Improved Chaos Particle Swarm Optimization Algorithm For Wireless Sensor Networks Node Localization

Hongyi Cao*, Nan Feng, Junhui Yang

School of Xi'an Medical University, Xi'an 710021, China

Abstract

In the wireless sensor network localization algorithm, in order to reduce the positioning error and improve the positioning accuracy, an optimized localization algorithm based on adaptive inertia weight is proposed, which combines the wireless signal propagation path loss model and the improved particle swarm algorithm. First, estimate the distance between the unknown node and the wireless network node according to the wireless signal propagation path loss model. And then use the improved particle swarm algorithm to do the later optimization. According to the distance between the particle position and the global optimal position after each iteration, the inertia weight of the particle is dynamically adjusted to make it have dynamic adaptability. And use the evolution rate as the search termination condition to speed up the convergence rate of the algorithm. The simulation results show that the adaptive inertial weight algorithm can reduce the average positioning error and improve the positioning accuracy of the nodes in wireless sensor networks compared with the PSO algorithm and the wireless signal propagation path loss model algorithm based on the improved PSO algorithm.

Keywords: wireless sensor network, node localization, particle swarm algorithm, adaptive inertia weight.

1. INTRODUCTION

Wireless sensor networks (co WSN) consist of a large number of inexpensive micro-sensor nodes with sensing, computing and communication capabilities, which perform functions such as data acquisition and processing as well as communication (Johansson, 2014; Huang et al., 2014; Chen et al., 2015). Functions such as tracking, monitoring and positioning objects in complex environments are extensively used, and the precise positioning of the sensor nodes is a supporting technology for wireless sensor network applications. At present, the sensor node location methods are mainly divided into two kinds, namely, distance-dependent and distance-independent. The positioning method without distance measurement is usually carried out through the network connectivity and some other network information for the location of the nodes, and the positioning accuracy is mostly very low. Currently, distance-related positioning methods mainly include receiving signal strength indicator (RSSI), signal arrival angle (AOA), signal arrival time (TOA), signal arrival time difference (TDOA) and so on which are node location methods based on the distance between adjacent nodes (Chatterjee and Siarry, 2016; Coelho and Mariani, 2009; Colle et al., 2014). The RSSI ranging location technology does not need other hardware devices compared with other ranging location methods and is widely used in large-scale wireless sensor networks because of its low cost and low power consumption. However, the changes of the surrounding environment, the difference of the obstacles and the unknown transmission channel loss model causes unsatisfactory node localization accuracy in the actual environment. Aiming at this problem, the positioning accuracy can be improved based on RSSI by adopting intelligent optimization algorithm (He et al., 2010; Kung et al., 2009; Li et al., 2011; Li et al., 2005).

Based on the application characteristics of unknown environment in wireless channel loss model, this paper designs a localization algorithm of wireless sensor network based on chaos particle swarm optimization (PSO) based on adaptive inertia weight for the shortcomings of wireless network node location technology and RSSI ranging location calculation method. Based on the chaos particle swarm optimization function, the node localization problem is abstracted as a nonlinear constrained optimization problem. By constructing the weighted location objective function, the particle swarm optimization (PSO) technology is used to estimate the position of the node to be located.
2. ANALYSIS OF PROPAGATION PATH LOSS MODEL OF UNKNOWN WIRELESS SIGNAL

The wireless signal propagation path loss model changes in different positioning environments, therefore, it is impossible to compute the node spacing based on RSSI directly and the signal loss model in the ideal environment. However, the problem can be understood as a nonlinear constrained optimization problem. In the wireless sensor network composed of n nodes, there are two adjacent unknown node A and beacon node B, the location coordinates of A is \((x_a, y_a)\), the location coordinates of B is \((x_b, y_b)\), the space between two nodes is \(d\), then the unknown node A satisfies the formula (1):

\[
\begin{align*}
L_{ab} &= L_{ob} - L_{ab} \\
L_{ab}(d)_{ob} &= L_{ab}(d)_{ob} + 10\gamma \frac{d}{d_o} + X_{\psi}(dB) \\
d &= \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}
\end{align*}
\]

In formula (1), \(L_{ab}\) is the wireless signal loss observation value of unknown node A relative to neighboring beacon node B. The wireless signal loss \(L(d)_{ob}\) at location \(d_o\) can be calculated by formula (2):

\[
L(d)_{ob} = 10\lg \frac{G_iG_d2^2}{(4\pi)^2 d_o^2 L}
\]

And the predicted signal strength \(\epsilon_{ab}\theta_a\) of unknown node A satisfies:

\[
S_{ab}(\theta_a) = 10\gamma \frac{d}{d_o}
\]

Then formula (1) can be expressed as:

\[
L_{ab} = S_{ab}(\theta_a) + X_{\psi}(dB)
\]

If \(\epsilon_{ab}\theta_a\) is the deviation of the estimated signal strength \(\epsilon_{ab}\theta_a\) from the actual observation value \(L_{ab}\), then the total deviation \(\epsilon_a(\theta)\) between the predicted signal strength of the unknown node A and the actual observed value in this wireless sensor network is:

\[
\epsilon_a(\theta) = \sum_{i=1}^{n} |L_{ob} - S_{ab}(\theta_a)|
\]

Then, in the case of unknown wireless signal propagation path loss model, the location model of unknown node A based on RSSI can be abstractly described as the objective function of equation (6):

\[
f(\theta) = \min(\epsilon_a(\theta)) = \min(\sum_{i=1}^{n} |L_{ab} - S_{ab}(\theta_a)|)
\]

3. CHAOTIC PARTICLE SWARM OPTIMIZATION

3.1 Particle Swarm Optimization (PSO) is a kind of evolutionary computing technology proposed by Dr. Eberhart and Kennedy in 1995, which is derived from the study of bird population movement behavior. Similar to the genetic algorithm, the PSO algorithm is also an iterative optimization algorithm. Using the concept of "population" and "evolution", the system first initializes a set of random solutions and then iterates through some way to find the optimal solution. Unlike other genetic algorithms, particle swarm optimization does not use evolutionary operators for individuals as with other evolutionary algorithms, but rather considers each individual as a particle without weight and volume in the d-dimensional search space. The particle flies at a certain speed in the search space, and the flight speed of it is adjusted dynamically by the
combination of individual flight experience and the group’s flight experience (Chekroun et al., 2014; Victoria and Raju, 2014; Abdullin et al., 2014)

Because of the lack of mutation mechanism, the search particle of particle swarm optimization algorithm is extremely vulnerable to local extremum in the search process. And that the particle itself can not escape the constraints of the local extremum leads to premature algorithm (Krishnaveni and Raju, 2014; Sudheer and Raju, 2014). Chaotic particle swarm optimization algorithm introduces the chaos thought into particle swarm optimization algorithm. It can update the position and velocity of the particles by using the unique ergodicity and randomness of the chaotic variables, so that the search particles do not premature at a certain local optimum point, which will avoid the algorithm convergence. Suppose there are Nα flying particles in the D-dimensional search space, the position vector of these particles is \( X_α = (x_{α1}, x_{α2}, \ldots, x_{αD}) \), the velocity of the particle is \( V_α = (v_{α1}, v_{α2}, \ldots, v_{αD}) \), the individual extremum of particle \( α \), namely, the optimal position of the particle itself, is \( p_{Best_α} = (p_{α1}, p_{α2}, \ldots, p_{αD}) \), the global extremum of the particle , namely, the optimal location of the chaotic system is . The real-time equation of the position and velocity of particle \( α \) in the updating process is:

\[
v_α(t + 1) = \omega v_α(t) + c_1 \text{rand}_1(p_{Best_α}(t) - v_α(t)) + c_2 \text{rand}_2(g_{Best_α}(t) - v_α(t))
\]

\[
x_α(t + 1) = x_α(t) + v_α(t + 1)
\]

\[
ω = ω_{max} - \left(\frac{ω_{max} - ω_{min}}{T}\right)
\]

\[
y(n + 1) = λy(n)(1 - y(n))
\]

\[
x_{ad} = \min_d + (1 + y_a(d))(\max_d - \min_d) / 2 \quad a = 1,2,\ldots; d = 1,2,\ldots,D
\]

In the above formula, \( ω \) is the inertia weight factor whose role is to maintain a balance between global and local search ability; \( c_1 \) and \( c_2 \) are the learning factors, which can show the acceleration weight of each particle which evolves to the individual extremum and global extremum; usually the learning factors are in the range of 0 to 2; \( \text{rand}_1 \) and \( \text{rand}_2 \) object to two independent random numbers which are evenly distributed in the \([0,1]\) interval; \( λ \) is the control parameter of the chaotic map which is always in the range of 3.5 to 4; In the D-dimensional space, the iterated particle carries out the N-1 iterative operation through equation (11), it also produces N-1 particles, \( y_a(d) \) is the d-th dimension of the \( α \)-th particle produced in the chaotic map, \( x_{ad} \) is the optimal feasible solution of the optimization algorithm.

It can be seen from the above formula that in the optimization process, the algorithm first chooses the optimal particle in the population for chaos optimization, and then randomly selects the other particles’ chaotic optimal location information from the population randomly to replace it, which avoids the population to fall into prematurity and stagnate. But the algorithm is bound to cause the slow convergence rate, and if it still adjusts with the previous linear progressive decrease, it will lead to the decrease of the algorithm performance. Therefore, how to accelerate the convergence rate of the algorithm to early jump out of the local optimal will be the key to improving the algorithm performance.

3.2 Chaos particle swarm optimization algorithm (CPSO)

In order to overcome the shortcomings of PSO-based optimization algorithms such as easiness to fall into local extremum points, slow convergence rate and poor accuracy in the late evolutionary period, we introduce chaos optimization into PSO. Chaos optimization is a novel optimization method, which utilizes the unique ergodicity of chaotic system to achieve global optimization, and it does not require the objective function to have continuous and differentiable properties (Krishnaveni and Raju, 2014; Sudheer and Raju, 2014).

In general, the motion state with randomness obtained by the definite equation is called chaos, and the chaotic state variable is called chaos variable. For example, the formula \( Z_{n+1} = μZ_n(1 - Z_n) \) is a typical chaotic system.
In the formula, \( \mu \) is the control parameter, when the value of \( \mu \) is determined, an arbitrary time series \( Z_1, Z_2, Z_3 \ldots \) can be iterated out from any initial value \( Z_n \in [0, 1] \).

A chaotic variable in a certain range has the following characteristics: randomicity, that is, its performance is as random as the random variables; ergodicity, that is, it can experience all the states within the space without repetition; regularity, that is, the variable is determined by the iteration equation. The main measure of Chaos particle swarm optimization algorithm is to use the ergodicity of chaos motion and produce the chaotic sequence based on the optimal location obtained so far by the current whole particle swarm, then replace the position of a particle in the current particle swarm by the optimal location particle of the chaotic sequence randomly.

4. ADAPTIVE INERTIA WEIGHT CHAOTIC PARTICLE SWARM OPTIMIZATION ALGORITHM

4.1 Adaptive changes of Adaptive inertia weight (AIW)

The chaos particle swarm adopts nonlinear searching in the real search process. The linear decreasing strategy of the inertia weighting factor \( \omega \) is too simple (Zhang et al., 2010). The adjustment ability to the algorithm is very limited in the complex search process. To solve this problem, an adaptive mutation method of \( \omega \) is introduced to adjust the searching process of the algorithm to speed up the local and global searching ability of the particle. The method combines the inertia weight factor \( \omega \) with the change of the particle’s own fitness value. The basic idea is to make the value of the inertia weight factor in direct proportion to the fitness value of the particle itself. Assuming that the relative change rate of the self-adaptation value of the particle is:

\[
\vartheta = \frac{(\delta_{a}(t) - \delta_{a}(t - 1))}{\delta_{a}(t - 1)}
\]

\[
\omega_{a}(t) = (1 + e^{-\theta})^{-1}
\]

In equation (13), \( \delta_{a}(t) \) is the fitness value of the particle \( a \) in the \( t \)-th iterative process, \( \omega_{a}(t) \) is the inertia weight value of the particle \( a \) in the process of the \( t \)-th generation updating. As can be seen from the above equation, the value of \( \omega_{a}(t) \) increases with the increase of \( \theta \), decreases with the decrease of \( \theta \), and changes within the range 0 to 1. This method can speed up the convergence speed of particle to the global extremum, thus it can accelerate the convergence rate of the algorithm.

4.2 Analysis and improvement of premature convergence

In this paper, the population fitness variance is used to judge the degree of convergence of the particles, and the number of particles in the population is still assumed to be \( N \), \( \delta_{\text{average}} \) and \( \sigma^2 \) are the mean fitness and fitness variance of the population respectively:

\[
\sigma^2 = \sum_{a=1}^{N} \left( \delta_{a} - \delta_{\text{average}} \right)^2
\]

is the normalized scaling factor which is used to control the value of \( \sigma^2 \). \( \sigma^2 \) reflects the particle density of the population, the smaller the \( \sigma^2 \) is, the higher the aggregation degree of the particle, and the closer the algorithm to convergence; otherwise, the particle group is in the search state. Therefore, a threshold value \( \sigma_{\text{min}}^2 \) needs to be set to determine the convergence state. When \( \sigma^2 < \sigma_{\text{min}}^2 \), precocious maturation treatment is needed.

When the particle falls into the premature state, the algorithm stagnates. In this paper, a perturbation is added to the stagnation algorithm by Gaussian perturbation, which gives the particle power to jump out of the local optimum. I.e., randomly pick some dimension of the global optimal particle \( gBeat \) to carry out the Gaussian perturbation as follows:
Gaussian(μ, τ~) is the Gaussian perturbation with the mean value of 0, the standard deviation of τ, τ is also known as the optimal learning rate, the calculation method of it is as follows:

\[ \tau = \tau_{\text{max}} - \left[ (\tau_{\text{max}} - \tau_{\text{min}}) \right] \frac{\text{gen}}{\text{max gen}} \]  

(16)

τ_{\text{max}} and τ_{\text{min}} represent the upper and lower bounds of the learning capability range of τ respectively, gen and max gen represent the current iteration number and the maximum allowable iteration number of the algorithm. Gaussian perturbation can effectively prevent the particles from falling into the local optimum, and the perturbation of the mean optimal position and the global optimal position of the population are always carried out to help the particle jump out of the local optimal position.

### 4.3 Improved algorithm flow

Through the above analysis, the Adaptive Inertia Weight Chaos Particle Swarm Optimization (AIWCPSo) is used to adjust \( c_1 \), \( c_2 \) and \( \lambda \) automatically and adjust the inertia weight factor \( \omega \) adaptively during the running process. Find a more accurate global optimal value on the basis of accelerating the quick research speed of the particle and improve the positioning accuracy of the node. Based on the above basic principles, the basic algorithm flow in this paper is as follows:

![Improved algorithm flow chart](image_url)
5. SIMULATION AND ANALYSIS

In this paper, we use MATLAB to simulate the algorithm, and compare it with traditional particle swarm optimization and chaos PSO. Firstly, the advantages and disadvantages of the adaptive inertia weighting localization algorithm and the convergence of the improved particle swarm optimization algorithm are observed. Then, their influence on node localization error is explained from the four aspects, namely, the number of beacon nodes, the number of unknown nodes, the wireless range of the nodes and the positioning accuracy.

In the simulation, the nodes are deployed in the area of 100m × 100m, which is also the searching area of the particle swarm optimization algorithm. The main evaluation criterion for the localization of wireless sensor networks is the average positioning error whose formula is equation 17.

\[
\text{AverageError} = \frac{\sum_{a=1}^{N} \sqrt{(x_a - x)^2 + (y_a - y)^2}}{NR}
\]  

(17)

\((x, y)\) is the unknown node coordinates which are obtained by calculation, \(a = 1, 2, ..., N\) is the number of unknown nodes, \((x, y)\) is the actual coordinates of the unknown node, \(R\) is the wireless range of the node.

The relevant parameters were set as follows: learning factor \(c_1 = c_2 = 2\), \(\omega = 0.9\), particle maximum velocity \(V_a = 10\), particle population size \(N = 30\).

5.1 In this paper, the adaptive inertia weight chaos particle swarm optimization algorithm, the traditional particle swarm and chaos particle swarm algorithm are compared in terms of the convergence. Set the total number of nodes as 200, of which 20 nodes are wireless network nodes, and the simulation results are shown in Figure 2. It can be seen from Figure 2 that under the same conditions, the algorithm in this paper can find the optimal value after about 40 iterations, and the convergence speed is better than that of the other two algorithms. The faster convergence is due to the fact that after each iteration, the particles can adaptively adjust the flight speed of the next iteration according to their own position. Since the condition for judging convergence is added, it is possible to end the calculation of the return result when iterating 40 times, thereby saving the unnecessary iteration process.

![Figure 2. Comparison of convergence performance](image)

5.2 In the relationship simulation between the average location error and the number of the wireless network nodes, the number of unknown nodes is fixed at 200 and the number of nodes in the wireless network is increased from 10 to 50. The average positioning error obtained by using the three methods is compared. The experimental results are shown in Fig.3. It can be seen from the figure that with the increase of the number of beacon nodes, the average positioning error will be reduced in these three methods. The average positioning error of AIWCPSO in this paper is obviously better than that of PSO algorithm, and is also slightly better than that of CPSO algorithm. This is because the adaptive inertia weighting algorithm can find the optimal value according to its own condition, and the global search ability is stronger.
5.3 The relationship between the average location error and the number of unknown nodes. Figure 4 compares the effect of the number of unknown nodes on the average positioning error when the number of nodes in the wireless network is constant. In simulation, the number of beacon nodes is fixed at 20, and the number of unknown nodes is increased from 150 to 400. It can be seen from Figure 4 that with the increase of the number of unknown nodes, the positioning error will be increased, but the AIWCPSO algorithm in this paper reduces the occurrence probability of the local optimal value due to the adaptability of the particle when searching the space, which can better search for the global optimal value in the space, so the positioning error is the minimum.

5.4 The relationship between the average positioning error and the wireless range. In the case of a certain number of nodes, compare the impact of the wireless range of the nodes on average positioning error. In the experiment, the total number of nodes is 200, in which the number of beacon nodes is 20, and the wireless range of nodes increases from 10m to 50m. It can be seen from Fig.5 that the average localization accuracy of adaptive inertial weight algorithm is obviously superior to that of CPSO algorithm and localization algorithm based on PSO. With the increase of wireless range, the localization errors of these two algorithms are reduced. The algorithm in this paper can reduce the influence caused by the error of the estimated distance from the first two steps of the CPSO algorithm to a minimum, so the effect is better.
5.5 In Figure 6, the abscissa is the wireless network node, and the ordinate is the average positioning error of each wireless network node. It can be seen from the figure that the node localization accuracy optimized by AIWCPSO algorithm is obviously improved compared with that of CPSO algorithm and PSO algorithm. From the further comparative analysis from respects of the error mean, the error variance, and the calculation amount, we can get Table 1.

![Figure 6. Comparison of the average localization errors of the three algorithms](image)

<table>
<thead>
<tr>
<th>Positioning algorithm</th>
<th>Mean Positioning Error</th>
<th>Mean Calculation (s)</th>
<th>Mean Positioning Error Variance</th>
<th>Calculation amount(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO</td>
<td>3.1413</td>
<td>0.3251</td>
<td>0.3750</td>
<td></td>
</tr>
<tr>
<td>CPSO</td>
<td>1.5355</td>
<td>0.1675</td>
<td>1.5197</td>
<td></td>
</tr>
<tr>
<td>AIWCPSO</td>
<td>0.5265</td>
<td>0.1058</td>
<td>0.8291</td>
<td></td>
</tr>
</tbody>
</table>

6. CONCLUSION

Based on the CPSO algorithm, a localization algorithm based on adaptive inertia weight is proposed in this paper. Based on the model of wireless signal propagation path loss model and the improved particle swarm optimization algorithm, the inertia weight of the particle is dynamically adjusted according to the distance between the particle position and the global optimal position after each iteration to make it have dynamic adaptability. The adaptive inertia weight algorithm can effectively reduce the probability of local optimal solution, and can find the global optimal solution in space quickly. Compared with the standard PSO algorithm and the improved CPSO method, the adaptive inertia weighting positioning algorithm has the characteristics of simple algorithm, fast convergence speed, strong global optimization ability and less control parameters. The simulation results show that the algorithm has a certain improvement in the node location accuracy.

REFERENCES


