation results of three groups of models show that the optimization effect of this strategy on the basic PSO algorithm is obvious.

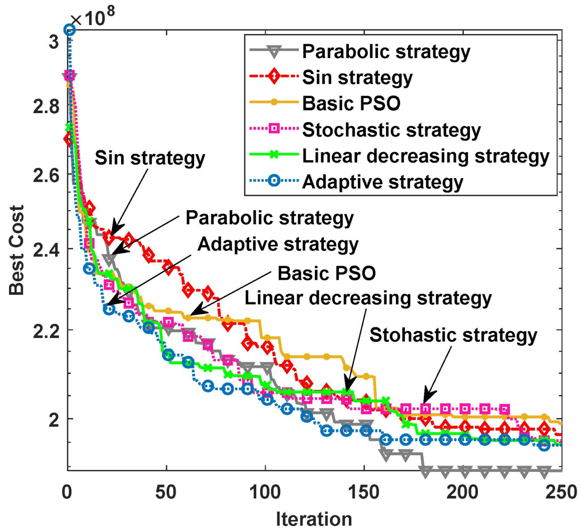
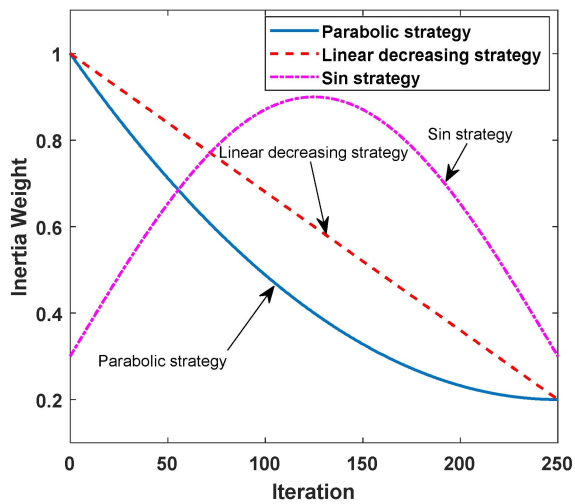
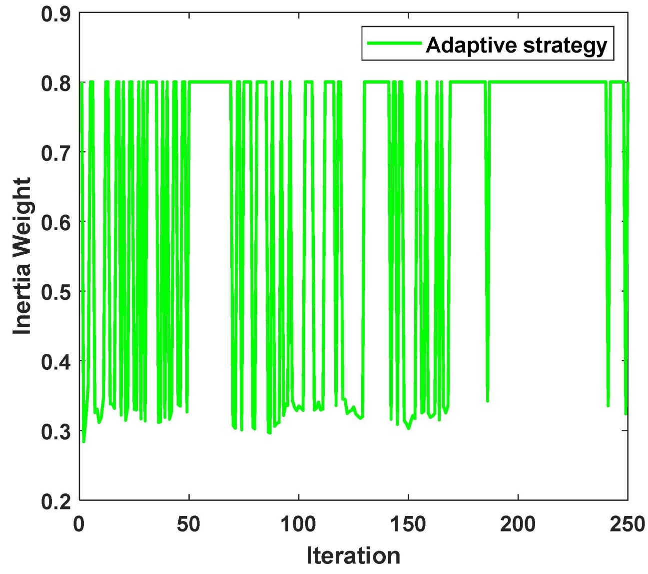


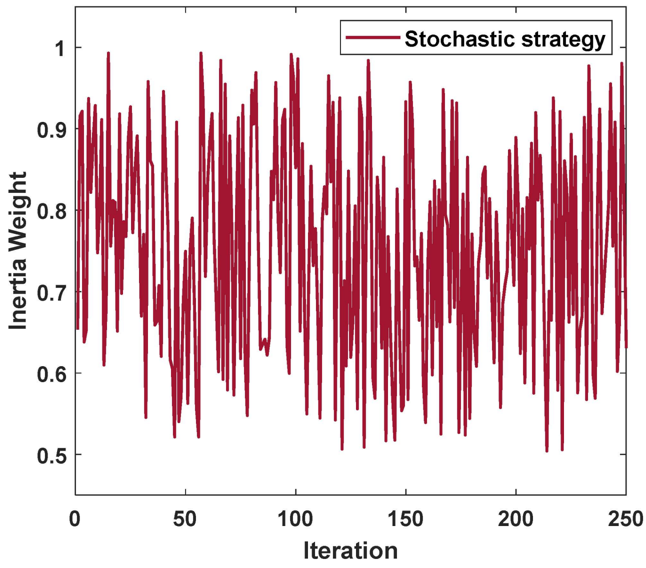

Fig. 11 Convergence curves under different algorithms (model three).




(a)

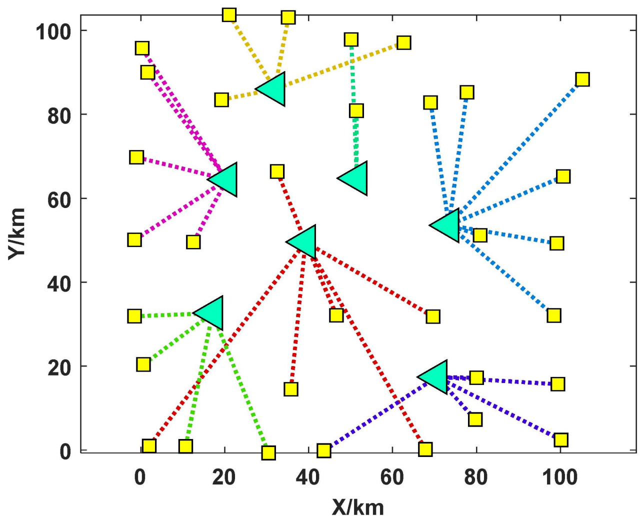



(b)

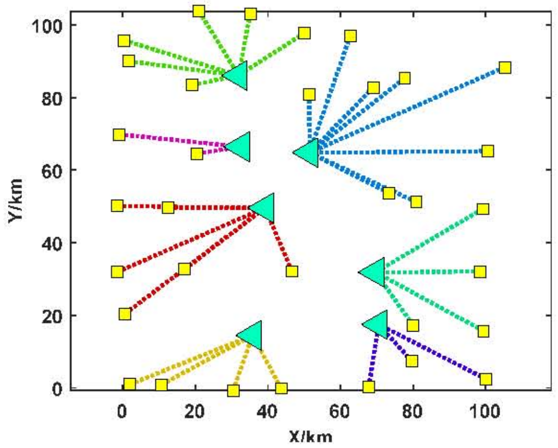



(c)

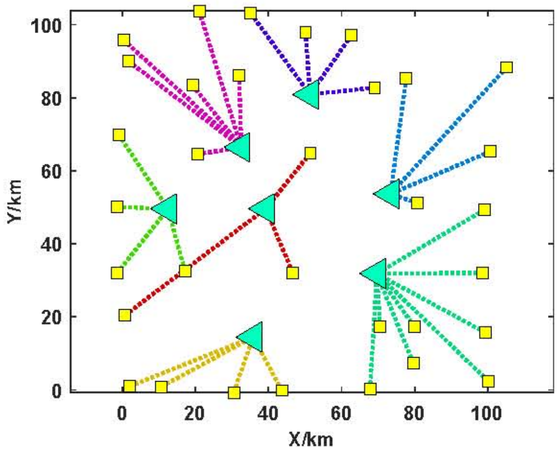
Fig. 12 Variable curves of different inertia weights (model three).



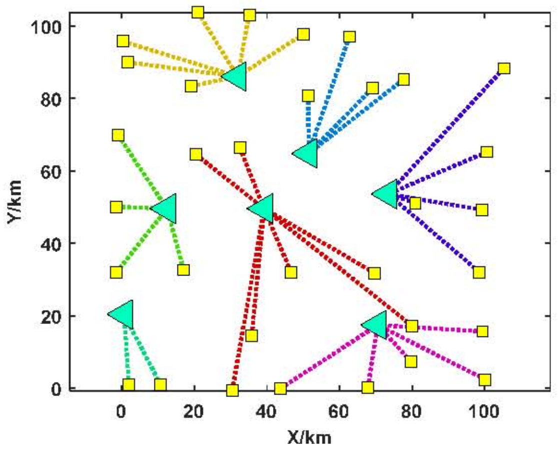

(a) Basic PSO



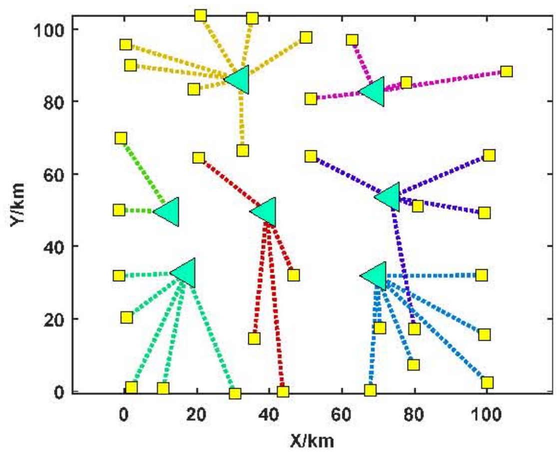

(b) PSO with linear decreasing strategy



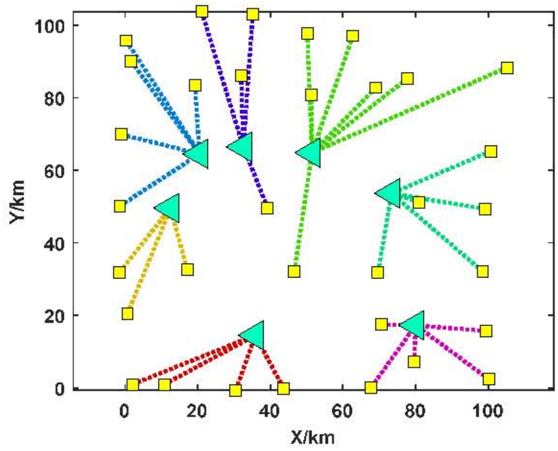

(c) PSO with parabolic strategy




(d) PSO with sin strategy




(e) PSO with stochastic strategy




(f) PSO with adaptive strategy

Fig. 13 Distribution map (model three).

TABLE 3. SIMULATION RESULTS UNDER DIFFERENT METHODS (MODEL THREE)

| Algorithm | Parameters | Optimal Solution | Mean value | Standard deviation |
|---|---|---|---|---|
| Basic PSO | $N = 40, P_{max} = 7$ | 198918107 | 207876583 | 6535794.30 |
| PSO with linear decreasing $\omega$ | $N = 40, P_{max} = 7$ | 194518440 | 202328519 | 8264634.13 |
| PSO with parabolic $\omega$ | $N = 40, P_{max} = 7$ | 189168516 | 199648731.6 | 6906347.39 |
| PSO with sin $\omega$ | $N = 40, P_{max} = 7$ | 196639248 | 204819431 | 6130585.60 |
| PSO with stochastic $\omega$ | $N = 40, P_{max} = 7$ | 195330954 | 203766826.3 | 5523963.14 |
| PSO with adaptive $\omega$ | $N = 40, P_{max} = 7$ | 194393330.5 | 199757698.5 | 3409794.72 |

(2) Parabolic strategy

The parabolic inertia weight strategy is similar to the linear decreasing strategy. The comparison of the relationship between two inertia weight strategies and the number of iterations is shown in Fig. 14. The parabolic strategy reduces faster in the early stage, resulting in faster convergence of particles to the vicinity of the global optimum. Through the comparison of three groups of simulation data, the simulation effect of this strategy is better than that of the linear decreasing strategy. However, due to the fast convergence speed in the early stage, the search in the spatial domain may be insufficient, resulting in large fluctuation of the solution value.

(3) Sin strategy

Based on the analysis of the simulation results of the three groups of models, this strategy has a certain effect on the optimization of the basic particle swarm, but the effect is not obvious, and there is no advantage over other strategies. By analyzing the variation curves of the value with iteration shown in Fig. 15, it can be seen that the particle swarm is doing local search before and after the iteration, and this optimization mode starts to conduct global search in the medium term so as to make the particle swarm converge, which leads to poor local optimization effect in the large range in the early stage of the particle swarm, and thus the overall optimization effect is not ideal.

(4) Stochastic strategy

Random inertia weight randomly selects a value greater than 0.5 and less than 1 at the end of each iteration, which can consider the global search and the local search. The comprehensive analysis of three groups of simulation results shows that it has better optimization effect, but with the increase of the solution model, the optimization effect decreases.

(5) Adaptive strategy

Comprehensive analysis of above three groups of simulation results shows that the adaptive inertia weight has obvious advantages over other strategies. By comparing the standard deviation of solution, it can also be seen that this strategy has strong solution stability. This strategy is to adjust the inertia weight of each particle in the particle swarm according to the particle feedback information. This mechanism is more targeted, more intelligent and better overall effect.

*D. Simulation comparison between improved PSO algorithm and other classical algorithms*

In order to verify the feasibility and efficiency of the improved particle swarm optimization algorithm, this section selects the adaptive inertia weight PSO with the best optimization effect mentioned above, and compares the simulation results of the strategy PSO with the simulation results of the two classical algorithms cat swarm algorithm and harmony search algorithm. The simulation model uses the above three models.

Each algorithm performs ten simulations, and the obtained simulation data is shown in Table 4. The simulation results of different algorithms with the iterative comparison curve are shown in Fig. 16, and the location allocation results of the optimal solutions of the two selected classical algorithms are shown in Fig. 17. Through the comparison and analysis of the data in Table 4, it can be seen that the PSO with adaptive improvement strategy has obvious advantages over the two classical algorithms in the optimal solution, average value and standard deviation of 10 simulations.

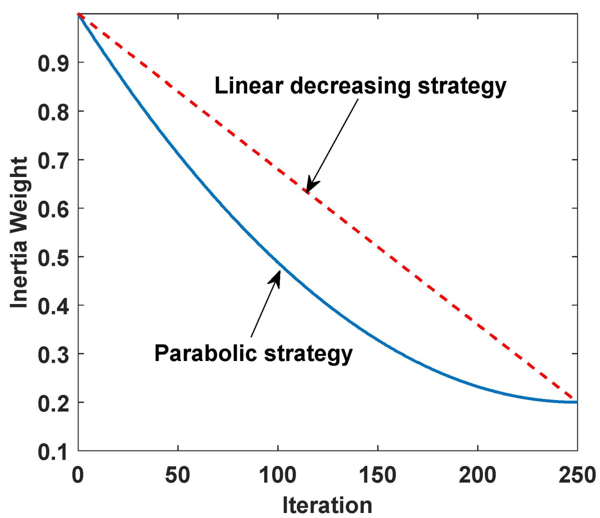



Fig. 14 Relation between inertia weights and iterative number.

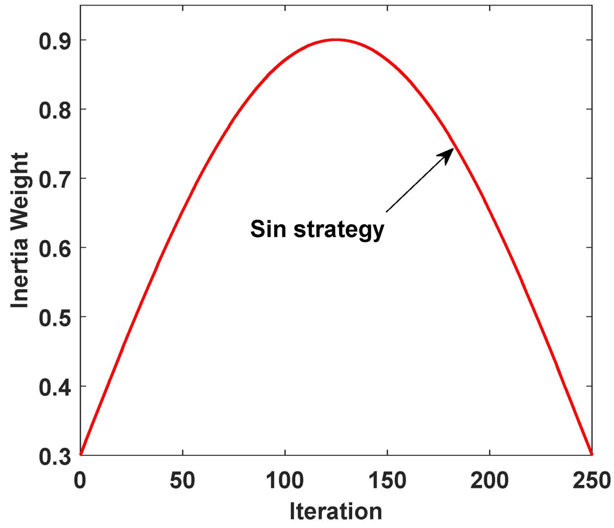
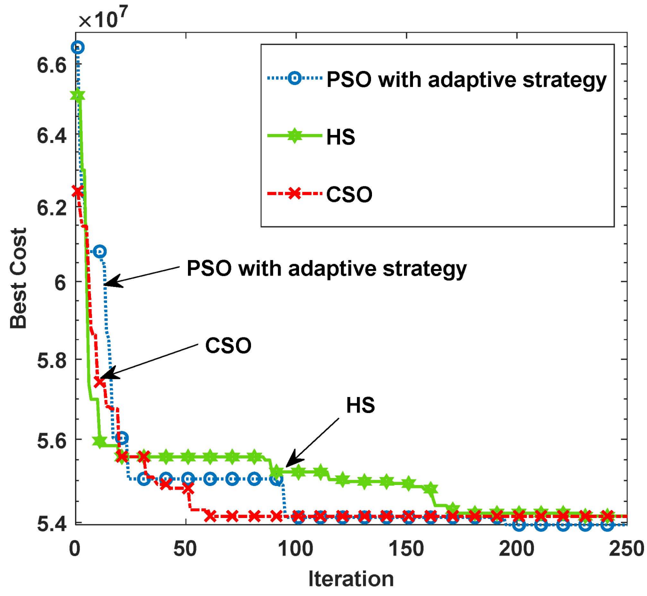
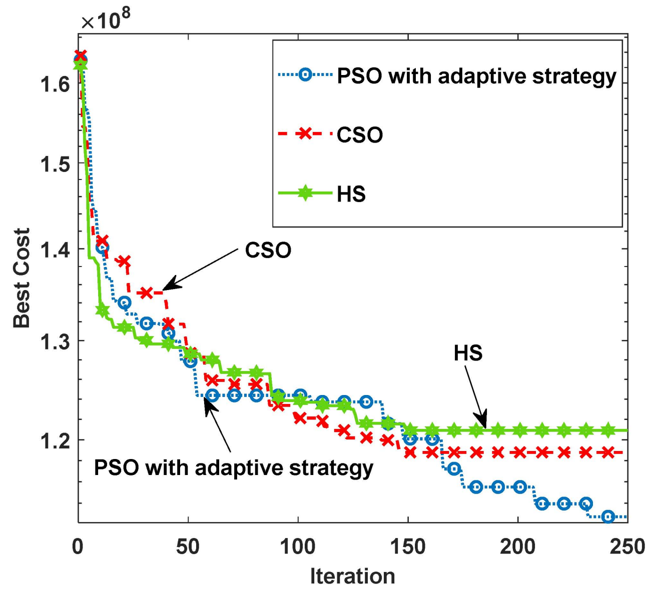


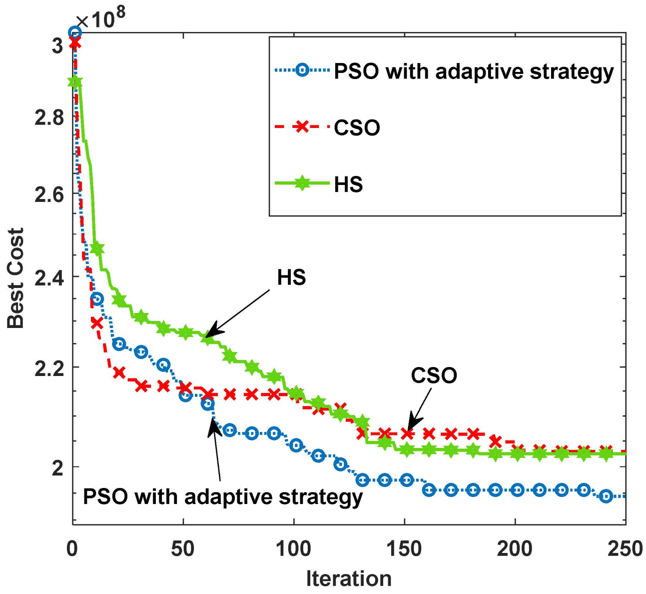

Fig. 15 Relation between sin inertia weight and iterative number.




(a) Model one

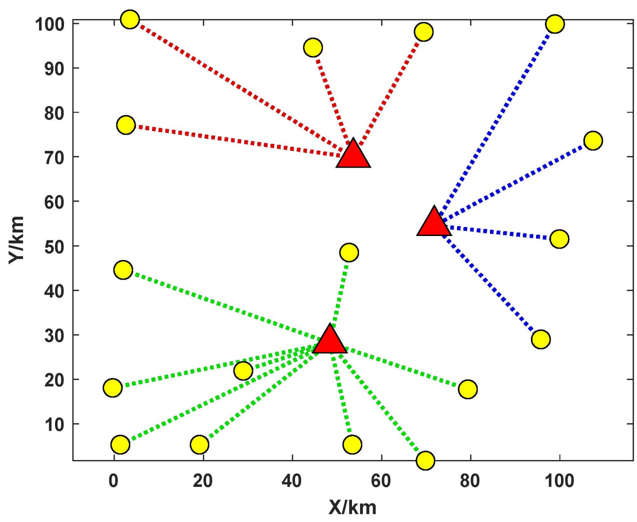



(b) Model two

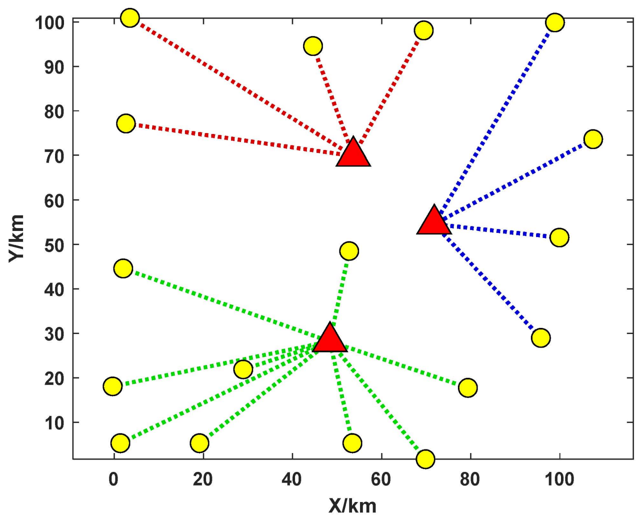



(c) Model three

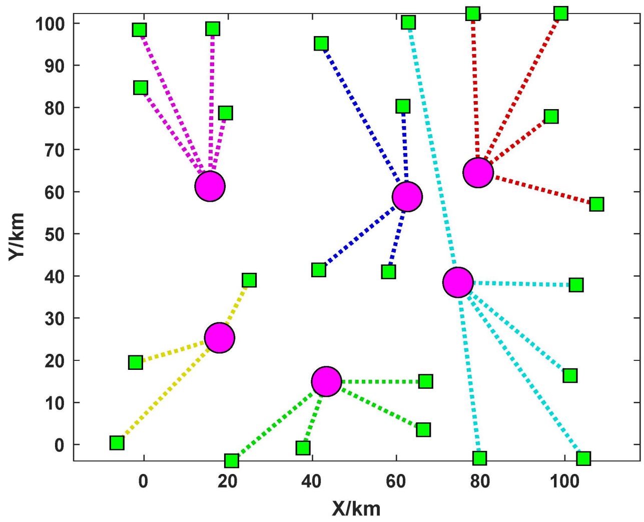
Fig. 16 Convergence curves under different algorithms.



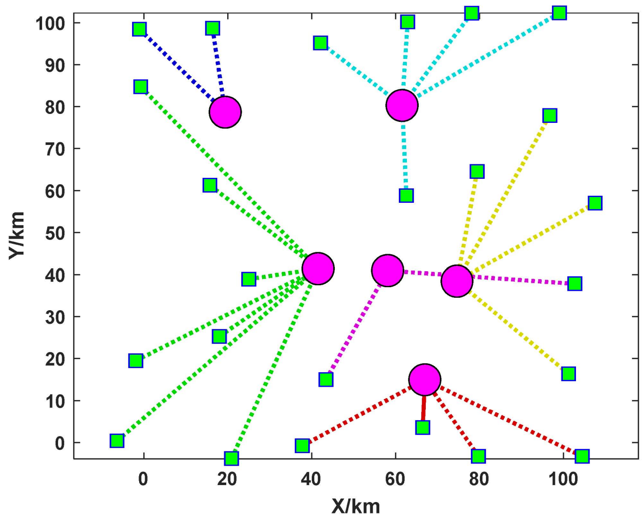

(a) CSO (model one)



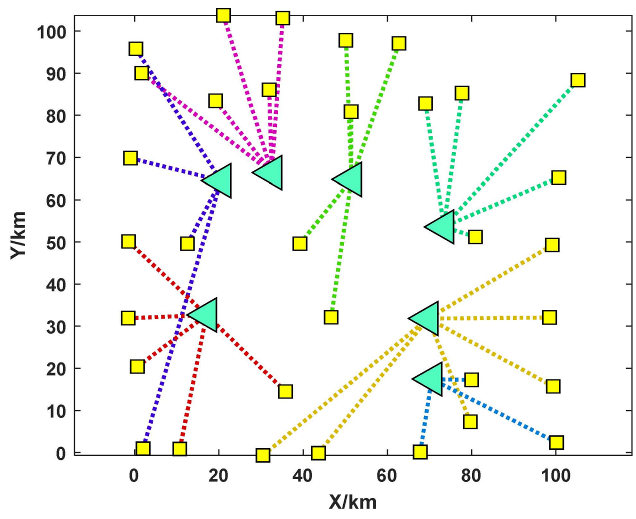

(b) HS (model one)



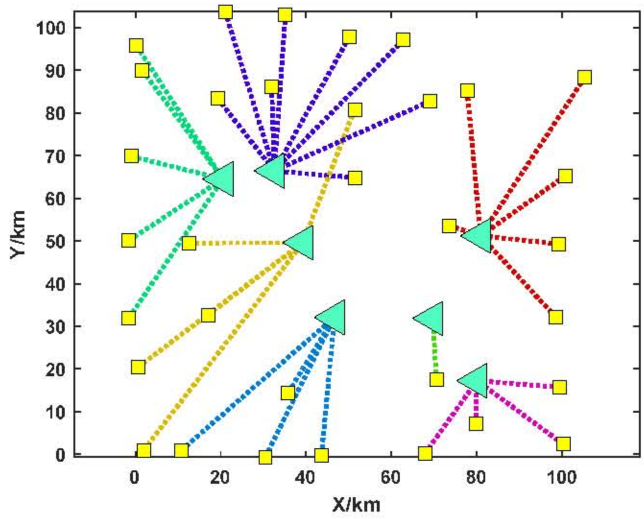

(c) CSO (model two)




(d) HS (model two)




(e) CSO (model three)




(f) HS (model three)

Fig. 17 Distribution map.

TABLE 4. SIMULATION RESULTS UNDER DIFFERENT ALGORITHMS

| Algorithm | Parameters | Optimal Solution | Mean value | Standard deviation |
|---|---|---|---|---|
| PSO with adaptive $\omega$ | $N = 20, P_{max} = 3$ | 53942415 | 54584234.9 | 409722.99 |
| CSO | $N = 20, P_{max} = 3$ | 54143520 | 55074211.4 | 576368.93 |
| HS | $N = 20, P_{max} = 3$ | 54143520 | 54790820.8 | 385685.04 |
| PSO with adaptive $\omega$ | $N = 30, P_{max} = 6$ | 112761326.5 | 116779565.4 | 2690613.52 |
| CSO | $N = 30, P_{max} = 6$ | 118776421 | 1832468.82 | 121795044.7 |
| HS | $N = 30, P_{max} = 6$ | 120106623.5 | 124755603.3 | 2989492.92 |
| PSO with adaptive $\omega$ | $N = 40, P_{max} = 7$ | 194393330.5 | 199757698.5 | 3409794.72 |
| CSO | $N = 40, P_{max} = 7$ | 202979668.5 | 210798843.3 | 5459655.61 |
| HS | $N = 40, P_{max} = 7$ | 202480440.5 | 210950435.2 | 4464584.32 |

It can be seen from the three sets of convergence curves in Fig. 16 that the improved PSO has stronger ability to optimize in the later stage of iteration than the two classical algorithms. By comparison, it can be concluded that the improved PSO has higher efficiency.

## V. CONCLUSION

The PSO algorithm has a wide range of applications in various fields of real life. Solving p-Hub allocation problem is also a hot topic in its application. This paper briefly introduces the research background and current situation of p-Hub allocation problem, and gives its mathematical model. This paper expounds the principle and procedure of PSO algorithm, and gives several improved strategies. By comparing the simulation results of different inertia weights, and the simulation results of three groups of models, it is concluded that five inertia weight strategies are better than the basic PSO algorithm. The adaptive inertia weight has obvious advantages in the optimization results and optimization stability. The parabolic inertia weight is similar to the linear decreasing inertia weight, but the simulation effect is better than the linear decreasing strategy. Both of them have good optimization effect on the PSO algorithm. The random inertia weight has a certain effect on the algorithm optimization. However, because of its random characteristics, the simulation effect decreases with the increase of the solution model. The sin strategy starts to improve its global search ability in the medium term, which leads to the poor optimization effect in the early iteration and affects the overall optimization effect. The feasibility and effectiveness of the p-Hub assignment problem based on PSO algorithm are verified by comparing it with other optimization algorithm.

## REFERENCES


[1] M. J. Feizollahi, and M. Modarres, "The Robust Deviation Redundancy Allocation Problem with Interval Component Reliabilities," *IEEE Transactions on Reliability*, vol. 61, no. 4, pp. 957-965, 2012.

[2] A. Ghaderi, and R. Rahmaniani, "Meta-heuristic Solution Approaches for Robust Single Allocation P-hub Median Problem with Stochastic Demands and Travel Times," *International Journal of Advanced Manufacturing Technology*, vol. 82, no. 9-12, pp. 1627-1647, 2015.

[3] A. M. Campbell, T. J. Lowe, and L. Zhang, "The P-hub Center Allocation Problem," *European Journal of Operational Research*, vol. 176, no. 2, pp. 819-835, 2009.

[4] J. Ebery, "Solving Large Single Allocation P-hub Problems with Two or Three Hubs," *European Journal of Operational Research*, vol. 128, no. 2, pp. 447-458, 2001.

[5] J. G. Klincewicz, "Avoiding Local Optima in the P-hub Location Problem Using Tabu Search and GRASP," *Annals of Operations Research*, vol. 40, no. 1, pp. 283-302, 1992.

[6] B. Rostami, N. Kmmerling, and J. Naoum-Sawaya, "Stochastic Single-allocation Hub Location," *European Journal of Operational Research*, vol. 289, no. 3, pp. 1087-1106, 2021.

[7] M. A. Gargouri, N. Hamani, and N. Mrabti, "Optimization of the Collaborative Hub Location Problem with Metaheuristics," *Mathematics*, vol. 9, no. 21, pp. 2759-2759, 2021.

[8] A. T. Ernst, and M. Krishnamoorthy, "Exact and Heuristic Algorithms for the Uncapacitated Multiple Allocation P-hub Median Problem," *European Journal of Operational Research*, vol. 104, no. 1, pp. 100-112, 1998.

[9] A. Ilic, D. Urosevic, and J. Brimberg, "A General Variable Neighborhood Search for Solving the Uncapacitated Single Allocation P-hub Median Problem," *European Journal of Operational Research*, vol. 206, no. 2, pp. 289-300, 2010.

[10] J. Kratica, Z. Stanimirovi, and T. Duan, "Two Genetic Algorithms for Solving the Uncapacitated Single Allocation P-hub Median Problem," *European Journal of Operational Research*, vol. 182, no. 1, pp. 15-28, 2007.

[11] M. Bashir, M. Mirzaei, and M. Randall, "Modeling Fuzzy Capacitated P-hub Center Problem and a Genetic Algorithm Solution," *Applied Mathematical Modelling*, vol. 37, no.5, pp. 3513-3525, 2013.

[12] D. Wang, J. S. Wang, S. Y. Wang, C. Xing, and X. D. Li, "ANFIS Soft Sensing Model of SMB Chromatographic Separation Process Based on New Adaptive Population Evolution Particle Swarm Optimization Algorithm," *Journal of Intelligent & Fuzzy Systems*, vol. 41, no. 6, pp. 6755-6780, 2021.

[13] W. Z. Sun, M. Zhang, J. S. Wang, S. S. Guo, M. Wang, and W. K. Hao, "Binary Particle Swarm Optimization Algorithm Based on Z-shaped Probability Transfer Function to Solve 0-1 Knapsack Problem," *IAENG International Journal of Computer Science*, vol. 48, no. 2, pp. 294-303, 2021.

[14] Y. X. Lu, J. S. Wang, and S. S. Guo, "Solving Path Planning Problem Based on Particle Swarm Optimization Algorithm with Improved Inertia Weights," *IAENG International Journal of Computer Science*, vol. 46, no. 4, pp. 628-636, 2019.

[15] S. Z. Gao, X. F. Wu, L. L. Luan, J. S. Wang, and G. C. Wang, "PSO Optimal Control of Model-free Adaptive Control for PVC Polymerization Process," *International Journal of Automation and Computing*, vol. 15, no. 4, pp. 482-491, 2018.

[16] C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling Multiple Objectives with Particle Swarm Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256-279, 2004.

[17] M. A. Abido, "Optimal Power Flow Using Particle Swarm Optimization," *International Journal of Electrical Power & Energy Systems*, vol. 24, no. 7, pp. 563-571, 2002.

[18] M. E. O'kelly, “A Quadratic Integer Program for the Location of Interacting Hub Facilities,” *European Journal of Operational Research*, vol. 32, no. 3, pp. 393-404, 1987.
[19] J. Kennedy, and R. Eberhart, “Particle Swarm Optimization,” *ICNN95-International Conference on Neural Networks*, vol. 4, pp. 1942-1948, 1995.
[20] J. Robinson, and Y. Rahmat-Samii, “Particle Swarm Optimization in Electromagnetics,” *IEEE Trans. Antennas Propag*, vol. 52, no. 2, pp. 397-407, 2004.
[21] A. Chatterjee, and P. Siarry, “Nonlinear Inertia Weight Variation for Dynamic Adaptation in Particle Swarm Optimization,” *Computers & Operation Research*, vol. 33, no. 3, pp. 859-871, 2006.
[22] Z. Z. Liu, and Y. Liu, “Clustering Routing Algorithm for Wireless Sensor Networks Based on S-PSO,” *Journal of Chinese Computer Systems*, vol. 38, no. 10, pp. 2207-2210, 2017.
[23] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, “Self-organizing Hierarchical Particle Swarm Optimizer with Time-varying Acceleration Coefficients,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240-255, 2004.