

# Research of the Positioning of an Improved WSN Particle Swarm Optimization in Node Ranging

Fangheng Xu

(Zhejiang Industry Polytechnic College Shaoxing Zhejiang 312000 China)  
[zjsxxfh@126.com](mailto:zjsxxfh@126.com)

## Abstract

*To order to know how to better solve the problem of node positioning in WSN, this paper mainly improves the particle swarm optimization of wireless sensor network node positioning in RSSI ranging. Aiming at the problem that local convergence exists in the algorithm, dynamic disturbance factors are introduced to reduce the speed of local convergence. And penalty function is introduced, thereby limiting the search range of feasible solutions and avoid consuming time in ineffective solution space. Compared with other algorithms, the improved algorithm has significant changes in the effects of convergence speed and stability analysis.*

**Keywords:** WSN, dynamic disturbance factor, penalty function

## 1. Introduction

The integration of data information acquisition, data fusion and data transmission of wireless network's information set has extensive prospect in emergency rescue and disaster relief, urban traffic congestion management and industrial production, *etc.* [1] Node positioning in wireless sensor network is of vital importance. Therefore, it has always been the focus of research in the field of wireless sensor network to improve the accuracy of node positioning in sensors. [2]

Aiming at the node positioning of wireless sensor network, both domestic and foreign scholars have conducted a lot of research, mainly based on ranging and ranging-free methods. The main representative methods include Received Signal Strength Indication (RSSI), Time of Arrival (TOA), Time Difference of Arrival (TDOA), and Angle of Arrival (AOA), *etc.* [3-4] Ranging-free algorithms mainly include centroid method and DV-Hop *etc.* [5-7] The RSSI ranging method is the most widely used based ranging method, and aiming at its relatively low positioning accuracy, numerous scholars have improved it. In Literature [8], positioning algorithm RMDV-Hop based on modified WSN of RSSI (RSSI Modified DV-Hop), and this algorithm limits the maximum transmission hops and uses the RSSI value to modify the transmission value of anchor nodes with one hop. The positioning accuracy and stability of improved algorithm have improved significantly compared with that of the original algorithm. In literature [9], the RSSI distance ratio is used to calculate the distance matrix between nodes combined with the Euclidean ranging technology. MDS algorithm is used to establish corresponding global coordinate system. In literature [10], neural network is applied to the RSSI ranging to reduce RSSI's impact on the environment. In this paper, particle swarm algorithm is introduced to the wireless sensor network nodes of RSSI ranging for the improvement. Because particle swarm algorithm is easy to be achieved with less parameters and stronger global searching ability, it is widely used in node positioning combined with algorithms. However, in the later phase, population diversity of the algorithm is reduced and the algorithm is easy to fall into local optimum. Aiming at this situation, dynamic disturbance factor is introduced in this paper to reduce the speed of local convergence; penalty function is introduced so as to limit the searching range of feasible solutions and avoid consume time in the invalid

solution space. Compared with other algorithms, the improved algorithm has significant changes in the effects of convergence speed and stability analysis.

## 2. Ranging Based on RSSI

Because the hardware structure in wireless sensor is comparatively simple, so is the RSSI ranging technology. And extra complicated hardware equipment is not needed. Therefore, it is very suitable for wireless sensor network nodes. In the RSSI ranging, it measures the power of signals received at the place of receiving nodes so as to estimate the signal loss during transmission, whereby convert the power loss during transmission to the distance between two nodes. It is shown as follows:

$$P(x) = PG_iG_r\lambda / x^2L \quad (1)$$

In the formula,  $P(x)$  refers to the accepted power  $x$  between nodes sending signals,  $P$  refers to the power of nodes sending signals,  $G_i$  refers to the transmission gain of sending node, and  $G_r$  refers to the transmission of receiving node.  $L$  refers to loss constant, and  $\lambda$  refers to the length of wave transmitting signals. Through formula (1), the actual distance between each node and node that sends signals. However, due to the influence of external interference factors like terrain, there may be great loss of power during the transmission of node signals. Thus, it is more difficult to determine the positions of unknown nodes, easily resulting in deviation during the transmission of signals, thus requiring more accurate positioning of the algorithm.

## 3. Basic Particle Swarm Algorithm

It is an algorithm to solve the problem of multi-objective combination optimization, and it is widely used due to less parameters and being easy to be realized. And its basic idea is to compare individuals in the space to particles without quality and volume. The algorithm is described as follows: the  $i$  particle is  $x_i = \{x_{i1}, x_{i2}, \dots, x_{im}\}$ , and the position it covers is  $p_i = \{p_{i1}, p_{i2}, \dots, p_{im}\}$ . Herein, the best position is  $p_{best}$ , the best positions of all the particles in the current group are  $p_{gbest}$ , and the speed of particle  $i$  is expressed with  $v_i = \{v_{i1}, v_{i2}, \dots, v_{in}\}$ . The movement formula of particle  $i$  is:

$$v_{id}(t+1) = w \times v_{id}(t) + c_1 \times rand() \times (p_{best} - x_{id}(t)) + c_2 \times rand() \times (p_{gbest} - x_{id}(t)) \quad (2)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (3)$$

Herein,  $w$  refers to inertia weight, mainly used for iterative linear change.  $c_1$  and  $c_2$  are constants, controlling the individual's optimum position and the global optimum position respectively. It is beneficial for the algorithm to find its optimum and improve its convergence speed.  $rand()$  is a random number, valued between (0,1). For each particle, the individual's optimal position  $P_i$  and the global optimal position are shown with the following formulas respectively:

$$P_i^{k+1} = \begin{cases} X_i^{k+1}, & f(X_i^{k+1}) \leq f(P_i^k) \\ P_i^k, & f(X_i^{k+1}) \geq f(P_i^k) \end{cases} \quad (4)$$

$$f(P_g) = \min[f(P_i)] \quad (5)$$

## 4. Application of Improved Particle Swarm Algorithm in Node Positioning

### 4.1 Introducing Dynamic Disturbance Factors

The process of current particle swarm algorithm optimization is making changes to the algorithm's parameters, that is, improving  $c_1$  and  $c_2$  in Formula (2) and Formula (3). It has been proven in literature [25] that there is absolute no relationship between the evolution of particles in the particle swarm algorithm and the speed of particles. Therefore, the Formula (3) is changed as follows:

$$x_{id}(t+1) = w \times x_{id}(t) + c_1 \times rand() \times (p_{best} - x_{id}(t)) + c_2 \times rand() \times (p_{gbest} - x_{id}(t)) \quad (6)$$

However, it can be found in Formula (4) that although this position is changed, the convergence may still reach its local optimum. Therefore, stagnation may be produced. In order to solve this stagnation problem, disturbance factor is introduced in this paper. When the algorithm stops, disturbance factors may be used to interfere with individual's optimal solutions and the global optimal solution respectively, thus making the algorithm stay far away from the stagnation region. The disturbance factors are shown as follows:

$$r_1^{\frac{t_p}{T_p}} = \begin{cases} F(0,1) & t_p \leq T_p \\ 1 & t_p \geq T_p \end{cases} \quad (7)$$

$$r_2^{\frac{t_q}{T_q}} = \begin{cases} F(0,1) & t_q \leq T_q \\ 1 & t_q \geq T_q \end{cases} \quad (8)$$

In the formula,  $t_p$  and  $T_p$  represent the stagnation step number of individual's optimal value and the threshold of individual optimal value's stagnation pace needing to be interfered with respectively. And  $t_q$  and  $T_q$  represent stagnation step number of global optimal value and the threshold of global optimal value's stagnation pace needing to be interfered with respectively. When the stagnation number and threshold are reached, the algorithm is interfered with. Therefore, the position adjustment equation to Formula (6) is as follows:

$$x_{id}(t+1) = w \times x_{id}(t) + c_1 \times rand() \times (r_1^{\frac{t_p}{T_p}} p_{best} - x_{id}(t)) + c_2 \times rand() \times (r_2^{\frac{t_q}{T_q}} p_{gbest} - x_{id}(t)) \quad (9)$$

It can be found in Formula (7) that when  $t_p > T_p$ , the algorithm  $t_q > T_q$  is stagnated and after the local optimal value is already converged, the current individual extreme value and global extreme value are disturbed with some certain lag. Secondly, after the algorithm is converged to local optimal value, it can still be implemented through disturbance factors. However, chances are that it is converged from one local optimum to another local convergence until it has gone through all the local optimum. When comparing the global optimum, the algorithm is converged, reducing the algorithm's effectiveness. The algorithm is related with the iteration number, so in order to overcome the low efficiency of algorithm, dynamic factors are introduced based on disturbance factors, which is as follows:

$$d_1 = \begin{cases} 1 & rand() < \frac{k - k_{\min}}{k_{\max}} \\ F(0,1) & rand() > \frac{k - k_{\min}}{k_{\max}} \end{cases} \quad (10)$$

$$d_2 = \begin{cases} 1 & rand() < \frac{k - k_{\min}}{k_{\max}} \\ F(0,1) & rand() > \frac{k - k_{\min}}{k_{\max}} \end{cases} \quad (11)$$

Herein,  $d_1$  are  $d_2$  dynamic factors aiming at individual local optimum and individual global optimum respectively. Set  $k$  as the number of iteration, and  $k_{\min}$  and  $k_{\max}$  as the minimum and maximum iteration numbers respectively. Previously, stagnation might be caused to the algorithm due to local convergence, so dynamic factors are added to make sure that the algorithm will not fall into stagnation due to local optimum. At the beginning stage, there is great influence and weight of the algorithm, global search is adopted mainly. And with the constant change of particle's speed and location, local search is carried out gradually. At this time, interference is carried out by constantly improving dynamic interference factors to prevent the algorithm's local convergence. Therefore, the combination of these two can obviously improve the algorithm's efficiency. It is shown as follows:

$$x_{id}(t+1) = w \times x_{id}(t) + c_1 \times rand() \times (r_1^{\frac{t_p}{T_p}} d_1 p_{best} - x_{id}(t)) \quad (12)$$

$$+ c_2 \times rand() \times (r_2^{\frac{t_g}{T_g}} d_2 p_{gbest} - x_{id}(t))$$

Meanwhile, in order to improve the particle swarm algorithm's overall convergence, inertia weight and current iteration times are connected. Suppose there is certain liner relation between  $w$  and  $k$ , so as make sure that the inertia weight value can be increased with the increase of iteration times, thus gradually representing a gradual increase of stability, guaranteeing it synchronous with the iteration times to some extent. The modification of  $w$  is as follows:

$$w = w_{\min} + \frac{k}{k_{\max}} (w_{\max} - w_{\min}) \quad (13)$$

Set  $w_{\max}$  and  $w_{\min}$  as the maximum and minimum value of weight respectively, and  $k_{\max}$  as the current maximum iteration times.

## 4.2 Representation of Penalty Function

There may be some shadow areas of research in the search zone to some extent. Therefore, in order to reduce the search range of the optimum solutions and reduce unnecessary search time, penalty function is introduced to convert the unconstrained particle movement into feasible solution with constraints. When the particles meet the constraints, the value of penalty function will be small, and then the particles do not meet the constraints, the value of penalty function will large so as reduce the fitness of particle's target function's value, guide the particles stay away from the feasible region so as to reduce the time of approaching the optimal solutions. The penalty expression is shown as Formula (14):

$$F(x, M) = f(x) + M \sum_{i=1}^m [\min(0, g_i(x))] \quad (14)$$

Suppose  $f(x)$  is the target function, then the minimized result is to keep the minimum of target function and use  $A$  to represent the set of feasible solutions. Therefore, herein  $\min f(x), x \in A$ ,  $A = \{x | g_i(x) \geq 0, i = 1, 2, \dots, m\}$ , herein,  $m$  is the range of constraints. Therefore,  $g_i(x) \geq 0, i = 1, 2, \dots, m$  is converted into  $\min(0, g_i(x))$ ,  $M$  is the penalty factor, and  $F(x, M)$  is the penalty function. When the feasible solutions meet the constraint conditions, the value of  $g_i(x)$  is small, and vice versa so as to give big penalty and convert the constraint problem into unconstrained optimization problem  $\min F(x, M)$ .

### 4.3 Description of the Algorithm

Step 1: Set the population size, the parameter penalty factor  $M$  as 0.5. And set the maximum and minimum value of inertia weight  $\omega$ , the maximum iteration times and learning factor  $c_1$  and  $c_2$ .

Step 2: Through the modification parameter of ranging error to carry out the constraint condition of feasible solutions and generate penalty function at the same time.

Step 3: Initialize the particle swarm algorithm and the position of feasible solutions at random.

Step 4: Initialize the value of each feasible solution's fitness, choose the particle's initial value as the individual extreme value  $P_i$ , and meanwhile choose  $P_g = \min\{P_1, P_2, \dots, P_i\}$ .

Step 5: Add one more time to the iteration time, and calculate the location of next feasible solution according to Formula (8).

Step 6: Calculate the fitness of a new generation of feasible solutions. Renew the position of individual's optimal solution and compare each particle's individual fitness and the individual's optimal solution  $P_i$  that has been searched. If the comparison result is small, it indicates that the current position is better, so then renew  $P_g$ . Otherwise, keep it unchanged. Renew the global optimum solution, and compare all the particles' optimal value  $P_i$  and the global optimal value  $P_g$ . If the result is small, then renew the global extreme value of the entire particle swarm. Otherwise, keep it unchanged.

Step 7: Judge whether the maximum iteration times have been reached, if so, finish the positioning. Otherwise, continue implementing Step 5.

### 5. Simulation Experiment

With the Windows xp, Core i3 as the CPU, 4GDDR3 as the memory and Matlab2010 as the simulation software, the comparison in this paper is carried out in two aspects: one is through the comparison with basic particle swarm algorithm and reference literature algorithm; another is through the comparison with the maximum likelihood value, motion estimation and maximum likelihood algorithm. The author analyzes the convergence speed of algorithm in this paper and the effect of stability analysis.

Set the nodes are distributed randomly at the 100m\*100m two-dimensional plane. The location of nodes is distributed at random and anchor nodes are chosen at random. For the same environmental location, set the simulation times as 100, the size of particle swarm as  $N=50$  and the iteration times as 50. Meanwhile, set the penalty factor  $M$  as 0.5,

learning factors  $c_1$  and  $c_2$  are 2, and the maximum and minimum value of inertia function  $\omega$  are 1 and 0.1 respectively. The average location error

$$AV = E\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad \text{and} \quad \text{positioning error variance}$$

$$MS = E[(x_1 - x_2)^2 + (y_1 - y_2)^2]$$

are chosen as the standard to evaluate the positioning performance.  $(x_1, y_1)$  is the estimated position of nodes and  $(x_2, y_2)$  is the actual position.

### 5.1 Convergence of Algorithm

In order to further compare the convergence of algorithm in this paper, algorithm in this paper, basic PSO algorithm, Literature [8] and Literature [11] are chosen to make comparison. And the comparison results are shown as follows:

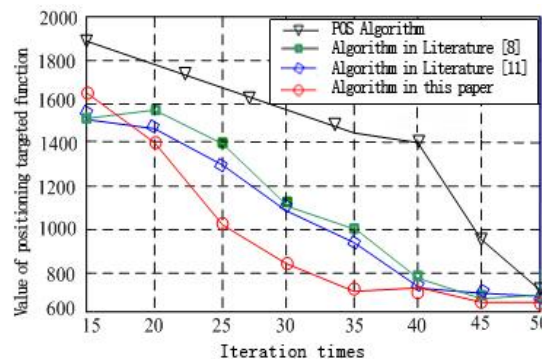


Figure 1. Convergence of Positioning Algorithm's Convergence Value

It can be found from this figure that compared with other three algorithms, algorithm in this paper is more efficient mainly because speed is not considered in algorithm in this paper so as to avoid the slow convergence speed of the algorithm. Besides, disturbance factors are added to the algorithm, reducing the convergence speed of the algorithm and making it better for the algorithm to find its global optimal solution, thus further improving the stability of the algorithm.

### 5.2 Comparison of Positioning Error of Estimation Algorithm

In this part, algorithm in this paper is compared with motion estimation algorithm, maximum likelihood algorithm and extreme likelihood algorithm, and the results are as shown in Figure 2:

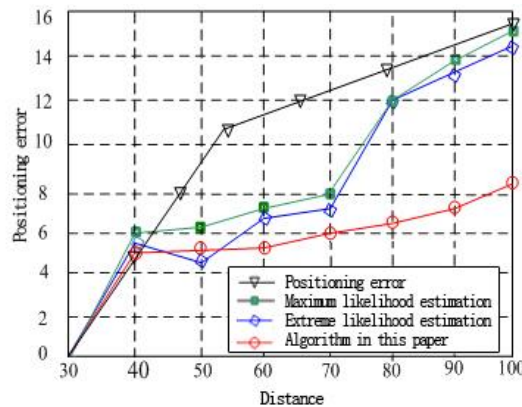


Figure 2. Comparison of Positioning Error

It can be found in Fig.2 that algorithm in this paper is superior to the several estimation aspects in positioning error, which is because disturbance factors are introduced to the algorithm to improve the efficiency of local search and avoid taking local optimum as the global optimum and finally provide the global optimal solution to the algorithm. Modification is made aiming at the specific error to restrict the probable range of node positioning, narrow the search range and improve the positioning effect.

### 5.3 Influence of Average Positioning Error to Ranging Error

Ranging errors are a problem that cannot be ignored. With the different ranging distance, the ranging error of both algorithm in this paper and other algorithms is increasing. However, under different ranging conditions, the average error of positioning algorithm in this paper is less than that of algorithm in the reference, which indicates that algorithm in this paper has more accuracy and less error fluctuation of positioning accuracy. This shows that algorithm in this paper has great error resistance, thus demonstrating relatively better positioning performance. it is as shown in Fig.3:

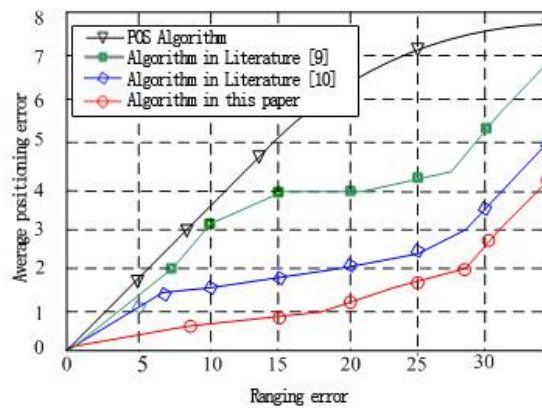


Figure 3. Average Positioning Error

## 6. Conclusion

In this paper, particle swarm algorithm is improved based on RSSI wireless sensor network node positioning. And aiming at the local convergence of the algorithm, dynamic disturbance factors are introduced, reducing the speed of local convergence. Penalty function is introduced to limit the searching range of feasible solutions and avoid consume time in invalid solution space. Compared with other algorithms, the improved algorithm has significant changes in the effects of convergence speed and stability analysis.

## References

- [1] M. Li and Y.H. Liu, "Range-free localization in anisotropic sensornetworks with Holes [J]", IEEE/ACM Transactions on Networking, vol. 18, no. 1, (2010), pp. 320-332.
- [2] L. Lazos and R. Poovendran, "High-resolution robust localization for wireless sensor networks [J]", IEEE Journal on Selected Areas in Communications, vol. 24, no. 2, (2006), pp. 233-246.
- [3] D.Y. Zhang and G.D. Cui, "A union node localization algorithm based on RSSI and DV-Hop for WSNs [C]", //Proceedings of Instrumentation, Measurement, Computer, Communication and Control, vol. 12, (2012), pp. 1094-1098, Harbin.
- [4] H. Woo and S. Lee, "Range-free localization with isotropic distance scaling in wireless sensor networks [C]", //Proceedings of International Conference on Information Networking, (2013).
- [5] H. Li, S.W. Xiong and Y. Liu, "An Improvement of DV-Hop Localization Algorithm for Wireless Sensor Network [J]", Chinese Journal of Sensors and Actuators, vol. 24, no. 12, (2012), pp. 1782-1786.
- [6] J.J. Bai and X.P. Yan, "Research of location based on mixed algorithm of weighted centroid and DV-Hop in WSN [J]", Application Research of Computers, vol. 26, no. 6, (2009), pp. 2248-2251.

- [7] K.J. Mao, X.M. Zhao and W.X. He, "Area Division Based-auto DV-Hop Location Algorithm in WSN [J]", Computer Science, vol. 39, no. 3, (2012), pp. 39-42.
- [8] S.L. Wu, W.L. Zhang and X.Y. Yang, "WSN Localization algorithm based on RSSI correction [J]", Journal of Chongqing University (natural Science Edition), vol. 37, no. 8, (2014), pp. 144-149.
- [9] G.P. Wu, H.Q. Yu and G.S. Fan, "Journal of East China Universtiy of Science and Technology (natural Science Edition) [J]", vol. 39, no. 5, pp. 596-600.
- [10] J. Sun and Y.G. Du, "Applicaition of RSSI-neural network in wirless sensor network Positioning [J]", Science Technology and Engineeringm, vol. 13, no. 6, (2013), pp. 1479-1482.
- [11] S.L. Chen, Li-lei and B.W. Zhu, "Computing method of RSSI probability centroid for location in wsn [J]", Journal of Zhejiang Universtiy (Engineering Science), vol. 48, no. 1, (2014), pp. 100-104.
- [12] V. Geetha and V. Pranesh, "Kallapurb.Clustening in Wireless Sensor Networks:Performance Comparison of LEACH&LEACH-C Protocols Using Matalb [J]", Procedia Technology, vol. 4, (2012), pp. 163-170.

### **Author**

**Fangheng Xu** (1980.9) male, master, research direction: database and information system, information security, wireless sensor