Research on Wireless Sensor Network Technology Based on Particle Swarm Algorithm

Guanghui Wang
Department of network management, Noncommissioned Officer Academy of PAP, Zhejiang Hangzhou 310012, China
wangguanghui@rtcsc.com

Abstract. The paper proposes a positioning method based on random particle swarm optimization algorithm. It is assumed that there are some anchor nodes in the wireless sensor network, and the distance information between adjacent nodes can be obtained. After the positioning node obtains enough distance and location information of the adjacent anchor nodes or the located nodes, the random particle swarm optimization algorithm is used Achieve positioning. Simulation shows that this method has higher performance than multilateral measurement method and positioning method based on standard particle swarm optimization algorithm.

Keywords: wireless sensor network; positioning technology; ranging; random particle swarm optimization

1. Introduction
The wireless sensor network consists of several nodes with wireless communication capabilities. These nodes are densely arranged in the physical phenomenon area, forming a wireless network in a self-organizing and multi-hop manner, and cooperatively monitor, sense and collect information such as images, temperature, and humidity in the network coverage area in real time and send it to users wirelessly.

Many mechanisms such as sensor network coverage, data perception, and data processing are involved in the collection and transmission process, and the first step is to cover the target area. At present, the deployment of nodes in wireless sensor networks mainly includes the following methods: deterministic deployment, random deployment and mobile deployment. The deterministic deployment method refers to the deployment of nodes one by one after designing the deployment location of nodes in advance. Random deployment refers to the deployment of sensor nodes in a scattered way. Removable deployment means that some or all of the nodes in the network can move according to a certain strategy after deployment. Since random deployment has the advantages of low price and easy implementation, most of the applications of wireless sensors use random deployment. The optimization of wireless sensor networks using random deployment methods is to reduce coverage blind areas, reduce the number of nodes, and improve network coverage [1]. These problems are all NP problems, and it is difficult to get a better solution with traditional optimization methods. The standard PSO algorithm cannot guarantee global convergence, and the convergence speed is slow. Therefore, this paper proposes a distributed
node positioning method using random particle swarm optimization (SPSO) in order to obtain better positioning performance.

2. Overview of wireless sensor networks

2.1. Overview of wireless sensor networks
Wireless sensor network (WSN) involves sensing technology, network communication technology, wireless transmission technology, embedded technology, distributed information processing technology, microelectronic manufacturing technology, software programming technology, etc. It is a highly interdisciplinary, emerging and cutting-edge multidisciplinary Hot research areas. It is one of the IT technologies that will have a significant impact on human life style in the 21st century after the Internet. The wireless sensor network merges the logical information world with the objective physical world, and changes the way of exchange between man and nature [2]. In the future, people will directly perceive the objective world through a network of sensors all around, thereby greatly expanding the functions of the network and the ability of humans to understand the world. A typical wireless sensor network system architecture is shown in Figure 1.

![Fig.1 System architecture of wireless sensor network](image)

In Figure 1, a large number of sensor nodes are randomly densely distributed in the entire observation area, forming a network through self-organization. After the sensor node performs preliminary processing on the detected information, it transmits it to the sink node in a multi-hop manner, and then reaches the management node where the end user is located via satellite, Internet or mobile communication network. End users can also manage and configure wireless sensor networks, issue monitoring tasks, or collect and return data through management nodes.

2.2. Energy management of wireless sensor network
The sensor node is composed of four parts: computing subsystem, communication subsystem, sensor subsystem and energy supply subsystem. The energy consumption of sensor nodes is mainly divided into the following aspects:

2.2.1. Computing Subsystem
The choice of the microprocessor (MCU) will have a great impact on the battery consumption of the node. For example, Intel’s Strong ARM is often used in high-end fields and consumes 400mW when executing instructions, while the power consumption of ATmega 103 LAVR is only 16.5mW. But the performance provided is much weaker. MCU usually has multiple operation modes such as active, idle and sleep, and each mode has different power consumption. For example, Strong ARM consumes 50mW...
in idle mode, but only 0.16mW in sleep mode. Switching between different operating modes also has power and delay overhead [3]. Therefore, different operating modes, switching between modes, and the duration of the MCU in each mode have a great impact on the energy consumption of the entire node.

2.2.2. Communication subsystem
There are many factors that affect the power consumption of the wireless transceiver circuit, including the modulation mode, data rate, transmission power, and operating cycle adopted by the node. Generally, the wireless transceiver circuit can work in four states, namely sending, receiving, idle and sleep states. The idle state also has a high-power consumption, which is almost on par with the receiving mode, so when the wireless transceiver circuit is in an idle state, it should be turned off as much as possible (that is, put into a sleep state).

2.2.3. Sensing subsystem
It includes a set of sensing and excitation devices that convert physical phenomena in the surrounding environment into electrical signals, which can be divided into analog and digital according to the output. In wireless sensors, energy consumption comes from multiple parts, including signal sampling and conversion of physical signals to electrical signals, signal modulation, and signal analog/digital conversion.

When a large number of sensor nodes form a sensor network, energy consumption is mainly consumed in the following aspects: (1) Communication energy consumption: users issue commands in the network and each sensor node sends back reports to the base station of the energy consumed by radio communication. (2) Sensing energy consumption: the energy consumed by sensors in the wireless sensor network when collecting signals. (3) Calculating energy consumption: the energy consumed by the microprocessor when the sensor performs simple local signal processing, and when multiple sensors perform coordinated signal processing.

3. Particle swarm optimization algorithm and its improvement

3.1. Implementation of particle swarm optimization algorithm

3.1.1. Basic principles
The PSO algorithm originated from the study of predation behavior of flocks of birds: it is assumed that a flock of birds is searching for food randomly. There is only one piece of food in this area, and all birds do not know where the food is located, but they rely on information exchanged with each other to approach the food step by step to find the food. PSO is derived from this model and used to solve optimization problems. In the PSO algorithm, the concepts of "group" and "evolution" are adopted, and operations are performed according to the size of individual fitness. It regards each candidate solution as a "particle" without weight and volume in an n-dimensional search space. Multiple particles coexist and cooperate to optimize. The algorithm first generates an initial population, that is, randomly initializes a group of particles in the feasible solution space, each particle is a feasible solution to the optimization problem, and an objective function determines a fitness value for it. Each particle will move in the solution space, and its direction and distance are determined by a speed [4]. Usually, the particles will follow the current optimal particle motion, and the optimal solution will be finally obtained after generational search. In each generation, the particle will track two extreme values, one is the optimal solution found so far by the particle itself, and the other is the optimal solution found so far by the entire population. The mathematical expression of the PSO algorithm is as follows:

Suppose a particle group containing m particles "flying" (searching) in D-dimensional space, the particle group can be represented by the following parameters:

\[ x_i = (x_{i1}, x_{i2}, ..., x_{id}) \] is the current position of particle i; \[ v_i = (v_{i1}, v_{i2}, ..., v_{id}) \] is the current flying speed of particle i; \[ P_i = (P_{i1}, P_{i2}, ..., P_{id}) \] is the historical optimal position of particle i, also called \( p_{best} \);
is the optimal position passed by all particles in the group, also called \( g_{\text{best}} \). The velocity and position of the particles in each dimension change according to the following formula:

\[
v_{id}(t+1) = v_{id}(t) + c_1 r_1 (p_{id}(t) - x_{id}(t)) + c_2 r_2 (g_{best} - x_{id}(t))
\]

(1)

\[
x_{id}(t+1) = x_{id}(t) + v_{id}(t+1)
\]

(2)

Among them, \( i = 1, 2, \ldots, m; d = 1, 2, \ldots, D \); \( c_1 \) is the own acceleration constant, \( c_2 \) is the global acceleration constant, usually a value between 0 and 2; \( r_1, r_2 \) is a random number on (0,1). If \( x_{id}, v_{id} \) exceeds the boundary value, replace \( x_{id}, v_{id} \) with the boundary value, which is a different constant according to different objective functions and different search spaces. The process of particle search for the optimal solution is shown in Figure 2:

![Fig.2 Schematic diagram of particle optimization search](image)

3.1.2. Algorithmic social behavior analysis

In the velocity evolution equation described by formula (1), the first part is the previous speed of the particle; the second part is the "cognition" part, because it only considers the particle's own experience and represents the particle's own thinking. If the speed evolution equation of the basic particle swarm algorithm only contains the cognitive part, namely:

\[
v_{id}(t+1) = v_{id}(t) + c_1 r_1 (p_{id}(t) - x_{id}(t))
\]

(3)

Then its performance deteriorates. The reason is the lack of information exchange between different particles and no social information sharing, making a group of size N equivalent to running N individual particles, so the probability of obtaining the optimal solution is very small. The third part of formula (3) is the "social" part, which represents the sharing of social information among particles. If only the social part is included in the speed evolution equation, that is:

\[
v_{id}(t+1) = v_{id}(t) + c_2 r_2 (g_{best} - x_{id}(t))
\]

(4)

The particles have no cognitive ability, that is, the "only society" model. In this way, particles have the ability to reach a new search space under the interaction. Although its convergence speed is faster than the basic particle swarm algorithm, it is easy to fall into the local optimum for complex problems.
3.1.3. Algorithm implementation steps
The standard PSO algorithm implementation process is as follows: first, determine the value of $m, c_1, c_2$; then set the initial value of $x_i (i=1, 2, ..., m)$ to make it evenly distributed in $[-v_{\text{max}}, v_{\text{max}}]$; then set the initial value of $v_i (i=1, 2, ..., m)$ to make it evenly distributed in $[-v_{\text{max}}, v_{\text{max}}]$, and calculate each value according to the optimization objective function. For each particle fitness value, initialize $p_{\text{best}}$ for each particle and $g_{\text{best}}$ for the group. After that, it enters the iterative loop. Within the specified number of loops, the next position and velocity of each particle in the search space is determined according to the evolution equation, and $p_{\text{best}}$ and $g_{\text{best}}$ are updated according to the fitness value of each particle in each generation, until the maximum iteration is reached. The number of times or the searched optimal position meets the predetermined threshold, and the final output $g_{\text{best}}$ is the optimal solution found by the algorithm. The algorithm flow chart is shown in Figure 3:

![Flow chart of particle swarm algorithm](image)

**Fig.3** Flow chart of particle swarm algorithm

3.2. Mutation particle swarm optimization algorithm with improved inertia weight
Like other optimization algorithms, particle swarm optimization has its shortcomings. Premature convergence and slow convergence in the later stages of evolution are two major problems of the PSO algorithm. During the operation of the standard PSO algorithm, if a particle finds a current optimal position, other particles will quickly move closer to it. When the optimal position is a local optimal position, the particle swarm cannot search for new ones in the solution space. The global
extremum of $g_{best}$, the algorithm is prone to premature convergence. From the evolution equation (1) of the algorithm, it can be seen that the main reason for premature convergence is: as the particles gather to the global optimal position $g_{best}$, the "social" and "cognitive" parts of the particle velocity $v$ will gradually become smaller, because $v$ less than 1, the speed of the particles will not increase any more. When the "social" part and the "cognitive" part approach 0, the speed will quickly drop to 0, and the particles will lose their ability to explore space and lack their position. Diversity, falling into a local optimal solution. In theory, PSO, a random group search-based optimization algorithm, must test every point in the space to ensure that it converges to the global optimal point [5]. However, the number of calculations that this approach brings is extremely large and unrealistic. Therefore, in this section, we design a variant particle swarm optimization algorithm (IWMPSO) to improve the inertia weight, and consider making the particles leap the entire search space as much as possible in the initial stage of the search, in order to obtain better diversity in the particle position; When the group falls into premature convergence, the mutation mechanism is used to make it jump out of the local optimum; in the later stage of evolution, the convergence speed and accuracy of the group are improved to make it converge to the global optimal solution as soon as possible.

4. Node scheduling based on improved particle swarm algorithm and energy

4.1. Related models of the algorithm
The node scheduling algorithm in this paper is based on ensuring the network coverage, and at the same time according to the principle of minimum energy consumption in the cluster, the nodes are scheduled to work/sleep, so the relevant coverage and energy consumption models need to be considered. Here we use the grid method proposed in [9] to simulate the coverage area of the entire network. The monitoring area of the sensor is divided into a number of equally spaced virtual grids. The vertices of the grid are called grid points, also called sampling points. When the grid density is large enough, the entire area coverage can be approximated as a point coverage, as shown in the figure 4 shown. Of course, the conversion from regional coverage to point set coverage will cause deviations between the coverage model and the actual situation [6]. Therefore, it is necessary to find a suitable mesh density to meet certain accuracy requirements while maintaining low computational complexity.

![Fig.4 Grid point simulation coverage area](image)

Under the acceptable fault tolerance rate, it depends on the transmission distance. The wireless transmission energy consumption model is shown in Figure 5.
4.2. Simulation experiment and result analysis

The algorithm proposed in this paper is compared with the multilateral measurement method and the iterative positioning method based on the standard PSO algorithm. Use Matlab7.0 for simulation experiment. WSN contains M=30 anchor nodes and N=100 unknown nodes, and is deployed in a square area of 100m×100m. Any three anchor nodes are not collinear, and the unknown nodes are evenly distributed in the area. The communication radius between the anchor node and the unknown node is r=25m. First, analyze the impact of the particle population on the algorithm. The number of particles varies from 30 to 120, the number of iterations is 100, and the other parameters of the algorithm remain unchanged. The average value is taken from 20 experiments. The experimental results are shown in Table 1:

| Number of particles | Number of selected working nodes | Network single round energy consumption (mJ) | Algorithm running time (S) |
|---------------------|---------------------------------|---------------------------------------------|---------------------------|
| 30                  | 21                              | 2115.1                                      | 58.750                    |
| 40                  | 20                              | 2081.6                                      | 84.356                    |
| 50                  | 19                              | 2019.5                                      | 106.953                   |
| 60                  | 18                              | 1904.4                                      | 119.469                   |
| 70                  | 18                              | 1921.7                                      | 136.844                   |
| 80                  | 18                              | 1891.8                                      | 148.239                   |
| 90                  | 18                              | 1931.6                                      | 170.326                   |
| 100                 | 18                              | 1891.8                                      | 183.875                   |
| 110                 | 18                              | 1901.3                                      | 204.093                   |
| 120                 | 18                              | 1882.3                                      | 225.688                   |

It can be seen from Table 1 that when the number of particles is 30, the algorithm has the worst effect. This is because the number of particles is too small and the global optimal solution cannot be found in the specified number of iterations. With the increase in the number of particles, the performance of the algorithm tends to stabilize, and the number of selected working nodes gradually converges to 18. This means that only 18 of the 50 sensor nodes that are randomly broadcast can be used to monitor the entire monitoring area. But at the same time the running time of the algorithm is also increasing. When too many particles are selected, the algorithm will become slower. Therefore, the proper selection of the number of particles has a great impact on the performance of the algorithm. After considering the performance and running time of the algorithm, the number of particles in this paper is 60.

In the particle swarm optimization algorithm in this paper, the end condition of the search is to reach the maximum number of iterations, which makes the given number of iterations have a certain impact on the performance and speed of the algorithm. We set the number of particles to 60, \( v_{\text{max}} = 6 \), and the number of iterations were 100, 200, 300, and 400 for the experiment. The experimental results are shown in Table 2:
Table 2. The influence of the number of iterations on the algorithm

| Number of iterations | 100   | 200   | 300   | 400   |
|----------------------|-------|-------|-------|-------|
| Select the number of working nodes | 18    | 18    | 18    | 18    |
| Network energy consumption (mJ)   | 1901.3| 1899.6| 1886.2| 1884.7|
| Algorithm running time (s)        | 123.634| 206.185| 323.928| 375.712|

It can be seen from Table 2 that the change in the number of iterations has a small effect on the algorithm, but it can greatly affect the speed of the algorithm. As the number of iterations increases, the network energy consumption gradually decreases. But for speed, as the number of iterations decreases, it has been greatly improved. Therefore, on the basis of ensuring the effect of the algorithm, we can appropriately reduce the number of iterations to improve the speed of the algorithm. After the summary of the experiment, this article sets the number of iterations to 100. Figure 6 shows the positioning error under different ranging errors. When there is no ranging error, the accuracy of the multilateral measurement method is better than that of the standard PSO algorithm and SPSO algorithm. However, when there is a ranging error, the positioning error of the multilateral measurement method increases rapidly with the increase of the ranging error. When the distance error reaches 50%, the positioning error reaches 19.67m.

![Fig.6 Positioning errors under different ranging errors](image)

When there is a ranging error, the positioning error of the standard PSO algorithm and the SPSO algorithm is smaller than that of the multilateral measurement method. The positioning error of the standard PSO algorithm is about 30.62% to 43.10% of the multilateral measurement method, and the positioning error of the SPSO algorithm is about 29.79% to 41.50% of the PSO algorithm, which is about 84.29% to 97.29% of the standard PSO algorithm. It can be seen that the SPSO algorithm has the best positioning performance, especially when the ranging error is greater than 25%, the positioning error of the SPSO algorithm is significantly smaller than that of the standard PSO algorithm.

5. Conclusion
The localization problem in wireless sensor networks can be modeled as an optimization problem, which can be solved using bionic algorithms. Particle swarm algorithm is a kind of swarm intelligence algorithm, which has been widely used in the routing protocol, topology control and node positioning of wireless sensor networks.
References

[1] Wang, J., Ju, C., Kim, H. J., Sherratt, R. S., & Lee, S. A mobile assisted coverage hole patching scheme based on particle swarm optimization for WSNs. Cluster Computing, vol. 22, pp. 1787-1795, January 2019.

[2] Chen, X., & Zou, S. Improved Wi-Fi indoor positioning based on particle swarm optimization. IEEE Sensors Journal, vol. 17, pp. 7143-7148, Twenty-one 2017.

[3] Gumaida, B. F., & Luo, J. An efficient algorithm for wireless sensor network localization based on hierarchical structure poly-particle swarm optimization. Wireless Personal Communications, vol. 97, pp. 125-151, January 2017.

[4] Ibrahim, R. A., Ewees, A. A., Oliva, D., Abd Elaziz, M., & Lu, S. Improved salp swarm algorithm based on particle swarm optimization for feature selection. Journal of Ambient Intelligence and Humanized Computing, vol. 10, pp. 3155-3169, August 2019.

[5] Yang, A., Yang, X., Liu, W., Han, Y., & Zhang, H. Research on 3D positioning of handheld terminal based on particle swarm optimization. Journal of Internet Technology, vol. 20, pp. 563-572, February 2019.

[6] Tam, N. T., & Hai, D. T. Improving lifetime and network connections of 3D wireless sensor networks based on fuzzy clustering and particle swarm optimization. Wireless Networks, vol. 24, pp. 1477-1490, May 2018.

[7] Sakamoto, S., Ozera, K., Barolli, A., Ikeda, M., Barolli, L., & Takizawa, M. Implementation of an intelligent hybrid simulation systems for WMNs based on particle swarm optimization and simulated annealing: performance evaluation for different replacement methods. Soft Computing, vol. 23, pp. 3029-3035, September 2019.