Modified Parallel Tunicate Swarm Algorithm and Application in 3D WSNs Coverage Optimization

Jianpo Li[1], Geng-Chen Li[1], Shu-Chuan Chu[2], Min Gao[1], Jeng-Shyang Pan[1,2,3*]

[1] School of Computer Science, Northeast Electric Power University, China
[2] College of Computer Science and Engineering, Shandong University of Science and Technology, China
[3] Department of Information Management, Chaoyang University of Technology, Taiwan
jianpoli@163.com, 605544372@qq.com, scchu0803@gmail.com, 15822553806@163.com, jspan@cc.kuas.edu.tw

Abstract

As the application of Wireless Sensor Networks (WSNs) in today's society becomes more and more extensive, and the status is getting higher and higher, the node layout of sensors has also begun to attract social attention. In reality, the coverage of WSNs in 3D space is particularly important. Therefore, it is worth investigating an efficient way to find out the maximum coverage of WSNs. In this paper, a Modified Parallel Tunicate Swarm Algorithm (MPTSA) is proposed based on modified parallelism, which can improve the convergence of the algorithm and optimal global solution. Next, the proposed MPTSA is implemented and tested on 23 benchmark functions to verify the algorithm performance. Finally, a WSNs network layout scheme based on MPTSA is proposed to improve the coverage of the whole network. Experimental results show that, compared with the traditional PSO (Particle Swarm Optimization), improved PSO (PPSO and APSO), GBMO (Gases Brownian Motion Optimization) and traditional TSA, MPTSA family algorithms show better performance in WSNs network layout.

Keywords: Network layout, Modified parallel TSA, Tunicate swarm algorithm, Wireless sensor networks

1 Introduction

In recent decades, many scholars at home and abroad have proposed a variety of evolutionary algorithms [1-4]. For example, the classic Differential Evolution (DE) algorithm [5], Particle Swarm Optimization (PSO) [6], Gases Brownian Motion Optimization (GBMO) [7] and the QUasi-affine TRansformation Evolutionary (QUATRE) algorithm [8] have been proposed. In [9], the population is generated randomly through vector coding to carry out mutation and crossover operations between two individuals to search for optimization. These algorithms are inspired by biological evolution or biological habits in nature [10-11]. Therefore, each algorithm has its own rules and characteristics according to its inspiration source [12-15]. The evolutionary algorithm includes two parts: exploration and development [16]. Exploration is a process in which a group of organisms finds the best solution in situ, while development is an external search for the latest solution strategy [17-21]. In this paper, a new heuristic algorithm named TSA [22] inspired by the successful group behavior of membrane animals in the deep sea [23] is studied. But like most evolutionary algorithms, this algorithm has some problems, such as poor convergence, easy to fall into the global optimum, etc [24-27].

With the development of industrial engineering, communication theory and various fields in society, the birth of WSNs provides a lot of convenient conditions for people's life [28-30]. People can monitor the surrounding environment and the status of various devices through WSNs in real-time. At present, WSNs has been integrated into many aspects of society and is widely used in various fields, such as environmental monitoring, smart home, agricultural production, image processing, urban transportation, etc [31-33]. Therefore, the research of WSNs network layout is particularly important [34-37]. The node deployment of WSNs directly affects the whole network coverage, and the network coverage further influences the monitoring quality of monitoring areas [38-39]. There are two traditional sensor node placement methods: deterministic coverage and random coverage. Deterministic coverage is that sensor nodes are placed in a static environment according to a predetermined location. In the random coverage, sensor nodes are randomly deployed on vehicles such as automobiles, airplanes, and ships [40-41]. At present, most of the researches is focused on the coverage of the 2D plane, which is obviously far from enough [42]. Considering the actual situation, we do more research on the coverage problem under the 3D environment [43-44]. Therefore, the coverage research in the 3D environment has been widely concerned by researchers.

There are two scenarios in the three-dimensional deployment environment. On the one hand, Nodes are deployed in the entire three-dimensional space, such as underwater sensor deployment. In [45], Khalfallah et al. proposed a new 3D underwater WSNs deployment plan (3D-UWSN-Deploy). For solid detection in rivers, it is necessary to ensure the monitoring quality and network connectivity and to minimize the number of underwater sensor nodes. The simulation results show that the scheme is effective in terms of deployment cost, monitoring quality, and network connectivity. A WSNs deployment optimization algorithm based on distributed parallel particle swarm optimization is proposed in [46], which takes coverage and life cycle as the optimization goals to maximize coverage and extend network life in a three-dimensional space with obstacles. On the other hand, Nodes are deployed on three-dimensional surface

*Corresponding Author: Jeng-Shyang Pan; E-mail: jspan@cc.kuas.edu.tw

DOI: 10.53106/160792642022032302004

coverage research, such as mountain surface coverage. Nazarzehi et al. [47] proposed an optimized deployment algorithm for the surface coverage of WSNs nodes. A decentralized random algorithm is used to drive nodes to move at the vertices of a truncated octahedral mesh to achieve full coverage of the 3D area. In order to maximize the coverage of WSNs, Anand et al. [48] proposed a deterministic deployment plan suitable for three-dimensional curved surfaces. By modifying the traditional Voronoi diagram algorithm and using it to classify terrain sectors, the experiment showed that the algorithm was significantly enhanced in terms of node uniformity, coverage and other performance. Literature [49] studied the deployment of WSNs on 3D terrain surfaces. In order to ensure the full coverage of 3D surfaces, while ensuring network connectivity, while minimizing energy consumption, Boufare et al. proposed a distribution based on an improved virtual force strategy. The deployment algorithm (3D-IDVFA-TC) can move nodes on 3D smooth, undulating and rough surfaces. The simulation results show that the algorithm can guarantee full coverage regardless of the 3D terrain, while effectively reducing node energy consumption.

To solve the problem mentioned above, the modified parallel strategies are added to the traditional TSA to improve the algorithm. Finally, the proposed MPTSA is used to optimize the 3D coverage of WSNs. The working arrangements of other parts are as follows:

Firstly, the traditional TSA is introduced, and the shortcomings of the traditional algorithm are analyzed. Secondly, aiming at the shortcomings of TSA, four modified parallel strategies are proposed to improve the performance of TSA. The performance of the proposed algorithm is verified by the test function. Next, a 3D coverage model of WSNs is introduced, and the proposed algorithm is used to optimize the model to verify the applicability of the algorithm in this field. Finally, the work of this paper is summarized.

2 Tunicate Swarm Algorithm

TSA is a new meta-heuristic algorithm based on bioinspired, which was proposed by Satnam Kaur and others in 2020 to optimize nonlinear constraint problems. Tunicate has the ability to find food sources in the ocean. Yet, in a given search space, we don't know where the food comes from. Membrane animals use their own two behaviors to find the best food source. Their behaviors include jet propulsion and swarm intelligence. In order to simulate the jet propulsion behavior more realistically, the membrane must satisfy the following three conditions: avoiding the conflicts between the search population, moving to the position of the best searching individual, and keeping the distance from the best searching individual [50]. Then, the population will update the position according to the individual's optimal solution. The inspiration of the algorithm is the swarm behavior of the successful survival membrane in the deep sea. Next, we will describe the heuristic and mathematical modeling of TSA in detail.

2.1 Avoid Conflicts between Search Agents

In order to avoid the conflicts between the current tunicate and other ones, the vector $\vec{A}$ is used to calculate the new particle position.

$$\vec{A} = \frac{\vec{G}}{\vec{M}} \tag{1}$$

$$\vec{G} = \vec{c_1} + \vec{c_2} - \vec{F} \tag{2}$$

$$\vec{F} = 2 \cdot \vec{c_3} \tag{3}$$

Where $\vec{G}$ represents gravity and $\vec{F}$ represents advection in the deep sea, $\vec{c_1}, \vec{c_2}, \vec{c_3}$ are random numbers between [0, 1], $\vec{M}$ represents the interaction force between individuals, which is calculated as follows:

$$\vec{M} = \overrightarrow{P_{\min}} + \vec{c_3} \cdot \overrightarrow{P_{\max}} - \overrightarrow{P_{\min}} \tag{4}$$

Where $\overrightarrow{P_{\min}}$ and $\overrightarrow{P_{\max}}$ represent the initial and subordinate speeds to make social interaction. In this study, $\overrightarrow{P_{\min}}$ is equal to 1 and $\overrightarrow{P_{\max}}$ is equal to 4.

2.2 Moving towards the Best Neighbor Particle

After avoiding the collision between neighboring particles, each particle moves to the best one among the neighboring particles.

$$\overrightarrow{PD} = \left\lfloor \overrightarrow{FS} - r_{and} \cdot \overrightarrow{P_P(x)} \right\rfloor \tag{5}$$

Where $\overrightarrow{PD}$ is the distance between the location of the food and the current particle, x indicates the current iteration, $\overrightarrow{FS}$ is the location of the food (the current global optimal solution). Vector $\overrightarrow{P_p}$ indicates the position of tunicate and r_{and} is a random number in range [0, 1].

2.3 Moving to Global Optimal Solution

Each particle will continue to move towards the current global optimal solution (the location of the food source).

$$\overrightarrow{P_P(x')} = \begin{cases} \overrightarrow{FS} + \vec{A} \cdot \overrightarrow{PD}, & if\ r_{and} \geq 0.5 \\ \overrightarrow{FS} - \vec{A} \cdot \overrightarrow{PD}, & if\ r_{and} \geq 0.5 \end{cases} \tag{6}$$

Where $\overrightarrow{P_P(x')}$ represents the position of particle x after moving towards food source $\overrightarrow{FS}$.

2.4 Swarm Behavior

In order to simulate the population behavior of the whole population, the first two optimal solutions are saved and the positions of other particles are updated according to the position of the optimal particle. The group behavior of the membrane animals group was defined according to equation (7):

$$\overrightarrow{P_P(x+1)} = \frac{\overrightarrow{P_P(x)} + \overrightarrow{P_P(x+1)}}{2+c_3} \qquad \textbf{(7)}$$

Each particle is updated to a random position according to equation (7).

3 Modified Parallel Tunicate Swarm Algorithm

3.1 Traditional Parallel Strategy

Most heuristic algorithms have a fatal shortcoming, which is easy to fall into local optimality. In view of this defect, many researchers put forward different improvement ideas to solve this key problem. Among them, the idea of parallelism has a very good effect in many improvement ideas. The following will explain the traditional idea of parallelism [51-53].

In order to construct a parallel architecture, groups are grouped to form groups of parallel structures. Information is then exchanged between each group at a set number of iterations. The particle with the worst fitness function value in each group is replaced by the particle with the best fitness function value in all groups. Figure 1 shows the structure of the traditional parallelism strategy.

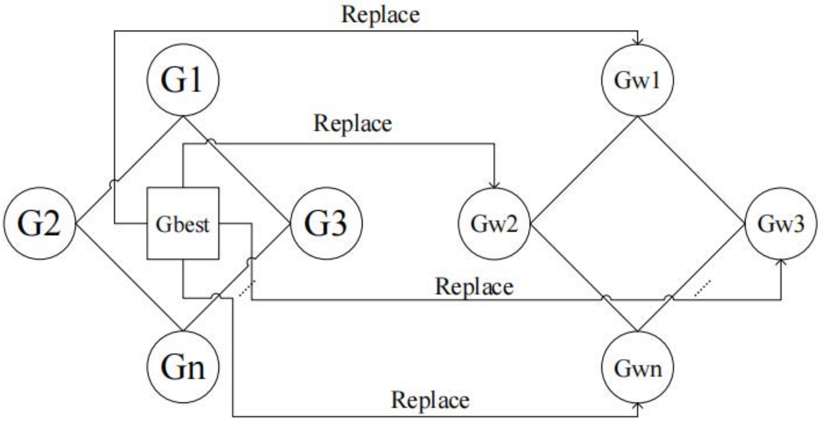


Figure 1. Traditional parallelism strategy

In Figure 1, the $G_1, G_2, G_3, ..., G_n$ means that the population is evenly divided into n groups. G_{best} represents the position of the particle with the best fitness function value in all groups. $G_{w1}, G_{w2}, G_{w3}, ..., G_{wn}$ respectively indicate the position of the particle with the worst fitness function value in each group. The traditional idea of parallel is to divide the population into n groups. In each information exchange, G_{best} with the best fitness function value is selected from n groups, and G_{best} is used to replace $G_{w1}, G_{w2}, G_{w3}, ..., G_{wn}$ with the worst fitness function value in each group. This operation will continue until the end of the algorithm.

Inspired by the traditional idea of parallelism, this paper proposes four modified parallelism strategies, and combines it with TSA to propose four MPTSA using different communication strategies.

3.2 Modified Parallel Tunicate Swarm Algorithm

Like most heuristic algorithms, TSA also has some disadvantages, such as poor convergence and easy to fall into local optimum. Therefore, in this part of the work, we will modify TSA to improve the convergence and accuracy of the algorithm. In addition to the different communication strategies, the four MPTSA will use the same grouping method and preparation before communication. The specific operation is as follows.

Assuming that there are pop particles in the tunicate population, the whole population is divided into g group, then there are particles in each group, and the particles in the g group are optimized at the same time. When the number of iterations t reaches an integer multiple of R, an information exchange is carried out, and the four proposed MPTSA will use four different communication strategies for information exchange. At the end of this operation, the population will return to the optimization state at the position after information exchange to continue the optimization work. The final result is output until the algorithm runs to the pre-set maximum number of iterations $iter_{\max}$ or meets the required precision. Four different communication strategies are described in detail below.

3.2.1 Communicate by Optimal Values

After the grouping operation above, all the particles have entered the working state. When the number of iterations reaches R, the particles in each group are sorted according to the value of fitness function, and the population after the arrangement is divided into two parts: Part A and B . Among them, the A part represents the particles whose fitness function values rank in the first half. Then, the particle order after the A part arrangement is: $x_{Abest}\left(x_{A1}\right) > x_{A2} > \ldots > x_{Aworst}\left(x_{A\frac{pop}{2}}\right)$.

The B part represents the particle in the bottom half of the fitness function. Then, the order of particles after the B part is: $x_{Bbest}\left(x_{B1}\right) > x_{B2} > \ldots > x_{Bworst}\left(x_{B\frac{pop}{2}}\right)$. It is worth noting that the fitness function value of x_{Aworst} is better than that of x_{B1} .The reason for this is that the algorithm itself has the disadvantage of poor convergence, and using the best particle to replace the worst particle can accelerate the convergence speed of the algorithm to a certain extent, so that the algorithm can find the global optimal solution faster.

Next, the operation of information exchange will take place. The strategy used in this section is to use optimal values for communication. In each information exchange, the best value of A and B will be used to replace the worst value of their parts respectively. The population will be adjusted according to equation (8).

$$\begin{cases} x_{iAworst} = x_{iAbest}\left(x_{iA1}\right), & if\ x \in Part\ A \\ x_{iBworst} = x_{iBbest}\left(x_{iB1}\right), & if\ x \in Part\ B \end{cases} \quad \textbf{(8)}$$

Where $x_{Abest}\left(x_{A1}\right)$ and $x_{Bbest}\left(x_{B1}\right)$ represent the position of the particle with the best fitness function value in the two parts of A and B respectively. x_{Aworst} and x_{Bworst} represent the positions of the particles with the worst fitness function values in the two parts of A and B respectively. i stands for the i-thgroup.

After the exchange of information, the replaced particle will be optimized again at the updated position. When the number of iterations t reaches an integer multiple of R again, information will be exchanged again, and the cycle will continue until the end of the algorithm. The schematic diagram is shown in Figure 2.

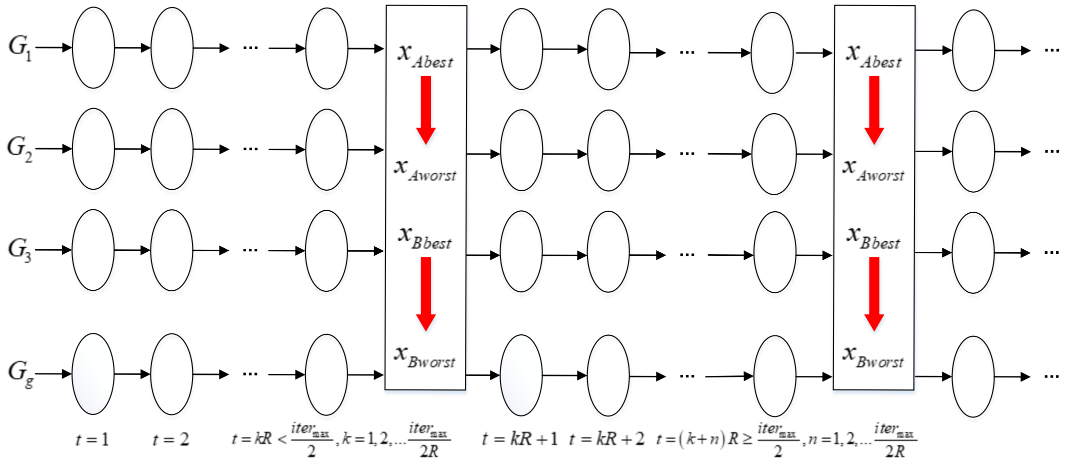

G_1
G_2
G_3
G_g
...
x_{Abest}
x_{Aworst}
x_{Bbest}
x_{Bworst}
$t=1$
$t=2$
$t=kR<\frac{iter_{\max}}{2},k=1,2,\ldots\frac{iter_{\max}}{2R}$
$t=kR+1$
$t=kR+2$
$t=(k+n)R\geq\frac{iter_{\max}}{2},n=1,2,\ldots\frac{iter_{\max}}{2R}$

Figure 2. Communicate by optimal values

3.2.2 Communicate by Average Values

Same as the first MPTSA, the algorithm performs grouping operation first, and then all particles enter the optimization state. When the number of iterations is $t = kR, k = 1, 2, \ldots, \frac{iter_{\max}}{2R}$, the particles are sorted according to the value of fitness function. However, different from the previous method, MPTSA using average value communication will divide the operation of information exchange into two different strategies for information exchange according to the current iteration number of t . Assume that the maximum number of iterations is set as $iter_{\max}$. When the number of iterations $t < \frac{iter_{\max}}{2}$, the average value is used for information exchange. As shown in equation (9) and equation (10), the average values of the positions of the particles in A and B were first calculated, which were denoted as x_{Aave} and x_{Bave} .

$$x_{Aave} = \frac{x_{A1} + x_{A2} + \ldots + x_{Aworst}}{pop/\left(2 \times g\right)} \quad \textbf{(9)}$$

$$x_{Bave} = \frac{x_{B1} + x_{B2} + \ldots + x_{Bworst}}{pop/\left(2 \times g\right)} \quad \textbf{(10)}$$

After the average is obtained, the algorithm replaces the worst value in the A part with the average value in the A part. The B part is the same as the A part.

When the number of iterations $t \geq \frac{iter_{\max}}{2}$, it will be the same as the first strategy, using the optimal value for information exchange. Therefore, the communication strategy proposed in this section is shown in equation (11) and equation (12). Figure 3 shows a schematic diagram of this communication strategy.

$$\begin{cases} x_{iAworst} = x_{iAave}, & if\ x \in Part\ A \\ x_{iBworst} = x_{iBave}, & if\ x \in Part\ B \end{cases}, t < \frac{iter_{\max}}{2} \quad \textbf{(11)}$$

$$\begin{cases} x_{iAworst} = x_{iAbest}, & if\ x \in Part\ A \\ x_{iBworst} = x_{iBbest}, & if\ x \in Part\ B \end{cases}, t \geq \frac{iter_{\max}}{2} \quad \textbf{(12)}$$

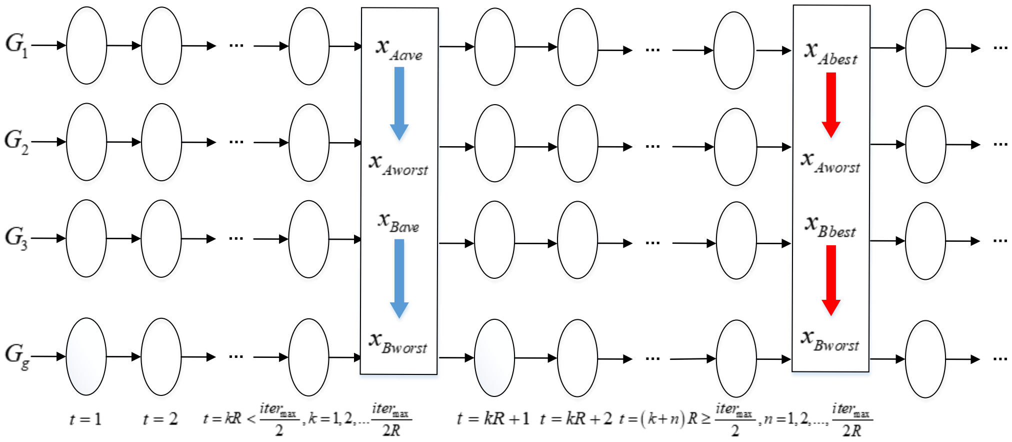

G_1
G_2
G_3
G_g
x_{Aave}
x_{Aworst}
x_{Bave}
x_{Bworst}
x_{Abest}
x_{Aworst}
x_{Bbest}
x_{Bworst}
$t=1$
$t=2$
$t=kR<\frac{iter_{\max}}{2}, k=1,2,\ldots\frac{iter_{\max}}{2R}$
$t=kR+1$
$t=kR+2$
$t=(k+n)R \geq \frac{iter_{\max}}{2}, n=1,2,\ldots,\frac{iter_{\max}}{2R}$

Figure 3. Communicate by average values

3.2.3 Communicate by Median

Before information exchange, this strategy still needs to judge the numerical relationship between the current iteration number t and the maximum iteration number $iter_{\max}$. When $t < \frac{iter_{\max}}{2}$, the particles in each group are first divided into two parts: A and B, and the median x_{Amed} and x_{Bmed} are selected from these two parts respectively. Then, select the worst of the two particles, x_{Aworst} and x_{Bworst}. Replace x_{Aworst} and x_{Bworst} with x_{Amed} and x_{Bmed}.

When $t \geq \frac{iter_{\max}}{2}$, the particles in each group are sorted by the fitness function values, which is consistent with the previous two strategies.

Information is exchanged using the optimal values consistent with the first strategy. Therefore, the communication strategy proposed in this section is shown in equation (13) and equation (14).

$$\begin{cases} x_{iAworst} = x_{iAmed}, & if\ x \in Part\ A \\ x_{iBworst} = x_{iBmed}, & if\ x \in Part\ B \end{cases}, t < \frac{iter_{\max}}{2} \quad \textbf{(13)}$$

$$\begin{cases} x_{iAworst} = x_{iAbest}, & if\ x \in Part\ A \\ x_{iBworst} = x_{iBbest}, & if\ x \in Part\ B \end{cases}, t \geq \frac{iter_{\max}}{2} \quad \textbf{(14)}$$

Figure 4. shows the structure of this communication strategy.

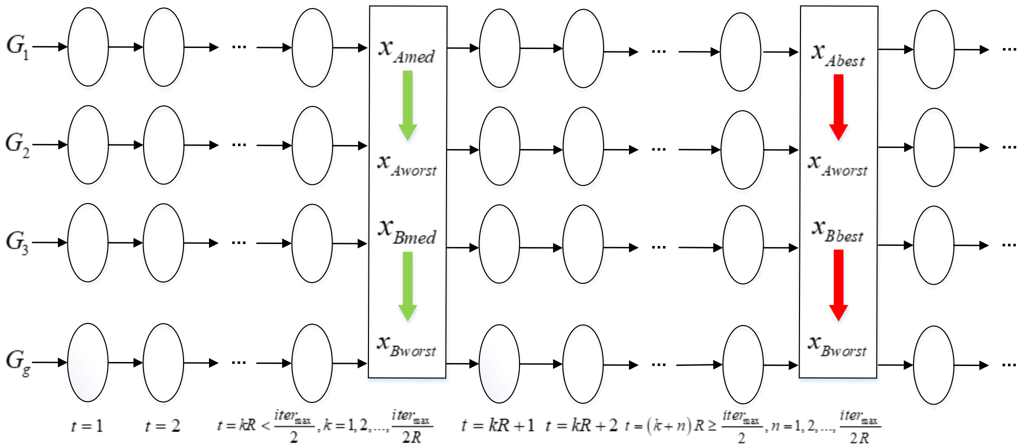

G_1
G_2
G_3
G_g
x_{Amed}
x_{Aworst}
x_{Bmed}
x_{Bworst}
x_{Abest}
x_{Aworst}
x_{Bbest}
x_{Bworst}
$t=1$
$t=2$
$t=kR<\frac{iter_{\max}}{2}, k=1,2,\ldots,\frac{iter_{\max}}{2R}$
$t=kR+1$
$t=kR+2$
$t=(k+n)R \geq \frac{iter_{\max}}{2}, n=1,2,\ldots,\frac{iter_{\max}}{2R}$

Figure 4. Communicate by median

3.2.4 Communicate by Random Number

Like the second and third methods, MPTSA using random number communication does not need to sort the fitness function values, but only needs to divide the particles in each group into two parts: the A part and the B part. But in the second and third strategies, it is the numerical relationship between t and $\frac{iter_{\max}}{2}$ that needs to be determined. In this strategy, $\frac{2}{3}iter_{\max}$ will be compared with the current iteration number t. When $t < \frac{2}{3}iter_{\max}$, a particle x_{Arand} and x_{Brand} are randomly selected from the two parts of A and B respectively. These two particles were used as substitutes to replace the particles x_{Aworst} and x_{Bworst} with the worst fitness function values.

When $t \geq \frac{2}{3}iter_{\max}$, the first communication strategy is still used for communication. The purpose of doing this is to make the algorithm have more chances to jump out of the local optimal solution and find the global optimal solution in the initial stage of algorithm operation. In the late convergence stage of the algorithm, the optimal value is used to replace the worst value, which can accelerate the convergence speed and improve the convergence of the algorithm. To sum up, the strategy of using random values for information exchange is shown in equation (15) and equation (16). For ease of understanding, Figure 5 shows the communication strategy proposed in this section.

$$\begin{cases} x_{iAworst} = x_{iArand}, & if\ x \in Part\ A \\ x_{iBworst} = x_{iBrand}, & if\ x \in Part\ B \end{cases}, t < \frac{2}{3}iter_{\max} \tag{15}$$

$$\begin{cases} x_{iAworst} = x_{iAbest}, & if\ x \in Part\ A \\ x_{iBworst} = x_{iBbest}, & if\ x \in Part\ B \end{cases}, t \geq \frac{2}{3}iter_{\max} \tag{16}$$

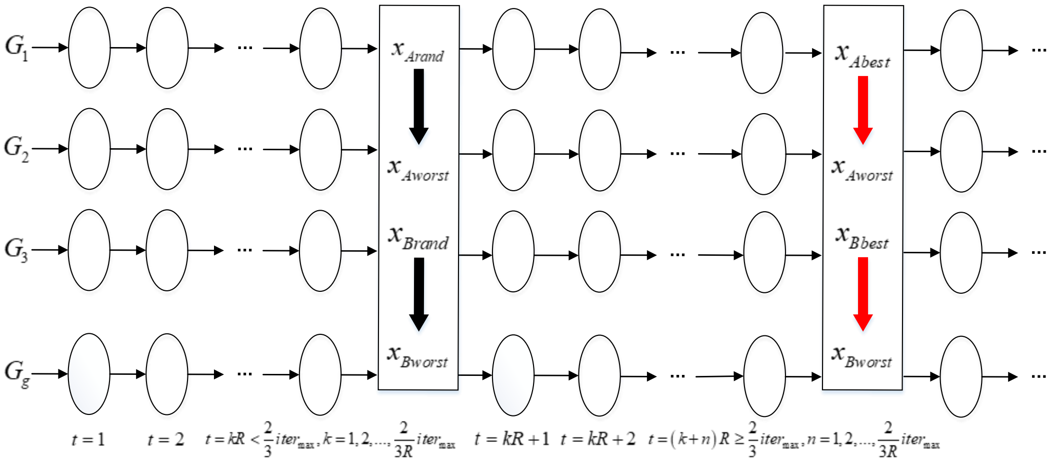

G_1
G_2
G_3
G_g
x_{Arand}
x_{Aworst}
x_{Brand}
x_{Bworst}
x_{Abest}
x_{Aworst}
x_{Bbest}
x_{Bworst}
$t=1$
$t=2$
$t=kR<\frac{2}{3}iter_{\max}, k=1,2,\dots,\frac{2}{3R}iter_{\max}$
$t=kR+1$
$t=kR+2$
$t=(k+n)R \geq \frac{2}{3}iter_{\max}, n=1,2,\dots,\frac{2}{3R}iter_{\max}$

Figure 5. Communicate by random number

This section introduces the traditional parallel strategy and proposes four different communication methods. Next, we apply the proposed four communication methods to the traditional parallel strategy and get four modified parallel strategies. At the same time, the traditional TSA is optimized by these four strategies, and the MPTSA series algorithms are obtained. According to different communication methods, they are named OMPTSA, AMPTSA, MMPTSA, RMPTSA respectively.

4 Experimental Simulation and Analysis of the Proposed Algorithm

In this part, in order to test the performance of the proposed algorithm, this paper will use 23 test functions to test, and this work is carried out in the Matlab2015b test environment.

4.1 Setting of Experimental Parameters

In order to verify the performance of the algorithm, we compared the proposed algorithm with the traditional PSO, GBMO, TSA and the improved versions of PSO (PPSO and APSO). In order to ensure the fairness of the experimental results, we will set uniform parameters for all algorithms except those unique to individual algorithms. The population number pop is set to 40, and the maximum iteration number $iter_{\max}$ is set to 1000. After all the algorithms with parallel strategy (PPSO, OMPTSA, MMPTSA, RMPTSA, AMPTSA) are initialized, the population will be divided into four groups, and the information will be exchanged every 20 iterations. That means $g = 4$ and $R = 20$. The relevant parameter settings of all algorithms are shown in Table 1

Table 1. Parameter setting of all algorithm

Algorithm	Parameter
PSO	$c = 2.0, w = 0.9, pop = 40, iter_{\max} = 1000$
PPSO	$g = 4, R = 20, c = 2.0, w = 0.9, pop = 40, iter_{\max} = 1000$
APSO	$c = 2.0, w = 0.9, pop = 40, iter_{\max} = 1000$
GBMO	$T = 900, a = 1.5, b = 1.2, pop = 40, iter_{\max} = 1000$
TSA	$P_{\min} = 1, P_{\max} = 4, pop = 40, iter_{\max} = 1000$
OMPTSA	$g = 4, R = 20, P_{\min} = 1, P_{\max} = 4, pop = 40, iter_{\max} = 1000$
MMPTSA	$g = 4, R = 20, P_{\min} = 1, P_{\max} = 4, pop = 40, iter_{\max} = 1000$
RMPTSA	$g = 4, R = 20, P_{\min} = 1, P_{\max} = 4, pop = 40, iter_{\max} = 1000$
AMPTSA	$g = 4, R = 20, P_{\min} = 1, P_{\max} = 4, pop = 40, iter_{\max} = 1000$

4.2 Analysis of Experimental Results

Table 2 shows the experimental results of the proposed MPTSA series algorithms tested by test functions. In the proposed MPTSA series, OMPTSA has the highest expectation. However, from the data in Table 2, it can be seen that its optimization ability is better than other algorithms only in F2, F3 and F20, and it is close to other algorithms in F7 and F16 to F19. However, MMPTSA showed the best results nine times. RMPTSA which uses random number to communicate wins 6 times, that is to say, the ability of this strategy is not ideal and does not achieve the expected effect. AMPTSA is more powerful than other MPTSA series algorithms, and it wins 13 times.

In general, F1 to F7 are unimodal functions, which have no local optimal solution but only one global optimal solution. In unimodal functions, OMPTSA and RMPTSA both win twice, while MMPTSA and AMPTSA, which communicate using median and mean values, win only once. In other words, OMPTSA and RMPTSA are more powerful than other algorithms in solving the problem without considering whether the algorithm will fall into local optimality, but there is no significant difference. In the next six multimodal functions, AMPTSA gets the best results five times, MMPTSA wins once. Since there are many local optimal solutions for multimodal functions, we can think that AMPTSA has the strongest ability to jump out of the local optimal, which is superior to other algorithms. In the fixed-dimension multimodal benchmark functions, the performance of MPTSA series algorithms is similar, and there is no obvious difference. It is worth noting that MMPTSA, which uses the median to communicate, performs better in this kind of problem than algorithms using other communication strategies. In this paper, we choose OMPTSA, which has the worst performance among the MPTSA series algorithms, to compare with the experimental results of other comparison algorithms. The specific data are shown in Table 3.

According to the experimental data in Table 3, it is clear that OMPTSA performs very well in the 23 test functions. Despite the improved PPSO and APSO, the algorithm proposed in this paper also shows more powerful optimization ability. However, it is worth noting that in the experimental data in Table 3, only the comparison results of OMPTSA, which is the least capable among the MPTSA series algorithms, and other algorithms are shown. Therefore, GBMO performs better in multimodal functions than OMPTSA. But if the AMPTSA was used for comparison, the difference would not be particularly significant. A brief analysis of the types of test functions used and the results from the test functions follows.

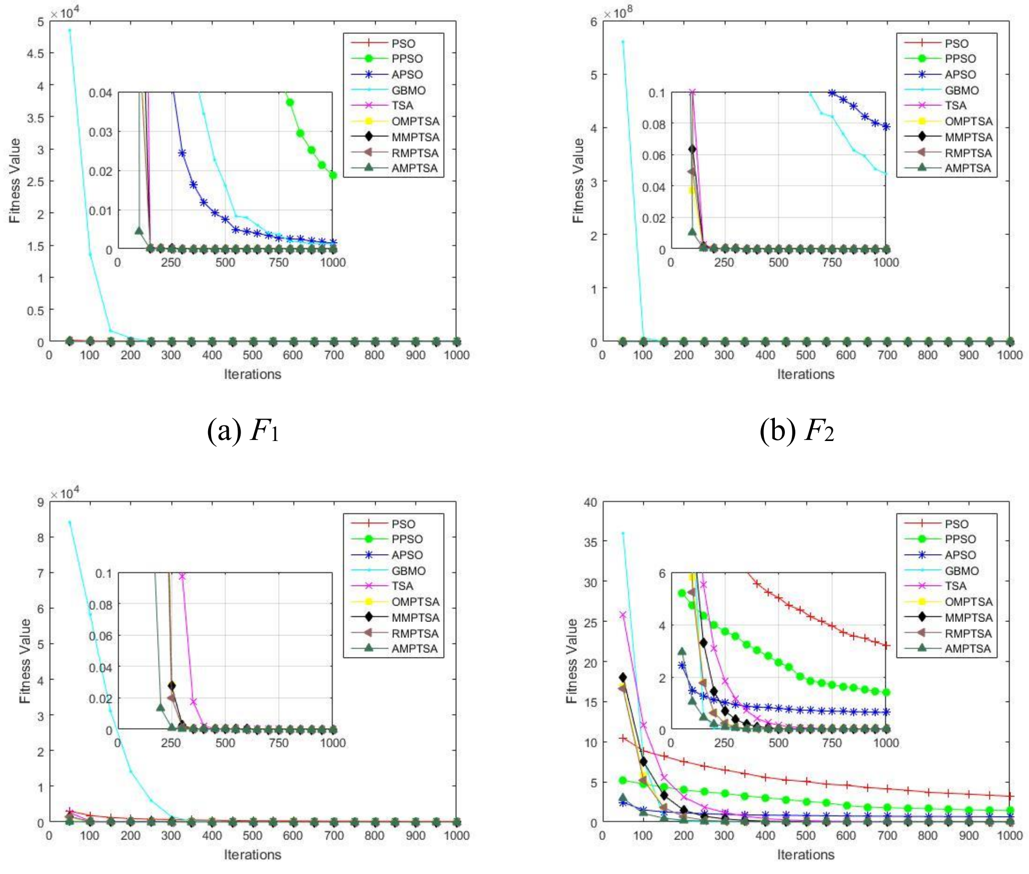
T-test is a way to compare the mean value of two groups of data. It can be used for normal distribution of unknown population standard deviation and small sample. To compare the experimental results more significantly, T-test was used to test the proposed method. Table 4 shows the T-test results of OMPTSA, MMPTSA, RMPTSA and AMPTSA compared with TSA. The "+" indicates that the algorithm is better than TSA, the "-" indicates that the algorithm is worse than TSA, and the "=" indicates that the algorithm and TSA test results are similar.

4.2.1 Unimodal Benchmark Functions

F1 to F7 are unimodal functions. This type of function has only one global optimal solution but no local optimal solution, so it is more suitable for testing the convergence of the algorithm. It can also be seen from the data in Table 3 that among the seven functions, the performance difference of most algorithms is not very obvious, especially the MPTSA series algorithms. Figure 6 shows the test results after testing by unimodal functions. In addition to F6, MPTSA series algorithms all show very strong optimization ability, and the experimental results are also better than other comparison algorithms. However, the results shown in Figure 6 show that AMTPSA is significantly faster in terms of convergence rate. In other words, the AMTPSA that uses averages to communicate has a strong convergence. While other algorithms in the MPTSA series are not as fast as AMPTSA, they are better than other algorithms in both accuracy and convergence.

Table 2. The results of simulation experiment

Function	OMPTSA		MMPTSA		RMPTSA		AMPTSA	
	AVG.	VARP.	AVG.	VARP.	AVG.	VARP.	AVG.	VARP.
F1	4.36E-54	2.12E-106	2.88E-51	1.54E-101	4.00E-54	2.00E-106	2.31E-52	4.31E-103
F2	1.67E-32	7.51E-64	1.50E-30	1.31E-59	2.04E-32	1.76E-63	2.22E-31	6.28E-62
F3	4.42E-19	1.51E-36	3.12E-17	4.94E-33	1.18E-18	1.26E-35	1.86E-18	3.87E-35
F4	8.21E-07	6.51E-12	5.77E-05	5.60E-09	9.10E-08	8.20E-15	9.97E-06	1.65E-10
F5	2.81E+01	7.71E-01	2.75E+01	8.42E-01	2.79E+01	1.41E+00	2.77E+01	8.26E-01
F6	3.19E+00	3.41E-01	2.97E+00	2.82E-01	2.96E+00	5.62E-01	2.90E+00	1.41E-01
F7	1.81E-03	5.11E-07	2.23E-03	3.88E-07	1.32E-03	3.18E-07	1.78E-03	4.82E-07
F8	-6.82E+03	2.58E+05	-6.86E+03	1.62E+05	-6.82E+03	2.68E+05	-7.19E+03	1.43E+05
F9	1.22E+02	1.03E+03	1.22E+02	3.25E+02	1.31E+02	8.59E+02	1.42E+01	7.04E+01
F10	7.77E-01	1.67E+00	4.25E-01	1.18E+00	1.04E+00	1.36E+01	2.98E-01	7.81E-01
F11	4.31E-03	4.39E-05	5.90E-04	4.88E-06	4.01E-03	2.61E-05	3.05E-04	2.69E-06
F12	3.41E+00	6.96E+00	3.05E+00	4.56E+00	4.70E+00	1.24E+01	4.37E-01	1.00E-01
F13	2.49E+00	2.76E-01	2.38E+00	1.48E-01	2.49E+00	2.46E-01	2.29E+00	9.03E-02
F14	3.48E+00	1.27E+01	2.51E+00	3.82E+00	3.26E+00	1.07E+01	3.29E+00	9.21E+00
F15	1.10E-03	1.29E-05	3.54E-04	2.83E-08	4.08E-04	7.50E-08	3.59E-04	1.51E-08
F16	-1.03E+00	1.67E-15	-1.03E+00	1.34E-15	-1.03E+00	1.25E-15	-1.03E+00	1.45E-15
F17	3.98E-01	3.11E-11	3.98E-01	4.12E-12	4.03E-01	6.32E-04	3.98E-01	3.99E-12
F18	3.00E+00	5.82E-14	3.00E+00	1.48E-12	3.00E+00	1.30E-13	3.00E+00	1.42E-12
F19	-3.86E+00	4.04E-11	-3.86E+00	9.24E-11	-3.86E+00	7.90E-11	-3.86E+00	1.57E-10
F20	-3.30E+00	2.25E-03	-3.31E+00	1.27E-03	-3.32E+00	4.55E-04	-3.31E+00	8.83E-04
F21	-9.08E+00	4.07E+00	-9.86E+00	8.03E-01	-8.90E+00	4.54E+00	-9.65E+00	2.50E+00
F22	-9.81E+00	2.50E+00	-1.01E+01	8.73E-01	-9.80E+00	2.46E+00	-1.03E+01	3.92E-03
F23	-9.41E+00	5.26E+00	-1.03E+01	9.12E-01	-9.85E+00	3.68E+00	-1.05E+01	1.75E-03

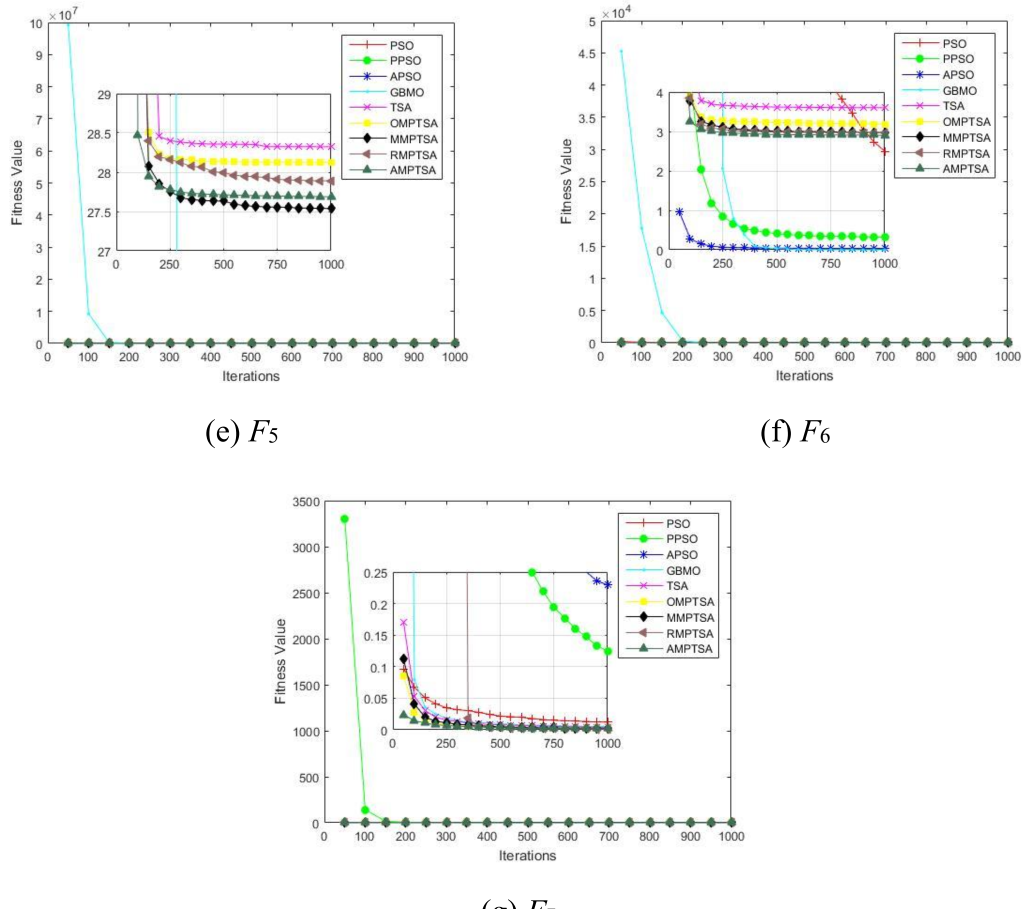


(a) *F*1

(b) *F*2

(c) *F*3

(d) *F*4



(e) F_5

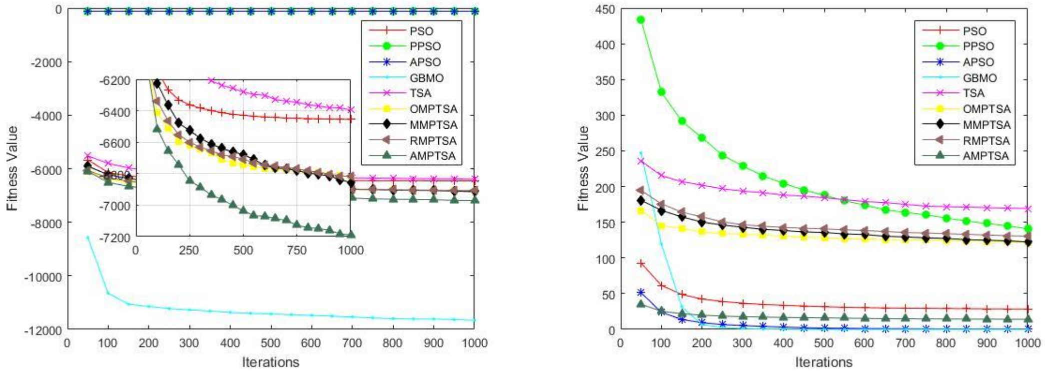
(f) F_6

(g) F_7

Figure 6. Convergence tendency for unimodal benchmark functions

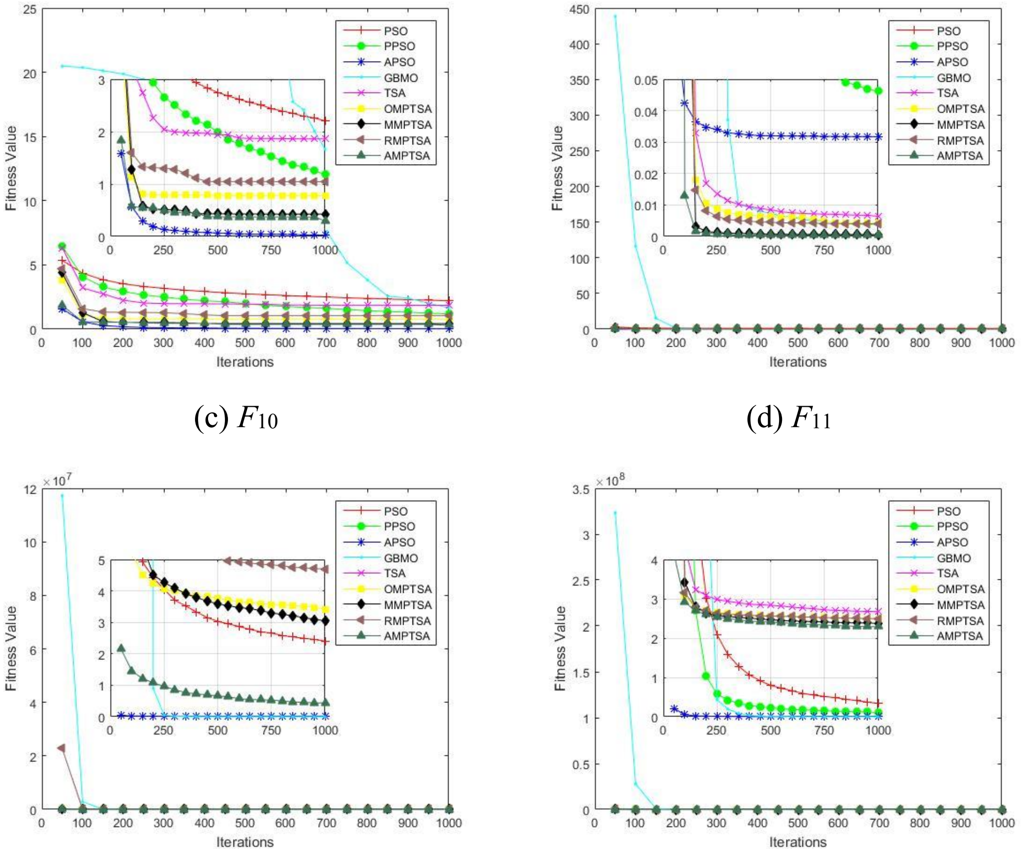
4.2.2 Multimodal Function Analysis

Multimodal functions include F8 to F13. This type of function has many locally optimal solutions, so it is used to test the ability of the algorithm to jump out of the local optimum. From the data in Table 3 and the experimental results in Figure 7, it can be seen that the performance of GBMO and AMPTSA is better than other algorithms in the nine algorithms. Therefore, it can be judged that these two algorithms have a stronger ability to jump out of the local optimal in the running process. RMPTSA, which uses random numbers to communicate, has this ability in theory, but the effect is not ideal. The reason why this happens is that random numbers have a strong randomness, and in F11, RMPTSA shows the ability to compete with AMTPSA and outperform other algorithms.



(a) F_8

(b) F_9



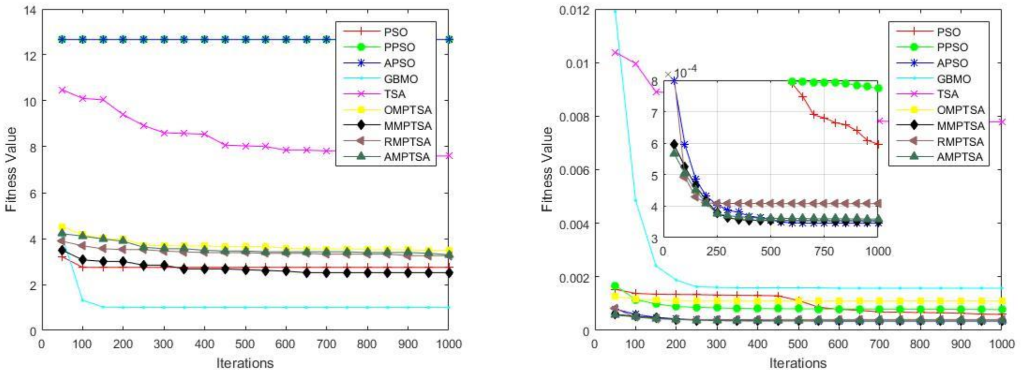
(c) F_{10} (d) F_{11}

(e) F_{12} (f) F_{13}

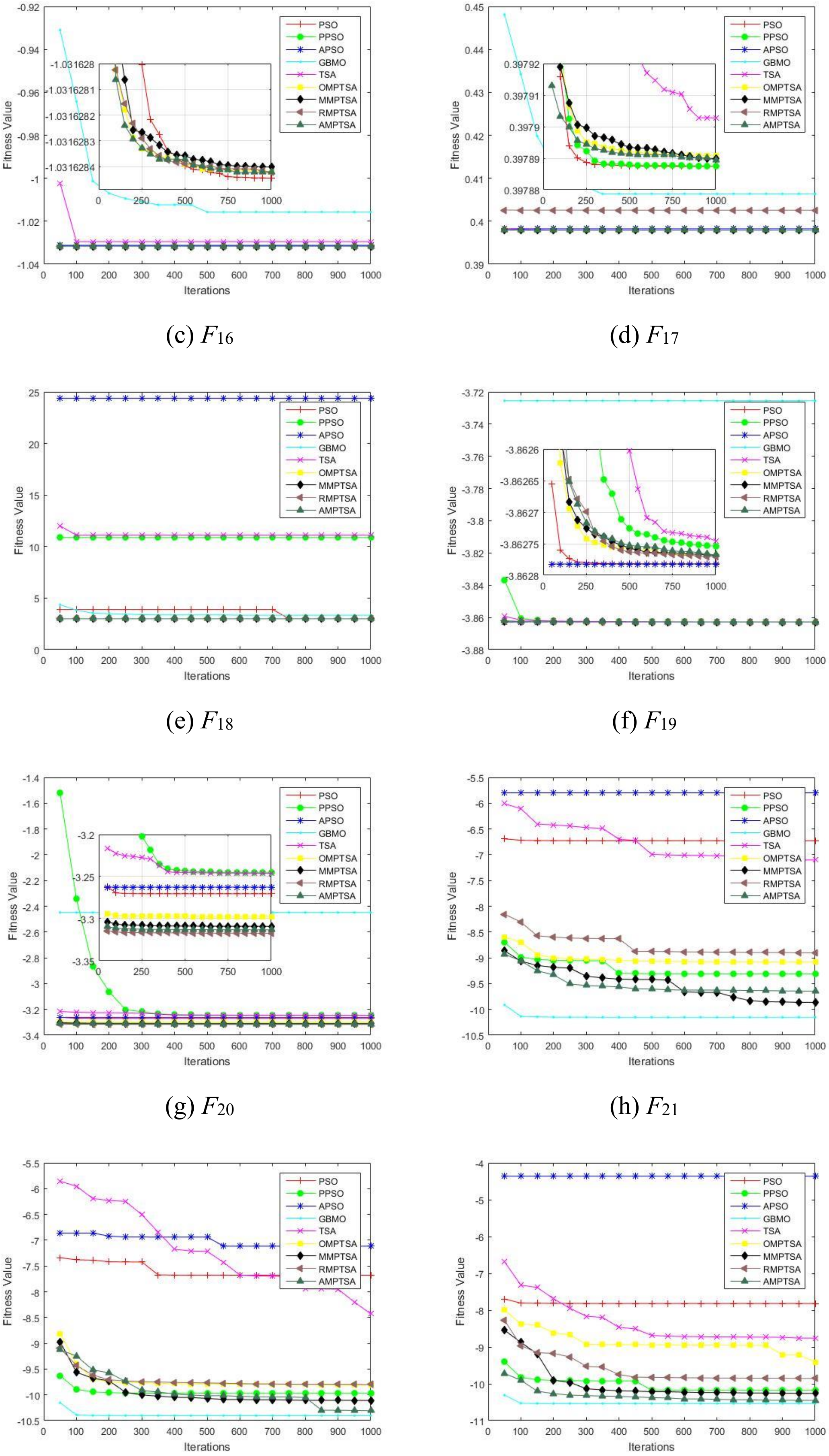
Figure 7. Convergence tendency for multimodal benchmark functions

4.2.3 Fixed-Dimension Multimodal Benchmark Functions Analysis

Figure 8 shows the test results of the algorithms in the Fixed-dimension multimodal benchmark functions. Fixed-dimension multimodal benchmark functions have only a few local minima and are not very high in dimension. In F14 and F15, MMPTSA produces good test results, although other MPTSA family algorithms do not perform particularly well. The test results of F16, F17 and F19 show little difference among all the algorithms. However, in F22 and F23, the performance of AMPTSA was slightly different from that of GBMO, but it outperformed other algorithms. In addition, MPTSA series algorithms are superior to other algorithms. Through the above experimental analysis, the optimization ability of the proposed MPTSA is confirmed.



(a) F_{14} (b) F_{15}



(c) F_{16}

(d) F_{17}

(e) F_{18}

(f) F_{19}

(g) F_{20}

(h) F_{21}

(i) F_{22}

(j) F_{23}

Figure 8. Convergence tendency for fixed-dimension multimodal benchmark functions

Table 3. The results of simulation experiment

Function	PSO		PPSO		APSO		GBMO		TSA		OMPTSA	
	AVG.	VARP.	AVG.	VARP.	AVG.	VARP.	AVG.	VARP.	AVG.	STSD.	AVG.	STSD.
F1	2.68E+00	6.62E-01	1.87E-02	8.84E-05	1.49E-03	3.37E-06	1.25E-03	5.16E-06	4.11E-50	1.28E-98	4.36E-54	2.12E-106
F2	1.11E+00	9.88E-02	1.05E+01	1.29E+03	7.79E-02	6.65E-04	4.78E-02	2.34E-03	6.41E-30	1.97E-58	1.67E-32	7.51E-64
F3	1.49E+02	1.99E+03	2.64E+00	2.47E+00	4.84E+00	8.31E+00	4.95E+00	1.07E+02	2.84E-13	1.25E-24	4.42E-19	1.51E-36
F4	3.21E+00	1.35E+00	1.41E+00	1.73E+00	6.44E-01	2.45E-01	1.79E-03	3.88E-06	1.40E-03	5.42E-06	8.21E-07	6.51E-12
F5	2.11E+02	3.38E+04	1.47E+02	1.50E+04	5.46E+01	1.24E+03	5.94E-02	1.14E-02	2.83E+01	5.33E-01	2.81E+01	7.71E-01
F6	2.50E+00	8.15E-01	3.22E-01	9.76E-01	3.45E-02	1.32E-02	5.49E-04	9.02E-07	3.60E+00	4.69E-01	3.19E+00	3.41E-01
F7	1.23E-02	2.17E-05	1.25E-01	1.03E-02	2.30E-01	1.15E-02	5.37E-03	2.07E-05	3.76E-03	3.51E-06	1.81E-03	5.11E-07
F8	-6.46E+03	5.14E+05	-1.18E+02	1.79E-05	-1.18E+02	1.76E-07	-1.17E+04	1.40E+04	-6.39E+03	3.84E+05	-6.82E+03	2.58E+05
F9	2.85E+01	1.04E+02	1.41E+02	3.07E+03	4.52E-01	1.70E-01	3.26E-03	3.25E-05	1.69E+02	1.68E+03	1.22E+02	1.03E+03
F10	2.20E+00	2.12E-01	1.18E+00	1.41E+00	2.64E-02	1.53E-04	1.66E+00	1.88E+00	1.86E+00	2.35E+00	7.77E-01	1.67E+00
F11	9.85E-01	3.30E-03	4.61E-02	5.15E-03	3.17E-02	2.91E-03	3.79E-03	6.45E-06	6.39E-03	1.51E-04	4.31E-03	4.39E-05
F12	2.40E+00	2.22E+00	5.50E+00	6.57E+00	1.96E-02	5.66E-03	4.44E-06	5.58E-11	8.39E+00	1.21E+01	3.41E+00	6.96E+00
F13	3.42E-01	2.29E-02	1.04E-01	3.99E-02	9.76E-04	8.09E-06	3.26E-05	1.98E-09	2.68E+00	3.93E-01	2.49E+00	2.76E-01
F14	2.74E+00	4.30E+00	1.27E+01	2.23E-26	1.27E+01	1.47E-24	9.98E-01	1.91E-20	7.60E+00	2.58E+01	3.48E+00	1.27E+01
F15	5.97E-04	2.38E-07	7.74E-04	1.40E-08	3.46E-04	2.05E-08	1.57E-03	4.46E-08	7.79E-03	1.58E-04	1.10E-03	1.29E-05
F16	-1.03E+00	2.37E-16	-1.03E+00	4.13E-09	-1.03E+00	2.45E-06	-1.02E+00	6.31E-04	-1.03E+00	6.22E-05	-1.03E+00	1.67E-15
F17	3.98E-01	3.66E-15	3.98E-01	1.20E-13	3.98E-01	3.75E-06	4.06E-01	6.07E-05	3.98E-01	3.38E-10	3.98E-01	3.11E-11
F18	3.00E+00	3.29E-14	1.09E+01	3.14E+02	2.44E+01	9.92E+02	3.35E+00	8.16E-02	1.11E+01	4.45E+02	3.00E+00	5.82E-14
F19	-3.86E+00	4.69E-15	-3.86E+00	1.48E-09	-3.86E+00	7.10E-30	-3.73E+00	1.42E-02	-3.86E+00	5.97E-10	-3.86E+00	4.04E-11
F20	-3.27E+00	3.47E-03	-3.25E+00	3.40E-03	-3.26E+00	3.53E-03	-2.45E+00	1.22E-01	-3.25E+00	1.07E-02	-3.30E+00	2.25E-03
F21	-6.73E+00	1.23E+01	-9.31E+00	3.55E+00	-5.80E+00	9.15E+00	-1.02E+01	1.06E-09	-7.10E+00	1.02E+01	-9.08E+00	4.07E+00
F22	-7.68E+00	1.30E+01	-9.97E+00	2.69E+00	-7.12E+00	1.15E+01	-1.04E+01	2.12E-09	-8.43E+00	1.04E+01	-9.81E+00	2.50E+00
F23	-7.82E+00	1.31E+01	-1.02E+01	1.79E+00	-4.36E+00	9.91E+00	-1.05E+01	4.89E-09	-8.76E+00	8.92E+00	-9.41E+00	5.26E+00
WIN	13		14		15		12		21			
LOSE	4		3		2		0		2			
SIMILAR	6		6		6		11		0			

Table 4. The T-test results of the compared algorithms (OMPTSA, MMPTSA, RMPTSA and AMPTSA) on TSA

Function	OMPTSA	MMPTSA	RMPTSA	AMPTSA
F1	+	+	+	+
F2	+	+	+	+
F3	=	=	=	=
F4	+	+	+	+
F5	=	+	=	+
F6	+	+	+	+
F7	+	+	+	+
F8	+	+	+	+
F9	+	+	+	+
F10	+	+	=	+
F11	=	+	=	+
F12	+	+	=	+
F13	=	+	+	+
F14	+	+	+	+
F15	+	+	+	+
F16	=	=	=	=
F17	=	=	=	=
F18	+	+	+	+
F19	=	=	=	=
F20	+	+	+	+
F21	+	+	+	+
F22	+	+	+	+
F23	+	+	+	+

5 The proposed MPTSA is Applied to the Simulation Experiment of WSNs 3D Node Deployment

5.1 Coverage Optimization Model for WSNs

The coverage problem of WSNs is usually closely related to the perception model of each node and the location deployment of all nodes [54-55]. In short, the sensor node perception model constructs the geometric relationship between the physical location and the spatial location of the sensor node, which can be used as a measure of the service quality of the sensor sensing function. In theoretical research, common sensor node perception models include: 0-1 model, exponential model, statistical model, obstacle model, etc. In this paper, 0-1 model is used to optimize the coverage of WSNs. Therefore, we will introduce the 0-1 model in detail.

0-1 model: Generally speaking, the perception model of sensor nodes is usually simplified to 0-1 model, that is, a certain point in the region is covered (1) or not covered (0) by sensor nodes. In the related literature, the most commonly used 0-1 sensing model is the perception disk model. All points in the range of a certain sensor node as the center and a fixed length r as the radius are considered to be able to be covered by the sensor node. Suppose that the coordinate of node i in the detected area is $\left(x_i, y_i, z_i\right)$, the sensing radius of the node is r_i, and the coordinate of the target node j is $\left(x_j, y_j, z_j\right)$, then the distance between node i and target node j is :

$$d_{i,j}=\sqrt{\left(x_i-x_j\right)^2+\left(y_i-y_j\right)^2+\left(z_i-z_j\right)^2} \quad \textbf{(17)}$$

The perception quality of node i to node j is expressed by $o_{i,j}$. When the position of the concerned node j is in the circle of the sensing range r_i of node i, the perceived quality of node i to node j is 1, that is, the perceived degree of node i to node j is 1; otherwise, when node j is outside the sensing range of node i, the perceived degree of node i to node j is 0, so the mathematical expression is :

$$o_{i,j}=\begin{cases}1, & d_{i,j}\le r_i \\ 0, & otherwise\end{cases} \quad \textbf{(18)}$$

In the early research of WSNs coverage, the node perception model is generally 0-1 model, which ignores the influence of some external factors, making the problem simpler and more convenient for people to study the problem easily.

This paper aims to solve the problem of maximum coverage when the monitored area is fixed and the sensor nodes are limited. There are many ways to solve the problem of 2D plane coverage, and to achieved good performance. However, if the sensor nodes are placed on the 2D plane, the simulation experiment will be obviously different from the actual situation. In this paper, the sensor nodes are placed on the 3D terrain to simulate the actual coverage problem more realistically.

One essential procedure to solve the coverage problem is to find the optimal deployment strategy. The different strategies have a significant influence on coverage rate especially in 3D terrain [56]. In recent years, many researchers utilized intelligent computing algorithm to settle similar problems.

Sensor nodes are set on the ground, so only two coordinate values of a point are needed to calculate the coordinates of the point. Therefore, the algorithm can optimize the deployment strategy by optimizing any 2D sensor node location. Each individual of the algorithm represents a deployment strategy. Each individual updates their own location and calculates the fitness function value according to equation (19).

$$R(t)=\frac{1}{M}\sum_{j=1}^{M}\left(\sum_{i=1}^{P}o_{i,j}\right) \tag{19}$$

Where $R(t)$ is the coverage rate of the t-th iteration, P represents the number of sensor nodes, M is the number of pixels on the 3D terrain, $o_{i,j}$ indicates whether the pixel j is covered by node i, which is obtained according to equation (18).

5.2 Simulation Experiment of MPTSA Optimizing WSNs Coverage Model

In this part, we apply the proposed MPTSA series algorithms to the WSNs node deployment problem in 3D space. 2D planar node deployment is already insufficient to meet the requirements of today's society. Therefore, this simulation experiment simulated some mountains and low-lying terrain in an area of $50m\times 50m$. On the basis of 2D, this experiment also considered the height information of mountain and other obstacles. Next, according to the different number of sensor nodes and communication radius, the simulation experiments are carried out respectively. In order to ensure fairness, this experiment will make the communication radius of the nodes the same when the sensor nodes are set to be different. And in order to verify the superiority of the proposed method, this experiment will use the comparison algorithm in the previous paper to carry out comparative experiments. The population number of each algorithm pop will be set to 40. The maximum number of iterations $iter_{\max}$ will be set to 10. Among them, PPSO and MPTSA series algorithms using the idea of parallelism are used for grouping, and the population will be divided into 4 groups, and information exchange will be conducted for 3 times per iteration. Other parameters will remain the same as those in the previous section. The specific experimental parameters are shown in Table 5.

Table 5. Experimental settings for parameters

Parameter	Parameter Values
Sensing region area	50m × 50m
pop	40
$iter_{\max}$	10
Number of sensor nodes	30-55
Communication range	5m-10m
Number of groups	4

5.2.1 Set Different Number of Nodes

In this part of the experiment, we will set different number of sensor nodes to verify the effectiveness of the proposed method in this case. Set the number of sensor nodes to $\{30,35,...,50\}$. In order to ensure fairness, the communication radius of the sensor node will be set at 5 meters in this experiment. Table 6 shows the experimental results obtained by using nine methods when the number of nodes is different. Figure 9 shows the experimental results in this case as a line graph. It can be seen that when the number of sensor nodes increases, the coverage of WSNs is also gradually increasing. For the convenience of observation, Figure 9 enlarges the result when the number of nodes is set to 50. It can be seen from this that when there are 50 nodes in the network, MMPTSA, which uses the median to communicate, has a very good performance in optimizing network coverage. Of the nine methods compared, it ranked first. AMTPSA came in second, but only 0.31 percent off the best approach. This suggests that MMPTSA and AMPTSA are better at solving this problem than other approaches. While the other two methods in the MPTSA series of algorithms, although not the best, but compared with other methods also perform a little stronger. It is worth mentioning that the traditional PSO algorithm performed the worst in this experiment, with the coverage rate only reaching 63.52%. The method proposed in this paper reaches 68.15%. In the experiment to test the algorithm performance, GBMO, which has a relatively moderate performance, only ranks third from the bottom in the experimental results of this experiment. It can be seen that GBMO is limited in its ability to solve the problem of network coverage.

Table 6. Coverage of different number of nodes

NodesNum	PSO	PPSO	APSO	GBMO	TSA	OMPTSA	MMPTSA	RMPTSA	AMPTSA
30	0.4605	0.4672	0.4693	0.4629	0.4637	0.4888	0.5060	0.4895	0.4952
35	05073	0.5198	0.5233	0.5097	0.5177	0.5329	0.5586	0.5411	0.5432
40	0.5533	0.5629	0.5638	0.5583	0.5617	0.5832	0.6046	0.5878	0.5939
45	0.5951	0.6032	0.6097	0.6001	0.6012	0.6229	0.6475	0.6294	0.6354
50	0.6352	0.6402	0.6483	0.6409	0.6392	0.6624	0.6815	0.6661	0.6784
55	0.6649	0.6735	0.6749	0.6706	0.6714	0.6923	0.7758	0.7016	0.7612

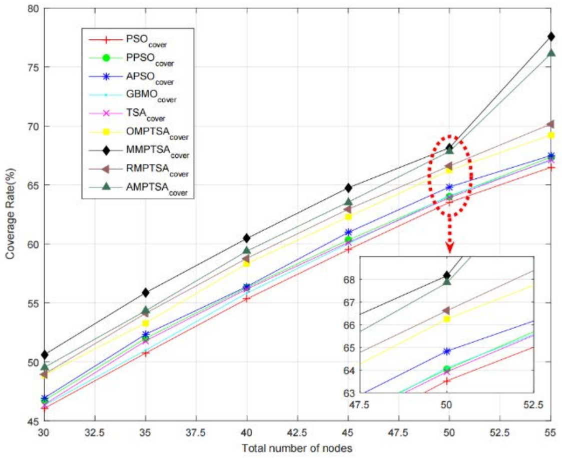


Figure 9. Experimental results of different number of nodes

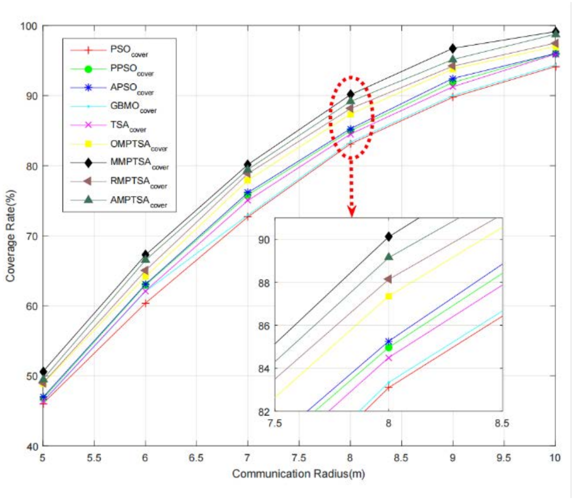


Figure 10. Experimental results of different communication radius

5.2.2 Set Different Communication Radius of Nodes

In this part, the communication radius of the node is set to different values for experiment. The communication radius will be set to $5m, 6m, ..., 10m$. Similarly, in order to ensure the fairness of the experiment, the number of sensor nodes used is 30 in all the methods. Table 7 shows the coverage comparison results obtained by the nine methods when the communication radius of the nodes are different. Figure 10 shows the experimental results in the form of line graph, and in order to see the difference of each method more clearly, the data when the communication radius is 8m is amplified. It can be seen from the experimental results that the communication radius has a great influence on the coverage rate. The same PSO method was used in the experiment. When the communication radius was 5m, the coverage rate was only 46.05%. When the communication radius is increased to 10m, the coverage rate can reach 94.15%. After comparing the experimental results of the nine methods used, it is found that MMPTSA is still the most powerful, with the result of 99.13%. As in the previous experiment, AMPTSA came in second, but only 0.37% lower than MMPTSA. The other methods of the proposed MPTSA series of algorithms still show better performance than other contrast algorithms in this part of the experiment. For example, OMPTSA, which has the worst performance, gets only 97.03%, but it is still higher than any other method except MPTSA family of algorithms. Thus it can be seen that the MPTSA series algorithm proposed in this paper has a strong effectiveness and superiority in solving the problem of 3D node deployment of WSNs.

Table 7. Coverage of different communication radius

NodeNum	PSO	PPSO	APSO	GBMO	TSA	OMPTSA	MMPTSA	RMPTSA	AMPTSA
5m	0.4605	0.4672	0.4693	0.4629	0.4637	0.4888	0.5060	0.4895	0.4952
6m	0.6037	0.6294	0.6305	0.6203	0.6211	0.6415	0.6724	0.6502	0.6651
7m	0.727	0.7582	0.7615	0.7294	0.7502	0.7792	0.8015	0.7886	0.7946
8m	0.831	0.8497	0.8526	0.8334	0.845	0.8736	0.9012	0.8814	0.8915
9m	0.8976	0.9188	0.9245	0.9	0.9126	0.9377	0.9675	0.942	0.9515
10m	0.9415	0.9599	0.9601	0.9437	0.9591	0.9703	0.9913	0.975	0.9876

6 Conclusion

In the research of evolutionary algorithms, how to solve some problems of the traditional algorithm is the most concerned topic of researchers, such as poor convergence, easy to fall into local optimum, and so on. TSA mentioned in this paper is no exception. Therefore, this paper proposes a solution to the problem of TSA, and proposes four different communication strategies for this method. The experimental results of 23 different types of test functions show that the performance of the proposed MPTSA series algorithm is better than other comparison algorithms. In the research of WSNs, it is very important to make the sensor nodes layout efficiently and maximize the coverage under limited conditions. In order to solve this problem and improve the network coverage, this paper applies the proposed MPTSA series algorithm to the research of 3D coverage of WSNs, and further proves the applicability of the algorithm in this field. But in the process of doing experiments, we found that MPTSA series algorithm still has some problems such as slow running speed. In future research, we will further improve the shortcomings of the algorithm.

References

[1] J. S. Pan, Z. Y. Meng, S. C. Chu, H. R. Xu, Monkey King Evolution: an enhanced ebb-tide-fish algorithm for global optimization and its application in vehicle navigation under wireless sensor network environment, *Telecommunication Systems*, Vol. 65, No. 3, pp. 351-364, July, 2017.

[2] S. C. Chu, J. F. Roddick, C. J. Su, J. S. Pan, Constrained Ant Colony Optimization for Data Clustering, *Pacific Rim International Conference on Artificial Intelligence*, Auckland, New Zealand, 2004, pp. 534-543.

[3] J. W. Zhuang, H. Luo, T. S. Pan, J. S. Pan, Improved flower pollination algorithm for the capacitated vehicle routing problem, *Journal of Network Intelligence*, Vol. 5, No. 3, pp. 141-156, August, 2020.

[4] C. Sun, Y. Jin, R. Cheng, J. Ding, J. Zeng, Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems, *IEEE Transactions on Evolutionary Computation*, Vol. 21, No. 4, pp. 644-660, August, 2017.

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

[6] B. Brandstatter, U. Baumgartner, Particle Swarm Optimization - Mass-Spring System Analogon, *IEEE Transactions on Magnetics*, Vol. 38, No. 2, pp. 997-1000, March, 2002.

[7] M. Abdechiri, M. R. Meybodi, H. Bahrami, Gases Brownian Motion Optimization: an Algorithm for Optimization (GBMO), *Applied Soft Computing Journal*, Vol. 13, No. 5, pp. 2932-2946, May, 2013.

[8] Z. Y. Meng, J. S. Pan, QUasi-affine TRansformation Evolutionary (QUATRE) algorithm: The framework analysis for global optimization and application in hand gesture segmentation, *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, Chengdu, China, 2016, pp. 1832-1837.

[9] C. L. Sun, J. Zeng, J. S. Pan, S. Xue, Y. Jin, A new fitness estimation strategy for particle swarm optimization, *Information Sciences*, Vol. 221, pp. 355-370, February, 2013.

[10] H. Wang, H. Sun, C. Li, S. Rahnamayan, J. S. Pan, Diversity enhanced particle swarm optimization with neighborhood search, *Information Sciences*, Vol. 223, pp. 119-135, February, 2013.

[11] J. S. Pan, Z. Meng, H. Xu, X. Li, Quasi-affine transformation evolution (quatre) algorithm: A new simple and accurate structure for global optimization, *IEA/AIE*, Morioka, Japan, 2016, pp. 657-667.

[12] Z. Y. Sun, D. N. Yang, D2D radio resource allocation algorithm based on global fairness, *Journal of Northeast Electric Power University*, Vol. 39, No. 1, pp. 81-87, February, 2019.

[13] S. C. Chu, X. Xue, J. S. Pan, X. Wu, Optimizing ontology alignment in vector space, *Journal of Internet Technology*, Vol. 21, No. 1, pp. 15-22, January, 2020.

[14] C. H. Chen, An arrival time prediction method for bus system, *IEEE Internet of Things Journal*, Vol. 5, No. 5, pp. 4231-4232, October, 2018.

[15] X. Xue, J. F. Chen, J. S. Pan, *Evolutionary Algorithm based Ontology Matching Technique*, Science Press, Bejing, 2018.

[16] P. C. Song, J. S. Pan, S. C. Chu, A parallel compact cuckoo search algorithm for three-dimensional path planning, *Applied Soft Computing*, Vol. 94, Article No. 106443, September, 2020.

[17] A. Q. Tian, S. C. Chu, J. S. Pan, H. Cui, W. M. Zheng, A compact pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station, *Sustainability*, Vol. 12, No. 3, Article No. 767, February, 2020.

[18] S. C. Chu, H. C. Huang, J. F. Roddick, J. S. Pan, Overview of algorithms for swarm intelligence, *International Conference on Computational Collective Intelligence*, 2011, Gdynia, Poland, pp. 28-41.

[19] J. S. Pan, X. Wang, S. C. Chu, T. Nguyen, A multi-group grasshopper optimisation algorithm for application in capacitated vehicle routing problem, *Data Science* and *Pattern Recognition*, Vol. 4, No. 1, pp. 41-56, July, 2020.

[20] X. S. Xue, J. S. Pan, A Compact Co-Evolutionary Algorithm for sensor ontology meta-matching, *Knowledge and Information Systems*, Vol. 56, No. 2, pp. 335-353, August, 2018.

[21] J. S. Pan, P. Hu, S. C. Chu, Novel parallel heterogeneous meta-heuristic and its communication strategies for the prediction of wind power, *Processes*, Vol. 7, No. 11, Article No. 845, November, 2019.

[22] S. Kaur, L. K. Awasthi, A. L. Sangal, G. Dhiman, Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization, *Engineering Applications of Artificial Intelligence*, Vol. 90, pp. 103541.1-103541.29, April, 2020.

[23] E. H. Houssein, E. D. Helmy, A. A. Elngar, D. S. Abdelminaam, H. Shaban, An improved tunicate swarm algorithm for global optimization and image segmentation, *IEEE Access*, Vol. 9, pp. 56066-56092, April, 2021.

[24] H. Yu, Y. Tan, C. Sun, J. Zeng, A generation-based optimal restart strategy for surrogate-assisted social learning particle swarm optimization, *Knowledge-Based Systems*, Vol. 163, pp. 14-25, January, 2019.

[25] P. Hu, J. S. Pan, S. C. Chu, Improved binary grey wolf optimizer and its application for feature selection, *Knowledge-Based Systems*, Vol. 195, Article No. 105746, May, 2020.

[26] J. S. Pan, P. C. Song, S. C. Chu, Y. J. Peng, Improved compact cuckoo search algorithm applied to location of drone logistics hub, *Mathematics*, Vol. 8, No. 3, Article No. 333, March, 2020.

[27] H. Wang, G. Zhang, E. Mingjie, N. Sun, A novel intrusion detection method based on improved SVM by combining PCA and PSO, *Wuhan University Journal of Natural Sciences*, Vol. 16, No. 5, Article No. 409, October, 2011.

[28] J. S. Pan, L. Kong, T. W. Sung, P. W. Tsai, V. Snášel, α-fraction first strategy for hierarchical model in wireless sensor networks, *Journal of Internet Technology*, Vol. 19, No. 6, pp. 1717-1726, November, 2018.

[29] C. M. Wu, H. L. Shang, QoS-aware resource allocation for D2D communications, *Journal of Northeast Electric Power University*, Vol. 40, No. 2, pp. 89-95, April, 2020.

[30] J. Zhang, P. Cao, Linear closed-form estimator for sensor localisation using RSS and AOA measurements, *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 31, No. 3, pp. 178-188, 2019.

[31] Y. F. Huang, C. H. Hsu, J. W. Wang, H. L. Hung, J. Jenq, Energy efficiency of dynamically distributed clustering routing for naturally scattering wireless sensor networks, *Journal of Network Intelligence*, Vol. 3, No. 1, pp. 50-57, February, 2018.

[32] J. S. Pan, T. T. Nguyen, T. K. Dao, T. S. Pan, S. C. Chu, Clustering formation in wireless sensor networks: A survey, *Journal of Network Intelligence*, Vol. 2, No. 4, pp. 287-309, November, 2017.

[33] J. Wang, Y. Gao, K. Wang, A. K. Sangaiah, S. J. Lim, An affinity propagation-based self-adaptive clustering method for wireless sensor networks, *Sensors*, Vol. 19, No. 11, Article No. 2579, June, 2019.

[34] J. S. Pan, T. T. Nguyen, S. C. Chu, T. K. Dao, T. G. Ngo, Diversity enhanced ion motion optimization for localization in wireless sensor network, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 10, No. 1, pp. 221-229, January, 2019.

[35] H. B. Li, S. N. Qi, C. M. Wu, Forest fire positioning monitoring system based on wireless sensor networks, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 9, No. 4, pp. 970-976, July, 2018.

[36] X. X. Dong, Y. Zhang, S. J. Yu, An uneven clustering routing protocol based on improved k-means algorithm for wireless sensor network in coal-mine, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 10, No. 1, pp. 53-62, January, 2019.

[37] T. G. Nguyen, C. So-In, N. G. Nguyen, Barrier Coverage Deployment Algorithms for Mobile Sensor Networks, *Journal of Internet Technology*, Vol. 18, No. 7, pp. 1689-1699, December, 2017.

[38] J. P. Li, G. Z. Lu, Dynamic cooperative localization algorithm for wireless sensor networks, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 9, No. 4, pp. 949-958, July, 2018.

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

[40] N. Liu, J. S. Pan, J. Wang, T. T. Nguyen, An adaptation multi-group quasi-affine transformation evolutionary algorithm for global optimization and its application in node localization in wireless sensor networks, *Sensors*, Vol. 19, No. 19, Article No. 4112, October, 2019.

[41] C. I. Wu, H. Y. Kung, C. H. Chen, L. C. Kuo, An intelligent slope disaster prediction and monitoring system based on WSN and ANP, *Expert Systems with Applications*, Vol. 41, No. 10, pp. 4554-4562, August, 2014.

[42] S. C. Chu, Z. G. Du, J. S. Pan, Symbiotic organism search algorithm with multi-group quantum-behavior communication scheme applied in wireless sensor networks, *Applied Sciences*, Vol. 10, No. 3, Article No. 930, February, 2020.

[43] T. Nguyen, J. Pan, J. Lin, T. Dao, T. Nguyen, An optimal node coverage in wireless sensor network based on whale optimization algorithm, *Data Science and Pattern Recognition*, Vol. 2, No. 2, pp. 11-21, December, 2018.

[44] J. S. Pan, L. Kong, T. W. Sung, P. W. Tsai, V. Snášel, A clustering scheme for wireless sensor networks based on genetic algorithm and dominating set, *Journal of Internet Technology*, Vol. 19, No. 4, pp. 1111-1118, July, 2018.

[45] Z. Khalfallah, I. Fajjari, N. Aitsaadi, P. Rubin, G. Pujolle, A novel 3D underwater WSN deployment strategy for full-coverage and connectivity in rivers, *ICC 2016 - 2016 IEEE International Conference on Communications*, Kuala Lumpur, Malaysia, 2016, pp. 1-7.

[46] B. Cao, J. W. Zhao, Z. H. Lv, X. Liu, X. Kang, S. Yang, Deployment optimization for 3D industrial wireless sensor networks based on particle swarm optimizers with distributed parallelism, *Journal of Network and Computer Applications*, Vol. 103, pp. 225-238, February, 2018.

[47] V. Nazarzehi, A. V. Savkin, Decentralized control of mobile three-dimensional sensor networks for complete coverage self-deployment and forming specific shapes, *IEEE Conference on Control Applications*, Sydney, New South Wales, 2015, pp. 127-132.

[48] N. Anand, R. Ranjan, B. S. Rai, S. Varma, A novel computational geometry-based node deployment scheme in 3D wireless sensor network, *International Journal of Sensor Networks*, Vol. 25, No. 3, pp. 135-145, January, 2017.

[49] N. Boufares, Y. B. Saied, L. A. Saidane, Improved Distributed Virtual Forces Algorithm for 3D Terrains Coverage in Mobile Wireless Sensor Networks, *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, Aqaba, Jordan, 2018, pp. 1-8.

[50] T. Fetouh, A. M. Elsayed, Optimal control and operation of fully automated distribution networks using improved tunicate swarm intelligent algorithm, *IEEE Access*, Vol. 8, pp. 129689-129708, July, 2020.

[51] Q. W. Chai, S. C. Chu, J. S. Pan, W. M. Zheng, Applying adaptive and self assessment fish migration optimization on localization of wireless sensor network on 3-d terrain, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 11, No. 2, pp. 90-102, June, 2020.

[52] Q. W. Chai, S. C. Chu, J. S. Pan, P. Hu, W. M. Zheng, A parallel woa with two communication strategies applied in dv-hop localization method, *EURASIP Journal on Wireless Communications and Networking*, Vol. 2020, pp. 1-10, February, 2020.

[53] X. Sui, S. C. Chu, J. S. Pan, H. Luo, Parallel compact differential evolution for optimization applied to image segmentation, *Applied Sciences*, Vol. 10, No. 6, Article No. 2195, March, 2020.

[54] J. P. Li, Q. H. Zhang, Z. T. Zhang, Y. Q. Yin, H. J. Zhang, Congestion control and energy optimization routing algorithm for wireless sensor networks, *Journal of Northeast Electric Power University*, Vol. 40, No. 4, pp. 69-74, August, 2020.

[55] J. S. Pan, Q. W. Chai, S. C. Chu, N. Wu, 3-d terrain node coverage of wireless sensor network using enhanced black hole algorithm, *Sensor (Basel, Switzerland)*, Vol. 20, No. 8, Article No. 2411, April, 2020.

[56] M. Gao, J. S. Pan, J. P. Li, Z. P. Zhang, Q. W. Chai, 3-D Terrains Deployment of Wireless Sensors Network by Utilizing Parallel Gases Brownian Motion Optimization, *Journal of Internet Technology*, Vol. 22, No. 1, pp. 13-29, January, 2021.

Biographies

Jianpo Li received his Ph.D. from the Department of Communication Engineering, Jilin University, China. In 2008, he joined the Northeast Electric Power University (NEEPU). He was a visiting scholar with New York University and University of Ottawa. Now he is a full professor and Vice-dean of the School of Computer Science, NEEPU. His research interests focus on wireless sensor networks.

Geng-Chen Li received his B.S. degree from Northeast Electric Power University, China, in 2019. He is currently pursuing the master degree with the Northeast Electric Power University, Jinlin, China. His research interests include intelligence algorithms and wireless sensor network.

Shu-Chuan Chu received a Ph.D. degree in 2004 from the School of Computer Science, Engineering and Mathematics, Flinders University of South Australia. She joined Flinders University in December 2009 after 9 years at the Cheng Shiu University, Taiwan. Currently, she is the Research Fellow with a Ph.D. supervisor in the College of Computer Science and Engineering of Shandong University of Science and Technology from September 2019. Her research interests are mainly in Swarm Intelligence, Intelligent Computing and Data Mining.

Min Gao received her B.S. degree from Nankai University Binhai College, China, in 2019. She is currently pursuing the master degree with the Northeast Electric Power University, Jinlin, China. Her research interests include intelligence algorithms and wireless sensor network.

Jeng-Shyang Pan received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology in 1986, the M.S. degree in communication engineering from National Chiao Tung University, Taiwan, in 1988, and the Ph.D. degree in electrical engineering from the University of

Edinburgh, U.K., in 1996. He is currently the Professor in Shandong University of Science and Technology.