Original Research Article

# Node location algorithm of artificial fish swarm mixing based on improved particle swarm

*Sun Shaowei[1,2], Shen Guodong[1], Liu Mingzhou[2], Zou Jiao[1], Wu Yue[3]*

*1 Anhui Transport Consulting & Design Institute Co., Ltd, Hefei, Anhui, 230088, China*

*2 School of Mechanical and Automotive Engineering, Hefei University of Technology, Hefei, Anhui, 230009, China*

*3 School of Computer and Artificial Intelligence, Hefei Normal University, Hefei, Anhui, 230601, China*

***Abstract:*** Node positioning technology is a hot research area in wireless sensor networks (WSN). In view of the problems of large node localization errors in traditional DV-Hop algorithm, premature convergence and local optimization in PSO algorithm, a node localization algorithm based on the combination of random weight particle swarm optimization (RWPOS) and artificial fish swarm algorithm (AFSA) is proposed.By analyzing the inertia weight, maximum velocity, and learning factor of particles, the particle swarm optimization algorithm was improved and combined with the artificial fish swarm algorithm. Finally, four algorithms were compared and analyzed in three-dimensional space to investigate the impact of different beacon node ratios, different node communication radii, and different node numbers on node positioning error.The MATLAB simulation results show that compared with the improvement of the single algorithm, the hybrid algorithm has strong convergence and effectively improves the node positioning accuracy, thus providing an important basis for the application of this algorithm in practical situations.

***Keywords:*** Node location; WSN; RWPOS; AFSA; Positioning accuracy

**Classification:** Electromagnetic theory

## 1. Introduction

Wireless sensor networks are the core component of the Internet of Things (IoT). As a new type of intelligent network system, they are usually composed of micro-sensor nodes, control systems, and wireless communication systems, and are mainly used in environmental monitoring, traffic management, industrial safety, and construction projects. In practical application environments, the unknown nodes of micro-sensors are generally randomly distributed, while the deployment of beacon nodes is known. Nodes form an ad hoc network system through wireless communication to acquire and transmit information. The location information of unknown nodes is the premise and foundation for the effective deployment and application of wireless sensor networks.[1] The xisting node positioning algorithms mainly include ranging-based positioning algorithms, non-ranging positioning algorithms, and combinations of multiple algorithms. The ranging-based node positioning algorithms mainly include the signal strength indicator (RSSI) algorithm, the time of arrival (TOA) algorithm, and the angle-of-arrival (AOA) algorithm. These algorithms have high positioning accuracy, simple calculation, and high coverage, but require complex hardware support. Non-ranging node positioning algorithms mainly include the centroid algorithm[5], the distance vector-hop (DV-Hop) algorithm[6], and the approximate point in triangle (APIT) algorithm[7]. These algorithms have low positioning accuracy, computational complexity, and do not rely heavily on complex hardware support.

DV-Hop algorithm is the most widely used non-ranging algorithm. This algorithm has low cost and simple implementation, but when the nodes are unevenly distributed, the positioning accuracy is low. In order to better compare and analyze the effect of node positioning in three-dimensional space, this article first describes the traditional DV-Hop positioning algorithm, then introduces the concepts of particle swarm optimization and artificial fish swarm algorithm, analyzes the causes of node positioning errors in both algorithms, proposes an improved fusion of particle swarm optimization and artificial fish swarm algorithm, introduces a correction for positioning errors, and finally verifies the rationality of the proposed hybrid algorithm through simulation experiments, significantly improving the accuracy of node positioning.

## 2. Traditional DV-Hop algorithm

DV-Hop localization algorithm is a typical ranging-free node localization algorithm, first proposed by Niculescu D et al. from Rutgers University in 2003[8]. This algorithm uses exchange routing protocols to obtain information such as the minimum and average hop counts between beacon nodes and unknown nodes through a small number of beacon points. Then, using this information, the distance between unknown nodes and beacon nodes is calculated, and finally the location of the unknown nodes is determined through coordinate calculation.

In three-dimensional space, the DV-Hop localization algorithm generally includes three stages[9]:

The first stage is to obtain the minimum hop count between beacon nodes and unknown nodes. The location of the beacon node is known. Initially, the location information and initial hop count value are broadcasted to the entire network in a flooding manner. Each time it is forwarded, the hop count value is incremented by one. Based on this mechanism, the minimum hop count between the beacon node and the unknown node is calculated.[9] The calculation formula is as follows:

$$d_{ij} = \mathrm{HopSize}_i \bullet hop_{ij}$$

In the formula, ($x_i$, $y_i$, $z_i$) and ($x_j$, $y_j$, $z_j$) represent the coordinates of beacon nodes $i$ and $j$, $hop_{ij}$ represents the minimum hop count between beacon nodes $i$ and $j$, and HopSizei represents the average hop distance of beacon node i.

The second stage is to calculate the average hop distance between the unknown node and the nearest equal beacon node. Estimate the average distance per hop between anchor nodes based on the actual distance between them divided by the number of hops between them, and broadcast it as a correction value. When locating an unknown node, only the correction value of the anchor node closest to itself is used as the average distance per hop between itself and other anchor nodes. The calculation formula is as follows:

$$HopSize_i = \frac{\sum_{i \neq j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}}{\sum_{i \neq j} hop_{ij}}$$

The third stage is to calculate the coordinates of the unknown nodes. When an unknown node calculates the distance to at least four known beacon points in three-dimensional space, it can use the four-sided measurement method or maximum likelihood estimation method to calculate the location of the unknown node, and finally obtain the coordinate information of the unknown node. If the coordinates of $m$ beacon nodes are $A_i$ ($x_i$, $y_i$, $z_i$), where $i$=1, 2, 3..., $m$, and $m \geq 4$; The coordinates of the unknown node are $U$(x, y, z), and the calculated distances between the beacon node and the unknown node are $d_1$, $d_2$, $d_3$..., and $d_m$. Taking the maximum likelihood estimation method as an example, the coordinates of the unknown node $U$ can be solved. The calculation formula is as follows:

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 = d_1^2 \\ (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 = d_1^2 \\ \vdots \\ (x - x_m)^2 + (y - y_m)^2 + (z - z_m)^2 = d_m^2 \end{cases}$$

By subtracting the last row from the previous ($n-1$) rows in the above formula, the coordinates of point $U$ can be calculated as shown in the following formula:

$$X = (A^T A)^{-1} A^T \mathrm{B}$$

Among them:

$$A=\begin{bmatrix} 2(x_1-x_m) & 2(y_1-y_m) & 2(z_1-z_n) \\ 2(x_2-x_m) & 2(y_2-y_m) & 2(z_2-z_n) \\ \cdots\cdots & \cdots\cdots & \cdots\cdots \\ 2(x_{m-1}-x_m) & 2(y_{n-1}-y_n) & 2(z_{n-1}-z_n) \end{bmatrix}$$

$$B=\begin{bmatrix} x_1^2-x_m^2+y_1^2-y_m^2+z_1^2-z_m^2+d_m^2-d_1^2 \\ \cdots\cdots \\ x_{m-1}^2-x_m^2+y_{n-1}^2-y_m^2+z_{n-1}^2-z_m^2+d_m^2-d_{m-1}^2 \end{bmatrix}$$

# 3. Particle swarm optimization algorithm

Particle Swarm Optimization (PSO) algorithm is a new type of stochastic swarm intelligent evolutionary algorithm. Each particle in the particle swarm is abstracted as a “particle” without mass and volume, and each “particle” in the swarm represents a possible solution to the optimization problem. Through communication and information sharing among particles, the “particles” constantly move in the state space and exert swarm intelligence to find the optimal solution.[10]

The standard particle swarm algorithm is described as follows: Assume that there is a particle population in an *N*-dimensional space, with *N* particles in the population and a solution space dimension of *D*. The position coordinates of each particle can be represented as $X_i$=($x_{i1}$, $x_{i2}$,..., $x_{iD}$), and the velocity can be represented as $v_i$=($v_{i1}$, $v_{i2}$...,, $v_{iD}$). The optimal position searched by particle *i* is denoted as *Pi*=($p_{i1}$, pi2,, piD), and the optimal position searched by the entire particle population so far is denoted as Qi=(qg1, $q_{g2}$,, $q_{gD}$). The update formula for the velocity and position of particle *i* is:

$$v_{id}^{k+1}=\omega\cdot v_{id}^{k}+c_1\cdot r_1\cdot\left(p_{id}-x_{id}^{k}\right)+c_2\cdot r_2\cdot\left(q_{id}-x_{id}^{k}\right)$$

$$x_{id}^{k+1}=x_{id}^{k}+v_{id}^{k+1}$$

Where: *i*=1, 2...,, *N*; $c_1$ and $c_2$ are called learning factors, preferably within the range of [0, 4]; $r_1$ and $r_2$ are random numbers uniformly distributed between [0, 1]; *k* is the number of iterations; $V_{id}^{k+1}X_{id}^{k+1}$ represents the position of the ith particle at the *k+1th* iteration, and represents the velocity of the ith particle at the *k+1th* iteration.[11]

Particle swarm optimization algorithm is a swarm intelligence algorithm, which has the advantages of simple calculation method, concise parameters and easy implementation. The disadvantage is that if the parameters are not properly selected or the particle population size is not appropriately selected, the algorithm particles are prone to “premature” behavior and falling into local optima.

# 4. Random weighted particle swarm optimization algorithm

In order to better utilize the characteristics of the particle swarm optimization algorithm, a random weight particle swarm optimization (RWPSO) algorithm is proposed, which improves the speed, inertia weight, learning factor, and position update to enhance the positioning accuracy and stability of the algorithm.[12]

### 4.1. Improvement of the maximum speed of particle motion

During the motion of particles, if the maximum speed is not limited, it may exceed or be limited by the search space range. On the one hand, if the particle velocity value is set too large, it can easily lead to particles flying out of the feasible solution region. On the other hand, if the particle speed is set too low, the particle speed will be limited and it is easy to fall into local convergence and cannot jump out of the local area. Therefore, setting a reasonable maximum speed can effectively prevent computational overflow and control spatial search accuracy issues.

It is noted that the particle motion speed $v_i$ is set in the range of [-$v_{max}$, $v_{max}$], the particle search space is [-$x_{max}$, $x_{max}$], and the maximum speed calculation formula is as follows:

$$v_{max}=\gamma \cdot x_{max},$$

In the formula, $v_{max}$ is the maximum speed of particle motion, and $x_{max}$ is the maximum amplitude of the spatial region, with a $\gamma$ value range between (0.1, 1.0).

### 4.2. Improvement of inertia weight

Inertia weight is $\omega$ used to reflect the mutual influence between particle speeds in the particle swarm optimization algorithm, determining the proportion of the particle's current speed inheritance. If you need to make particles better able to search, you need to choose a larger inertia weight, which is very helpful for enhancing the global optimization of the algorithm, but the convergence speed is not very fast. If you need to make particles have better development capabilities, you need to choose a smaller inertia weight, which can improve the local optimization effect of the algorithm and accelerate the convergence speed.

This article proposes introducing a weight factor into the particle swarm algorithm $\omega$, which linearly decreases with increasing iteration times $\omega$ to adjust the speed of particles in both global and local optima. A series of studies have shown that when the $\omega$ linearity decreases from 0.9 to 0.4, the particle update calculation formula is guaranteed as follows:

$$\omega = \omega_{max} - (\omega_{max} - \omega_{min}) \cdot \frac{\mu}{k}$$

Where, $\omega_{min}$ $\omega_{max}$=0.9, =0.4; k is the current iteration number; μ The value range is between (0.1, 1.0).

### 4.3. Improvement of learning factor

$c_1$ and $c_2$ are called learning factors. $c_1$ represents the influence of the individual's historical optimal position on the particle's motion, which is called the particle's self-learning ability. $c_2$ represents the influence of the group's historical optimal position on the particle's motion, which is also called the particle's group learning ability. By setting the learning factors appropriately, the convergence speed of the algorithm can be accelerated, which is beneficial for quickly finding the optimal solution.

In the early stages of particle swarm optimization iteration, particles require strong self-learning capabilities and relatively weak group learning capabilities, so $c_1$ is larger and $c_2$ is smaller in the early stages of iteration. In the later stages of iteration, particles need to have strong group learning ability and relatively weak self-learning ability, so $c_1$ is smaller and $c_2$ is larger in the later stages of iteration. The improved calculation formula for the values of $c_1$ and $c_2$ is as follows:

$$c_1 = c_{min} - (c_{max} - c_{min}) \cdot \left(\frac{M-T}{M}\right)^3$$

$$c_2 = c_{max} - (c_{max} - c_{min}) \cdot \left(\frac{M-T}{M}\right)^3$$

Where, $c_{max}$ and $c_{min}$ are the maximum and minimum values of the learning factor; $M$ is the maximum number of iterations, and $T$ represents the current number of iterations.

## 5. Node localization algorithm based on improved particle swarm and artificial fish swarm hybrid

The particle swarm algorithm has a slow convergence rate in the later stages of iteration and is prone to falling into local optima. The Artificial Fish Swarm Algorithm (AFSA) has a good global convergence rate during the iterative process, allowing for rapid local search and escaping local optimal solutions, which is beneficial for quickly finding the optimal solution.[14] Based on the respective advantages of the two algorithms, a node localization algorithm based on improved particle swarm optimization and artificial fish swarm algorithm (RWP-

SO-AFSA) is proposed by integrating random weight particle swarm and artificial fish swarm algorithms. The position update formula of the particle swarm algorithm is improved as follows:

$$X_{id}^{k+1} = X_{id}^{k} + \varphi * V_{id}^{k+1}$$

In the formula, is $\varphi$ called the mutation factor. In order to improve the diversity of solutions and prevent the algorithm from falling into local optimal solutions, the $\varphi$ calculation formula for is set as follows:

$$\varphi=1+0.33\cdot r$$

In the formula, $r$ represents a random number between (0, 1).

# 6. Positioning error analysis

## 6.1. Description of the positioning problem

The coordinates of unknown nodes calculated in the 3D DV-Hop localization algorithm are estimated values, often resulting in large positioning errors. The main factors contributing to errors are hop value errors and average hop data errors. If the coordinates of m beacon nodes are $_{Ai}$ ($_{xi}$, $_{yi}$, $_{zi}$), where $i$=1, 2, 3..., $m$, m≥4, and the node coordinates are $U$(x, y, z), the estimated calculated distances between the beacon nodes and the unknown nodes are $_{d1}$, $_{d2}$..., $_{d3}$, and $_{dm}$.

During the node positioning process, assuming the true distance between the unknown node and the beacon node is $si$, and the ranging error is denoted as $\varepsilon_i$, we have $|_{di-si}| \le \varepsilon_i$, where $i$ = 1, 2, 3..., $m$, and m ≥ 4. The unknown node coordinates $U$(x, y, z) can be converted to the following formula:

$$\begin{cases} d_1-\varepsilon_1 \le \sqrt{(x-x_1)^2+(y-y_1)^2+(z-z_1)^2} \le d_1+\varepsilon_1 \\ d_{21}-\varepsilon_2 \le \sqrt{(x-x_2)^2+(y-y_2)^2+(z-z_2)^2} \le d_2+\varepsilon_2 \\ \vdots \\ d_m-\varepsilon_m \le \sqrt{(x-x_m)^2+(y-y_m)^2+(z-z_m)^2} \le d_m+\varepsilon_m \end{cases}$$

In the above formula, if the value of $\varepsilon_i$ is smaller, the deviation between the true values of the unknown node's solution is minimized, indicating that the unknown node is closer to the true coordinates. Therefore, solving the problem of finding the coordinates of the unknown node is transformed into finding the optimal solution to a constrained problem. The objective function for target positioning is as follows:

$$f(x,y,z)=\sum_{i=1}^{m} \left| \sqrt{(x_i-y_i)^2+\left(y_i-y_j\right)^2+\left(z_i-z_j\right)^2}-d_i \right|$$

The concept of a moderate function is introduced to evaluate the accuracy of the particle's unknown calculations and to guide the search direction of the algorithm. When the function is minimized, the optimal solution is obtained. The calculation formula is as follows:

$$_{fitnessi}=f(x,y,z)_{\min}=\sum_{i=1}^{m}\frac{1}{\mathrm{m}}\sqrt{(x_i-y_i)^2+\left(y_i-y_j\right)^2+\left(z_i-z_j\right)^2}-d_i \mid$$

In the formula, $_{fitnessi}$ is the fitness value of particle $_i$. This article introduces the fusion of artificial fish algorithm based on particle swarm optimization and analyzes and improves some parameters to improve the accuracy of the positioning algorithm.[15]

## 6.2. Description of positioning error

In order to reduce the interference of random errors in the experiment, the relative positioning error (Rerror) is used to analyze the positioning algorithm. The true coordinates of the unknown node are denoted as $_{Zi}$($_{xi}$, $_{yi}$, $_{zi}$), and the estimated coordinates of the unknown node at the kth time are denoted as Eki($_{xki}$, yki, $_{zki}$). $N$ is the number of unknown nodes, $R$ is the broadcast radius of the node, and $K$ is the total number of calculations. The relative positioning error is defined as follows:

$$Rerror = \frac{1}{NRK}\sum_{k=1}^{K}\sum_{i=1}^{N}\sqrt{(x_i - x_{ki})^2 + (y_i - y_{ki})^2 + (z_i - z_{ki})^2}$$

# 7. Experimental simulation analysis

In order to verify the positioning performance of the improved optimization algorithm, simulation experiments were conducted using MATLAB R2021a software, and the positioning accuracy of four algorithms, including the DV-Hop positioning algorithm, particle swarm algorithm, improved particle swarm algorithm, and artificial fish swarm hybrid optimization algorithm based on particle swarm, was analyzed. Set the simulation environment as a 100 m*100 m*100 m three-dimensional space edge, randomly generate 120 nodes in the initial state, evenly distribute the number of beacon nodes in the space to 30, the number of particle populations to 30, the communication radius of each node to 25m, the maximum iteration times to 150, and the maximum particle speed to 50m/s. Figureure 1 is a three-dimensional node distribution diagram of a wireless sensor network in a simulation experiment. The red "+" in the diagram represents beacon nodes, and the blue "□" represents unknown nodes. **Figure 2** shows the error of a single unknown node in a simulation experiment of wireless sensor networks.

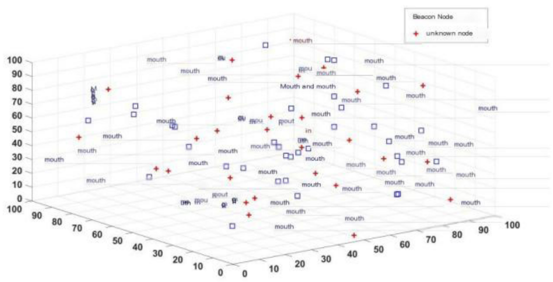


**Figure 1.** Three-dimensional positioning space node distribution map.

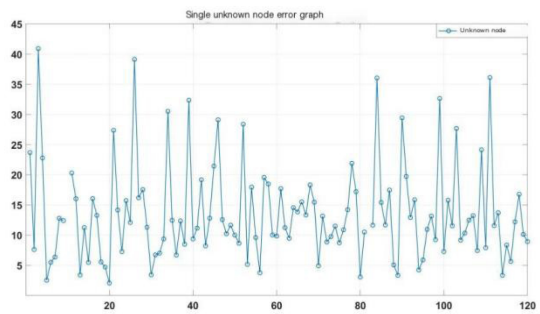


**Figure 2.** Error diagram of a single unknown node in a simulation test.

The analysis of the relative positioning error on the positioning algorithm through simulation testing mainly includes three parts: comparative analysis of relative positioning error under different anchor node ratios, comparative analysis of the impact of different node communication radii on positioning error, and comparative analysis of the impact of different node numbers on positioning error.

## 7.1. Comparative analysis of the relationship between the proportion of beacon nodes and positioning error

The total number of nodes in the network is set to 120, the communication radius is set to 25 meters, and the proportion of beacon nodes is increased from 0.05 to 0.35 in steps of 0.05, resulting in four simulation graphs of relative positioning error for four positioning algorithms. As the proportion of beacon nodes increases, the relative positioning error for all four algorithms gradually decreases, and the decreasing trend gradually becomes smoother. This is because when the proportion of beacon nodes is small, there are fewer beacon nodes in the network that are connected to unknown nodes, and the unknown nodes receive insufficient reference data, resulting in large positioning errors. As the proportion of beacon nodes increases, the number of beacon node information received by unknown nodes increases, and the positioning error decreases.

As shown in **Figur 3**, the relative positioning error of the PSO algorithm is reduced by about 12.5% compared to the original DV-Hop algorithm, and the positioning error of the RWPSO algorithm is reduced by about 18.1% compared to the PSO algorithm. As the proportion of beacon nodes increases, the improved RWPSO-AFSA algorithm in this paper reduces the positioning error by about 1.3% compared to the RWPSO algorithm. As the proportion of beacon nodes increases, the positioning error gradually decreases, indicating that the positioning algorithm proposed in this paper has achieved significant improvement.

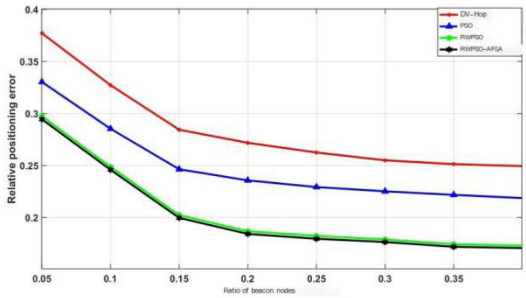


**Figure 3.** Beacon node ratio and positioning error relationship diagram.

### 7.2. Comparative analysis of the relationship between node communication radius and positioning error

The number of nodes is 30, and the total number of unknown nodes is 120. The communication radius of nodes has increased from 30m to 60m,

From the error comparison diagram, it can be seen that when the communication radius of the node increases, the relative positioning error of the four positioning algorithms will gradually decrease. When the communication radius is small, there are more hops between the unknown node and the beacon node, which increases the cumulative error when calculating the distance. Additionally, some nodes may not be able to locate due to not receiving the broadcast information, resulting in a larger positioning error. As the communication radius increases, the number of hops between nodes decreases, and the cumulative error also decreases. Moreover, previously isolated nodes may receive the broadcast information and achieve positioning, resulting in a reduction in positioning error.

As shown in **Figure 4**, the relative positioning error of the PSO algorithm is reduced by about 9.3% compared to the original DV-Hop algorithm, and the positioning error of the RWPSO algorithm is reduced by about 3.2% compared to the PSO algorithm. As the proportion of beacon nodes increases, the improved RWPSO-AFSA algorithm in this paper reduces the positioning error by about 2.1% compared to the RWPSO algorithm. As the communication radius of nodes increases, the positioning error gradually decreases, indicating that the improvement effect of the positioning algorithm proposed in this paper is significant.

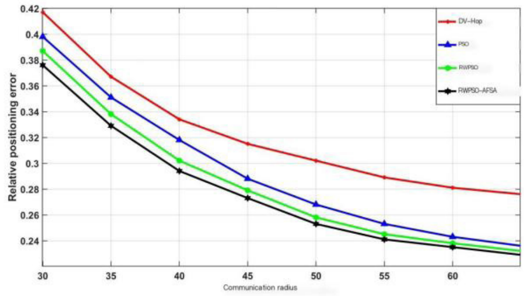


**Figure 4. N**ode and positioning error relationship diagram.

### 7.3. Comparative analysis of the relationship between the number of nodes and positioning error

The number of beacon nodes is fixed at a ratio of 0.3, and the communication radius of the nodes is 25m. The total number of unknown nodes increases from 120 to 250. From the Figureure, it can be observed that the positioning error of the four positioning algorithms decreases with the increase in the total number of nodes, and the decreasing trend becomes gradually smooth. This is because as the total number of nodes increases, the number of anchor nodes that exchange information with unknown nodes also increases, allowing some previously isolated unknown nodes to receive more broadcast information from beacon nodes, reducing the positioning error of the algorithm.

From **Figure 5**, it can be seen that the relative positioning error of the PSO algorithm is reduced by about 15% compared to the original DV-Hop algorithm, and the positioning error of the RWPSO algorithm is reduced by about 14.4% compared to the PSO algorithm. As the proportion of beacon nodes increases, the improved RWPSO-AFSA algorithm in this paper reduces the positioning error by about 1.4% compared to the RWPSO algorithm. As the total number of nodes increases, the positioning error gradually decreases, indicating that the positioning algorithm proposed in this paper has significant improvement effects.

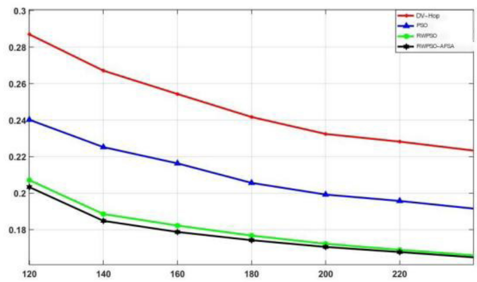


**Figure 5.** The number of nodes and positioning error relationship diagram.

## 8. Conclusion

In order to improve the performance of the traditional DV-Hop localization algorithm, the standard particle swarm optimization algorithm is introduced to estimate the location of nodes, but the localization estimation effect is not good. Therefore, this paper proposes a fusion algorithm of improved particle swarm optimization algorithm and DV-Hop. By improving the calculation of speed, inertia weight, and learning factor in the improved algorithm, the localization accuracy and stability are better, while maintaining the hardware cost and communication overhead unchanged. To a certain extent, it solves the problem of poor performance of the PSO-based DV-Hop algorithm. The particle swarm algorithm has a slow convergence rate in the later stages of iteration and is prone to falling into local optima. However, during the iterative process, the artificial fish swarm algorithm has a good global convergence rate and can quickly search locally, avoiding local optima and facilitating the rapid finding of optimal solutions. Inspired by the advantages of the artificial fish swarm algorithm and the particle swarm optimization algorithm, a node localization algorithm based on the improved particle swarm algorithm and artificial fish swarm algorithm (RWPSO-AFSA) is proposed by integrating the random weight particle swarm and artificial fish swarm algorithms. The position update formula of the particle swarm algorithm is improved. Although the algorithm has been improved by adding a small amount of computation to the calculation of mutation factors, there is no increase in the number of iterations during the operation, and the time complexity is at the same level as the original algorithm.

This article focuses on theoretically improving the analysis of wireless sensor network node localization algorithms, ignoring the fact that in practical applications, the number and energy of node sensors are usually limited, and the algorithm's computation time and speed can affect the stability and lifespan of the entire network system. Future research should focus more on reducing the computational load and energy consumption of algorithms.

## Acknowledgments

This study was supported by The Science and Technology Project of Housing and Urban-Rural Construction of Anhui Province in 2022, Number: 2022-YF044.

## About the author

Shaowei Sun was born in Chuzhou, Anhui, P.R. China, in 1988. He received the Master's degree from Anhui Polytechnic University, P.R. China. Now, he is pursuing a PhD degree at the School of Mechanical Engineering, Hefei University of Technology. His research interest include Intelligent transportation and informatization engineering.

## References

[1] Zhou Wenbo, Zhang Yong, Sun Liangyi, and Su Jun. Research on an Improved Node Positioning Algorithm for Wireless Sensor Networks[J]. Journal of Marine Electronics Engineering, 2021(05):53-57

[2] Hatami, K.Pahlavan, M.Heidari, F.Akgul.On RSSI and TOA Based Indoor Geolocation-A Comparative Performance Evaluation[J]. IEEE Wireless Communications and Networking Conference, 2006: 2267-2272.

[3] X.Cheng, T. A, D.Chen.TPS:a time-based positioning scheme for outdoor wireless sensor networks[C]. International Conference on Computer Communications. IEEE, 2004: 2685-2696.

[4] D.Niculescu, B.Nath.Ad Hoc Positioning System(APS) Using AOA[C]. IEEE INFOCOM, San Francisco, 2003: 1734-1743.

[5] Wang J, Urriza P, Han Y, et al. Weighted Centroid Localization Algorithm: Theoretical Analys is and Distributed Implementation[J]. IEEE Transactions on Wireless Communications, 2011, 10(10):3403-3413.

[6] Kumar S, Lobiyal D K. An Advanced DV-Hop Localization Algorithm for Wireless Sensor Networks[M]. ICIEA, Kluwer Academic Publishers, Wireless Personal Communications, 2013:1557-1561.

[7] Xiu-wu Y U, Hao Y U, Yong L, et al. APIT Location Algorithm Based on Weighted Hybrid Filtering and Center of Gravity Method[J]. Journal of beijing university of posts and telecommunications, 2019, 42(04), 32-37.

[8] CHEN Min, WANG Bo, LI Jun-hua. Wireless Sensor Network Theory and Practice [M]. Beijing: Chemical Industry Press, 2011.

[9] Liu Chuanzhou, Zhang Linghua. Optimization of positioning accuracy in wireless sensor networks based on DV-Hop algorithm [J]. Progress in Laser and Optoelectronics, 2021, 58(22): 498-504.
[10] Liang Yuming, Pei Xinghuan. Particle Swarm Optimization Artificial Fish Swarm Algorithm [J]. Computer Simulation, 2016, 33(06): 213-217+281.
[11] Localization in 3D Sensor Networks Using Stochastic Particle Swarm Optimization[J]. Wuhan University Journal of Natural Sciences, 2012, 17(06):544-548.
[12] Lou Guohong, Zhang Jianping Wireless sensor network node localization using particle swarm optimization for ranging correction [J] Journal of Jilin University (Science Edition), 2018, 56(03):650-656.
[13] Gao Yun Research and improvement of DV-Hop-based wireless sensor network localization algorithm [D]. Xi'an University of Electronic Science and Technology, 2020.
[14] Wang Lianguo, Shi Qiuhong, Hong Yi PSO AFSA hybrid optimization algorithm [J] Computer Engineering, 2010, 36(5): 176-178.
[15] Ren Xiaokui, Li Feng, Cheng Lin. An improved centroid localization algorithm based on dynamic loss factor and weight [J]. Computer Application, 2019(03): 824-828.