RFID Indoor Positioning Based on AP Clustering and Improved Particle Swarm Algorithm

Zhang Manman,1 Li Peng,1,2 Xu He,1,2 and Wang Ruchuan1,2

1School of Computer Science & Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
2Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Jiangsu Province, Nanjing 210003, China

Correspondence should be addressed to Li Peng; lipeng@njupt.edu.cn

Received 3 January 2022; Revised 12 May 2022; Accepted 3 August 2022; Published 25 August 2022

Academic Editor: Bin Gao

Copyright © 2022 Zhang Manman et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a method based on AP clustering and an improved particle swarm algorithm for radio frequency identification (RFID) indoor positioning, called the AP-PSO method. Firstly, an AP clustering algorithm is used to cluster the RSSI values of the experimental region tags with similarity, in order to achieve the division of tagged regions, reduce the search area of the later improved particle swarm algorithm, and reduce the search time. Secondly, the learning factor of the particle swarm algorithm is dynamically adjusted, in order to improve the search ability and convergence speed of the global optimal solution of particles. The experimental results show that the algorithm can effectively achieve RFID indoor positioning of the tags to be measured, with high positioning accuracy and with the algorithm spending less time.

1. Introduction

As the Internet continues to develop, the demand for location-based services (LBS) is growing, and location technology is gaining widespread attention. As part of this trend, RFID has become an important technology for indoor positioning, particularly because of its small size, low price, noncontact characteristics, and nonvisual distance. In practice, RFID is widely used in scenarios such as warehouse logistics, library book order finding, airport baggage location tracking, and medical observations [1–5].

Currently, RFID-based positioning technologies are divided into two categories, namely, range-based and non-range-based. Non-range-based technologies mainly lie in theoretical research, and their positioning accuracy is low. Range-based methods are mainly based on time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA), and received signal strength indicator (RSSI). The positioning techniques, which are based on signal transmission time and angle, require a high time angle and are less studied. The RSSI-based ranging method has the characteristics of simple operation, low overhead, and low cost and has become the mainstream research direction of indoor positioning technology [6–9].

Most RSSI-based RFID indoor positioning is based on the LANDMARC system, which was originally proposed in 2003. Although the system’s positioning accuracy is only about one meter, which makes it difficult to meet the demand for indoor positioning, the system introduced the concept of reference tags for the first time. A number of reference tags are arranged around the tags to be measured; the RSSI value of the reference tag is measured, and the position of the reference tag is recorded. The RSSI value of the tag to be measured is compared with the RSSI value of the reference tag, and the position of the reference tag with the smallest difference in RSSI value is selected as the position of the tag to be measured. This is the localization concept behind the LANDMARC system. When the system was first proposed, the positioning accuracy was not high enough to meet the needs pertaining to indoor positioning. Later, researchers explored methods to improve the positioning accuracy, based on the system. Indoor clustering is one of the key techniques used for mobile trajectory analysis; this technique can discover the movement patterns, location
preferences, and hotspot paths of indoor objects [10–14]. Peng [15] represented the RSSI distribution area by combining the interval number clustering representation with the statistical information of RSSI values, in order to reduce the distance estimation error caused by the uncertainty of RSSI values. Li et al. [16] proposed a K-means clustering algorithm for indoor localization based on signal strength. This approach improves the complexity of data clustering center selection, but when the differences between data are relatively large, the method may result in large subclasses being oversegmented, instead of reducing the localization accuracy. A K-means algorithm needs to be artificially given the initial clustering center, so multiple initial values have to be tried. It is also easy to fall into the local optimum, which affects the positioning accuracy. Ji et al. [17] proposed an improved K-means algorithm to solve this problem, but this algorithm also requires the number of K clusters to be given in advance. This is difficult to judge, and if the user chooses an incorrect number of clusters, the data that should be in the same class will be judged as being in two large classes, which eventually affects the positioning accuracy. Mao et al. [18] used the BKM clustering algorithm to optimize the localization method and improve the localization accuracy. However, the BKM clustering algorithm, like the classical K-means clustering algorithm, needs to randomly determine the number of clusters before clustering, which causes the clustering results to be easily affected by the number of clusters. The shortcomings of the K-means algorithm have led researchers to turn to the exploration of other clustering algorithms. Lan and Xiao [19] proposed a grid-based density peak clustering algorithm (GDCA), which combines the advantages of grid clustering algorithm and density peak clustering algorithm. The GDCA is able to handle complex clustering information while processing large spatial data sets, thereby enabling the positioning results obtained in dealing with RFID-based indoor positioning problems to meet practical needs. Yang [20] proposed a hierarchical clustering algorithm and introduced Spark’s distributed computing framework to parallelize the clustering algorithm, in order to improve the computational speed of the algorithm. The proposed algorithm can effectively perform indoor trajectory localization clustering and greatly improve the time efficiency, but the method is more complicated to implement.

In addition, RSSI-based indoor positioning problems can be transformed into area optimization problems, where classical optimization algorithms are applied to indoor positioning. For example, the gray wolf algorithm, cat swarm algorithm, and particle swarm algorithm, Xu et al. [21] proposed an RFID localization algorithm based on Newton interpolation and a hybrid gray wolf optimized SVR. The method uses Newton interpolation to value the signal in the rest of the indoor experimental area and adds it to the signal strength value dataset. Then, HGWOSVR is used to construct a nonlinear relationship between the signal’s received strength and location information to achieve localization. The algorithm can reduce the number of reference tags, but some of the data are formed by fitting, and there are errors that can affect the RFID positioning performance. Li et al. [22] proposed an indoor localization method based on fuzzy clustering and a cat colony algorithm for the problem of reference point mismatching and location search in a RSSI indoor localization algorithm. The method first uses fuzzy clustering to cluster the RSSI values of reference points and then uses a cat colony algorithm to perform a location search in the region. The method innovatively adds a feeding mechanism to the cat colony algorithm to achieve fast location search. However, the problem whereby the algorithm is prone to fall into local optimum is not considered, which also affects the localization accuracy. Chen et al. [23] used a particle swarm algorithm to optimize the LANDMARC localization process. This method can filter out the optimal neighbor tags with high localization accuracy, but it is easy for the algorithm to converge to local optimum, and convergence speed is slow. Jiang [24] proposed an improved particle filter-based RFID indoor localization algorithm to address the particle swarm algorithm’s problems of slow convergence and easily falling into local optimum. The algorithm first uses an extreme learning machine (ELM) to fit the nonlinear relationship between the received signal strength of the reader and the tag distance. Then, the adaptive learning factor is used to improve the particle filtering process, in turn improving the global optimization ability and convergence speed of the particles. The method can also improve the localization accuracy and convergence speed. However, the nonlinear relationship between the signal strength received by the reader and the distance to the tag when using an ELM is not the actual measured distance relationship; there is a certain amount of error, resulting in RFID positioning errors.

As mentioned above, most of the clustering algorithms used for indoor localization are K-means and improved K-means. However, the algorithms are required to artificially specify the initial clustering center, which will inevitably lead to errors caused by subjective factors, thus affecting the positioning accuracy. Another category of RFID positioning technology is mainly focused on a variety of optimization algorithms. The particle swarm algorithm, compared to other optimization algorithms, has simple parameters, simple methods, and the advantages of parallelization, but the algorithm also has the problems of easily falling into the local optimum and slow convergence. These problems affect the accuracy of positioning and require longer positioning time. To address the above problems, this paper proposes an AP-PSO-based localization algorithm. Firstly, an AP clustering algorithm is used to cluster the RSSI values of the experimental area tags with similarity, in order to achieve the division of the tagged area, reduce the search area of the later improved particle swarm algorithm, and reduce the search time. Secondly, the learning factor of the particle swarm algorithm is dynamically adjusted to improve the search ability and convergence speed of the global optimal solution of the particles. Moreover, the AP clustering algorithm does not need to define the center value of clusters in advance but automatically determines the center of clusters according to the similarity between data points. In addition, the clustering results are objective and reduce human interference. The experimental results show that the
algorithm can effectively achieve RFID indoor positioning of the tags to be measured, with high positioning accuracy and less time spent by the algorithm.

1.1. Contributions of This Article

(1) To address the problems whereby the particle swarm algorithm tends to fall into local optimum and converge slowly at the later stage of the search, the learning parameters of the particle swarm are improved. In addition, the learning parameters are dynamically adjusted, in order to improve the search ability and convergence speed of the global optimal solution of the particles and to improve the localization accuracy and reduce the time required for localization.

(2) The AP clustering algorithm is used to cluster the RSSI values of the experimental area tags, and the clustering results are objective. This reduces the interference of human factors and facilitates the improvement of the localization accuracy of the later particle swarm algorithm.

(3) This research is the first step to use clustering of the experimental area tags to achieve tagged area division, reduce the later improved particle swarm algorithm search area, reduce the search time, and improve time efficiency.

2. RFID System Architecture, Working Mechanism, and Indoor Positioning Model

2.1. RFID System Architecture. The traditional RFID model mainly includes three modules: reader, tag, and data monitoring program, as shown in Figure 1.

The reader is a device for reading and processing tag information. This module mainly consists of three parts: control, radio frequency, and antenna. A reader can be used as an independent part, or it can be formed into different parts and embedded in various systems. In the RFID system, the reader can be connected to the radio frequency tags to obtain and process the data in those tags. The half-duplex or full-duplex mode can be used to complete information transmission between the reader and the tag. The carrier frequency band of the RFID system can be divided into three frequency bands: low, high, and ultrahigh. The low frequency band has strong stability and low requirements in terms of distance and cost. Low frequency is often used in the design of antitheft systems, access control systems, simple management systems, and toys. High frequency (13 MHz) bands are mainly used to complete electronic payments, bus card swiping, advanced door lock management systems, etc. Ultrahigh frequency (435 MHz or 850 MHz ~ 900 MHz) can cover communication ranges of 10 meters and above, and the frequency of information transmission is high [7, 25].

2.2. RFID Working Mechanism. The reader can use an antenna to transmit radio frequency data. When the tag is within working range, an induced current will appear, and the reader will be activated after obtaining energy. The tag uses the antenna to transmit the saved information to the reader. After the reader obtains the tags data, the reader uses modulation and decoding to complete the subsequent processing of the information. The communication and transmission modes between the reader and the tag can be divided into electromagnetic induction and backscattering, and a frequency alternating field is used to complete the coupling [26–28], as shown in Figure 2.

2.3. Introduction to Positioning System. At present, RFID positioning systems mainly include a positioning system based on the distance-loss model, a positioning system based on VIRE, and a positioning system based on LANDMARC. Each of the three positioning systems has its own characteristics. The LANDMARC positioning system is the current research hotspot [29, 30]. The positioning method proposed in this paper is also based on the LANDMARC positioning system. The scene layout of the LANDMARC positioning system first needs to arrange the reference tags and the reader and record their coordinates. When the tag in the reader’s range receives the RF signal sent by the reader, a response signal will be returned to the reader, to obtain the tag. Assume that the RSSI value of the tags to be located is $Q = (Q_1, Q_2, \ldots, Q_q)$, and $q$ is the number of tags to be measured. In this paper, $q = 10$, and the RSSI value of the reference tags is $H = (H_1, H_2, \ldots, H_n)$, where $n$ is the number of reference tags. Then, the Euclidean distance between the tags to be positioned and the reference tags $H_i$ can be expressed as

$$D_{eq} = \sqrt{(H_i - Q_q)^2} \quad i \in \{1, 2, 3 \cdots n\}; \quad q \in \{1, 2, 3 \cdots 10\}. \quad (1)$$

Take $M$ reference tags that have the closest signal intensity to the tags to be located, and the actual coordinates of the tags to be located can be obtained:

$$(X, Y) = \sum_{m=1}^{M} W_m(x_m, y_m),\quad (2)$$

in $W_m = (1/D_{eq}^2)/(\sum_{m=1}^{M}(1/D_{eq}^2))$.

The LANDMARC positioning system introduced the concept of reference tags for the first time. According to the difference in signal strength between the reference tags and the target tags, the actual coordinates of the target tags are calculated. At the same time, RFID tags are inexpensive and reduce the cost of the system. However, this positioning system also has some problems, namely, that the positioning accuracy does not meet the demand [31].

3. Indoor Positioning Algorithm Implementation

3.1. Introduction to AP Clustering Algorithm. The AP clustering algorithm, which is a clustering algorithm based on “message passing” between data points, was proposed by
Frey Equals in 2007. The algorithm does not need to determine the number of clusters first but treats all data points as potential cluster centers. Then, the algorithm finds the most suitable cluster centers through message passing between nodes. Different from the traditional K-means clustering algorithm, in the AP clustering algorithm, all data points can be used as the clustering center, so there is no need to manually select the initial clustering center. At the same time, unlike the KNN algorithm, the algorithm can adjust its own reference to control the granularity of the clustering results; there is no need to manually determine the number of clusters before positioning [32–34].

3.1.1. The Basic Principle of the AP Clustering Algorithm. For the AP clustering algorithm, suppose the data sample set $H = \{H_1, H_2, \ldots, H_n\}$, and $H$ denotes the set of tag signal intensity values RSSI, where $n$ indicates the number of samples and $n$ is the number of tags. The AP clustering algorithm uses negative Euclidean distance to represent the similarity $s$ between $n$ data points. In this paper, the negative Euclidean distance $s$ is used to indicate the similarity of the signal intensity values between tags. The calculation of similarity is shown in

$$s(i, k) = -\sqrt{(H_i - H_k)^2}.$$  \hspace{1cm} (3)

The similarity between $n$ data points forms an $n \times n$ similarity matrix $S$. The value $s(k, k)$ on the diagonal of the $S$ matrix is used as a criterion to judge whether point $k$ can become a cluster center. The larger the value is, the greater is the possibility of $k$ points becoming cluster centers, which is also called the reference degree $p$ (preference). The AP algorithm finds clustering centers by passing two types of messages: Responsibility ($R(i, k)$) and Availability ($A(i, k)$). The Responsibility and Availability values of each data point are continuously updated by iteration, until $f$ high-quality cluster centers (exemplar) are generated. Meanwhile, the remaining data points are assigned to the corresponding clusters [35–37]. The Responsibility and Availability message delivery process is shown in Figure 3.

(1) Parameter Description.

(1) $R(i, k)$: $R(i, k)$ denotes the numerical message sent from point $i$ to candidate clustering center $k$, reflecting the suitability of point $k$ as a clustering center for point $i$

(2) $A(i, k)$: $A(i, k)$ denotes the numerical message sent from candidate clustering center $k$ to $i$, reflecting the suitability of point $i$ to select point $k$ as its clustering center

(3) Damping factor ($\alpha$): the main role is to converge the algorithm
Exemplar indicates the center of clustering

\[ S \text{(similarity)}: \text{the similarity between the } i\text{th data point and the } k\text{th data point is denoted as } s(i,k), \text{which indicates the similarity of the } k\text{th point to as the clustering center of the } i\text{ point} \]

\[ P \text{(preference): the reference degree of data point } k \text{ is noted as } p(k) \text{ or } s(i,k) \text{ and denotes the reference degree of point } k \text{ as the clustering center. The larger the value of } p \text{ is, the greater the number of clusters will be obtained. The smaller the value is, the fewer the number of clusters obtained will be} \]

(2) Formula Calculation.

(1) Responsibility: \( R(i, k) \)

**Input:** Input similarity matrix \( S \); damping factor \( \alpha \);

**Output:** \( f \) clustering centers and category members;

1: Set parameter \( p \) value;
2: Initialize \( A(i, k) = 0, R(i, k) = 0 \);
3: while (maxits<500 or convits<50) do
4: Calculation of \( A(i, k) \) and \( R(i, k) \);
5: if \( R(k, k) + A(k, k) > 0 \) then
6: Select \( k \) points as clustering centers;
7: end if
8: Iterative update \( A(i, k) \) and \( R(i, k) \);
9: maxits=maxits+1;
10: convits=convits+1;
11: end while
12: Get \( f \) cluster centers and category members

**Algorithm 1:** AP clustering algorithm: tag RSSI value clustering.

- **Table 1:** Description of key parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Name</th>
<th>Explanatory notes or take value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S )</td>
<td>Similarity matrix</td>
<td>Storage tags similarity</td>
</tr>
<tr>
<td>( p )</td>
<td>Preference</td>
<td>Measuring the number of clusters</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>Damping factor</td>
<td>Algorithm convergence (( \alpha = 0.9 ))</td>
</tr>
<tr>
<td>( O )</td>
<td>Number of particles</td>
<td>20</td>
</tr>
<tr>
<td>( V )</td>
<td>Particle velocity</td>
<td>([-V_{\max}, V_{\max}])</td>
</tr>
<tr>
<td>( X )</td>
<td>Particle position</td>
<td>([-X_{\max}, X_{\max}])</td>
</tr>
<tr>
<td>( c_1 )</td>
<td>Dynamic learning factor</td>
<td>([c_{\min}, c_{\max}])</td>
</tr>
<tr>
<td>( r_1 )</td>
<td>Random number</td>
<td>([0,1])</td>
</tr>
<tr>
<td>( w )</td>
<td>Weighting factors</td>
<td>0.9</td>
</tr>
<tr>
<td>( t )</td>
<td>Number of iterations</td>
<td>([T_{\max} = 500])</td>
</tr>
<tr>
<td>( Q )</td>
<td>RSSI value of the tags to measured</td>
<td>([Q_1, Q_2, \ldots, Q_{10}])</td>
</tr>
<tr>
<td>( F_{\mu} )</td>
<td>Adaptation function</td>
<td>Evaluation of good and bad particles</td>
</tr>
</tbody>
</table>
Input:
1: Similarity matrix $S$, damping factor $\alpha$,
2: Number of particles $O$, Weighting $w$, $V_{\text{max}}$, $V_{\text{min}}$, $X_{\text{max}}$, $X_{\text{min}}$;

Output: The optimal position of the particle $p_{\text{best}}$;
3: Divided tags are after clustering by Algorithm 1, $f$ clustering centers are obtained;
4: The RSSI value $[Q_1, Q_2, \cdots, Q_{\text{tag}}]$ of the tag to be measured is divided into $f$ classes based on similarity;
5: Improved PSO algorithm for location finding;
6: for each particle $h$ do
7: Initialize velocity $V_h$ and position $X_h$ for particle $h$;
8: Evaluate particle $h$ and set $p_{\text{best}} = X_h$;
9: end for
10: $g_{\text{best}} = \min \{ p_{\text{best}} \}$;
11: while not stop do
12: for $h$ to $O$ do
13: Update the velocity and position of particle $h$ using the improved $c_1$, $c_2$:
14: Evaluate the particle $h$ according to the fitness function $F_{h} = \sqrt{\sum_{j}^{O} (Q_j - L_j)^2}$
15: end for
16: if $F_{h}(X_h) < F_{h}(p_{\text{best}})$ then
17: $p_{\text{best}} = X_h$;
18: if $F_{h}(p_{\text{best}}) < F_{h}(g_{\text{best}})$ then
19: $g_{\text{best}} = p_{\text{best}}$;
20: end if
21: end if
22: end while
23: Output $p_{\text{best}}$ and get the coordinates corresponding to the RSSI value of $p_{\text{best}}$, which is then the position of the tag to be measured.

Algorithm 2: RFID indoor positioning based on AP-PSO algorithm.

$$A(i, k) = \min \left[ 0, R(k, k) + \sum_{i \neq j, k} \max \left( 0, R(i', k) \right) \right].$$

(5)

Equation (5) represents the cumulative evidence of point $i$ selecting point $k$ as the clustering center. This is called the attribution degree, which is the information passed from point $k$ to point $i$ and is equal to the self-availability degree $(R(k, k))$, plus the positive attraction degree obtained from other points $(i')$ $(R(i', k) > 0)$. Only positive attractiveness can support point $k$ in being a cluster center. If $R(k, k)$ is negative, then point $k$ is more suitable to belonging to other cluster centers and is not suitable to be a cluster center.

(3) Self-availability: $A(k, k)$

$$A(k, k) = \sum_{i \neq k} \max \left( 0, R(i', k) \right).$$

(6)

The self-availability degree $A(k, k)$ is equal to the sum of the positive attraction degrees obtained from the other points $(i')$. This mainly reflects the ability of $k$ to be the clustering center.

(3) Iterative Updates. During the implementation of the algorithm, a damping factor $\alpha$ is used to attenuate the attractiveness and attribution information, in order to prevent

$$R(i, k) = s(i, k) - \max_{k', t, k' \neq k} \left\{ A(i, k') + s(i, k') \right\}. \quad (4)$$

Equation (4) represents the cumulative evidence of the suitability of point $k$ as a cluster center for point $i$, called Responsibility. This is the information passed from point $i$ to point $k$ and is equal to the degree of suitability of point $k$ to be a cluster center for point $i$ $(s(i, k))$, minus the degree of suitability of the largest competing point $k'$ as a cluster center for point $i$ $(A(i, k') + s(i, k'))$.

(2) Availability: $A(i, k)$

**Figure 4:** Experimental area.
numerical oscillations during the update process.

\[ R_{t+1}(i,k) \leftarrow (1 - \alpha)R_{t+1}(i,k) + \alpha R_i(i,k), \]  
\[ A_{t+1}(i,k) \leftarrow (1 - \alpha)A_{t+1}(i,k) + \alpha A_i(i,k). \]  

The AP clustering algorithm takes the data point \( k \) with \( R(k,k) + A(k,k) > 0 \) as the cluster center after each iteration. The algorithm ends, when the number of iterations reaches a set threshold (e.g., maxits is set to 500) or when its cluster centers remain unchanged after several iterations (e.g., convints is 50). After determining all the cluster centers, assign the remaining data points to the corresponding cluster centers. Where \( \alpha \) is the damping coefficient, the value is \([0.5, 1]\); generally 0.9, \( t \) is the number of iterations, and the maximum value is set to 500 [38, 39].

Section 3.1.2 shows the pseudocode for clustering RSSI values using the AP clustering algorithm.

3.1.2. AP Clustering Algorithm Pseudocode. In the RFID location search process, in order to improve search efficiency, the area to be located is usually divided into multiple subareas. Existing methods used for subarea division mainly include grid division, honeycomb division, etc. [40]. These methods have difficulty with accurately reflecting the distribution of RSSI signals. After AP clustering, the RSSI values in the same category are highly similar, and the RSSI similarity between categories is low. The reference nodes with similar RSSI characteristics and obvious changes in RSSI signal become more concentrated. After using AP clustering, the reference tags are divided according to the similarity of signal strength, in order to form a tag set to prepare for the positioning of the subsequent particle swarm.

3.2. Particle Swarm Algorithm

3.2.1. Introduction to Particle Swarm Algorithm. Particle swarm optimization (PSO) is a swarm intelligent iterative optimization method, originally proposed by James Kennedy and Russell Eberhart [41, 42]. The basic concept of PSO stems from the fact that swarms of birds and fish adjust their travel and flight speed and direction, according to external information occurring at any time during the foraging process. They adjust their search path based on their own experiences and exchanges between the population, until they find the place with the most food. In mathematical problems, this can be understood as follows: a group of feasible solutions and their corresponding initial velocities are randomly initialized within the solution space of a certain problem. Each feasible solution is regarded as a particle, and the quality of the particle is judged by the fitness function (objective function). Each particle dynamically changes its position in the solution space according to its corresponding speed, until the particle finds the optimal position of the fitness value (calculated value of the objective function) in the solution space. Each iteration of the algorithm calculates two different forms of fitness. One is the historical best fitness of a single particle, that is, the value of the objective function corresponding to the best position of the individual. The other is the historical best fitness of the particle swarm, that is, the global best fitness of the value of the objective function corresponding to the good position. The velocity and position of the particles in the solution space are continuously updated, according to the above two optimal function values. Finally, the global optimal solution is found [43, 44].

Suppose there are \( O \) particles in a \( D \)-dimensional space. The position and velocity of the \( h \)-th particle at time \( t \) can be expressed as

\[ X^t_h = (X^t_{h1}, X^t_{h2}, \ldots, X^t_{hD}), \]
The iterative update formula of particle position and velocity is

$$v_{td}^t = w v_{td}^{t-1} + c_1 r_1 (p_{best}^t - x_{td}^{t-1}) + c_2 r_2 (g_{best}^t - x_{td}^{t-1}),$$

and $$c_1$$ and $$c_2$$ represent learning factors, and $$r_1$$ and $$r_2$$ represent random numbers in the interval [0,1]. Generally, particle position and velocity have upper and lower limits in D-dimensional space, namely, $$x_{td}^t \in [-x_{max,d}, x_{max,d}]$$, $$v_{td}^t \in [-v_{max,d}, v_{max,d}]$$.

The particle swarm algorithm has been widely used in many fields, specifically because of its fast convergence, few parameters, and simple implementation. Traditional particle swarm optimization algorithms have shortcomings, such as insufficient stability, low computational efficiency, and slow convergence speed. Aimed at the problem of RFID indoor positioning, this paper proposes an improved particle swarm algorithm that integrates the AP clustering algorithm to achieve indoor RFID positioning and improve convergence speed and positioning accuracy.

4. AP Clustering Combined with an Improved Particle Swarm Algorithm for RFID Positioning

The previous section introduced the basic principles of AP clustering and the particle swarm algorithm. This section mainly introduces how to use AP clustering and the improved particle swarm algorithm to achieve RFID indoor positioning.

4.1. Detailed Principle of Positioning Algorithm

4.1.1. Dividing Tag Areas. Perform AP clustering of the collected signal strength values of the reference tags according to

- Readers and tags
- Antenna

Figure 6: Experimental materials.

Figure 7: Collect RSSI at 60 cm intervals.
their similarity. This is done to obtain the cluster centers and the number contained in each class; that is, divide the reference tags into regions according to the similarity of the signal strengths to obtain the tag set. The signal intensity value of the tag to be measured is compared with the signal intensity value of the cluster center, and the tags to be measured are classified into different classes, according to the degree of similarity.

4.1.2. Positioning Search. After different tag sets are obtained by the AP clustering algorithm, the tags to be measured are used as the basic particle of the particle swarm optimization algorithm. The position of the tags to be measured is the particle position. Particle swarm optimization is used to solve the objective function, in order to obtain the estimated value of the coordinate of the tags to be measured, so as to achieve RFID indoor positioning.

In the RFID indoor localization process, during the early stage of the particle search, it is necessary to ensure the particle diversity and improve the search capability of the optimal region. In the late stage of the search, the particles should be kept stable and converge as soon as possible. However, the traditional particle swarm optimization algorithm has insufficient stability and low computational efficiency. The traditional particle swarm algorithm uses fixed learning factors; the $c_1$ and $c_2$ values remain unchanged, which cannot adequately solve the contradiction between the diversity and stability of the particles in the early and late stages of optimization, which in turn affects the optimization performance of the algorithm. In fact, the $c_1$ and $c_2$ values affect the performance of the particle swarm algorithm search. The larger the values of $c_1$ and $c_2$ are, the stronger the local search capability will be. The smaller the values of $c_1$ and $c_2$ are, the stronger the global search capability will be. In order to improve the optimization performance of the particle swarm algorithm, a larger $c_1$ and $c_2$ should be set in the initial stage of the search, so as to effectively expand the particle’s optimization area and enhance the ability to search for optimal solutions. In the later stage of the search, a smaller $c_1$ and $c_2$ should be set, to show that the particles converge to the optimal solution accurately and quickly. The function value change characteristics of the cosine function in the interval $[0, \pi]$ are similar to the change characteristics of the particle swarm algorithm learning factor. In the RFID indoor positioning process, the initial decrease of the interval cosine function is relatively slow, which can effectively promote the search speed of the tags to be measured. To improve particle diversity, as the iteration progresses, the interval cosine function decreases faster, which can effectively improve the local search performance of the tags to be measured. The improved learning factor can be expressed as follows:

$$c_1' = c_2' = \frac{c_{\text{max}} - c_{\text{min}}}{2} \cos^2 \left( \frac{\pi t}{T_{\text{max}}} + \frac{\pi}{2} \right) + c_{\text{max}} - c_{\text{min}} = \frac{2}{C_0/C_1}.$$  \hspace{1cm} (13)

In formula (13), $c_{\text{max}}$ and $c_{\text{min}}$, respectively, represent the maximum and minimum values of the learning factor, and $T_{\text{max}}$ represents the maximum number of iteration set in advance. By incorporating formula (13) into formula (11), the speed and position update formulas of the improved particle swarm algorithm can be obtained, as shown in formulas (14) and (15). According to the new position and velocity update formula, the positioning search of the particle swarm algorithm is realized.

$$v_{hd}' = wv_{hd}' + c_1' r_1(p_{\text{best}} - x_{hd}') + c_2' r_2(g_{\text{best}} - x_{hd}'), \hspace{1cm} (14)$$
4.1.3. Fitness Function. In the entire particle iteration process, it is necessary to construct a fitness function that can characterize the relationship between the particle and the tags to be measured as an evaluation index of particle quality. The Euclidean distance between the particles generated by the adaptive factor particle swarm and the tags to be measured is calculated according to the received signal strength as the fitness function, which specifically is

\[ F_{jt} = \sqrt{\sum_{j=1}^{Q} (Q_j - L_j)^2}, \]  

where \( Q_j \) and \( L_j \) are the quality and the received signal strength of the \( j \)-th tag, respectively.
where $L_j$ is the particle signal intensity value generated by the particle swarm and $Q_q$ is the signal intensity value of the tags to be measured. The above formula shows that the smaller the distance between the particle and the tags to be measured is, the smaller is the fitness function value and conversely with larger fitness function values. According to the fitness function value, the performance of the particle position can be evaluated. During the iteration process, the particle updates the particle velocity and position, according to Equation (14) and Equation (15), and continuously searches for the optimal particle until the fitness function meets the preset threshold or the algorithm reaches the maximum number of iterations to terminate the optimization algorithm. In addition, the coordinates that corresponding to the signal strength value of the optimal particle output show the location of the tag to be measured, thus realizing the RFID indoor positioning.

4.2 Description of Key Parameters. Table 1 shows the key parameter settings and descriptions of the proposed AP-PSO positioning method.

4.3 AP-PSO Algorithm Locating Pseudocode. The following Algorithm 2 is the core pseudocode of the AP-PSO algorithm, which is proposed in this paper to implement RFID localization. The improvement of the learning factor of the particle swarm algorithm is illustrated in the pseudocode, which uses dynamic learning factors to update the particle
velocity and position to achieve fast and accurate localization of RFID.

5. Experimental Results and Analysis

5.1. Experimental Deployment

Experimental equipment is as follows: one reader, four antennas (same antenna type), 73 passive tags, and several plastic clips for holding the tags. The reader is IMPINJR420, which can support four antennas. The software for signal intensity value collection is tagee. Experimental area: this paper is an experiment conducted in a teaching and research room. Because there are tables and chairs placed in the teaching and research room, an area of 5 m long and 3.6 m wide is divided in the teaching and research room for positioning experiments in order to avoid the experimental results being affected. The experimental area is shown in Figure 4. Four antennas are connected to the reader through feed lines, and the antennas are placed on the four corners of the experimental area with the antennas at the same height of 1 m from the ground; the reader is connected to the computer through the network port. The tags are clamped and placed in the experimental area at different distance intervals, while 10 tags to be measured are randomly placed in the experimental area with gray clamps, and the positions of the tags to be measured are recorded.
Figure 4 shows the experimental area. Figure 5 is a simulated plan view of the experimental area with red pentagrams indicating the antenna, green squares indicating the reader, blue circles indicating the tags to be measured, and crosses indicating the reference tags. Figure 6 shows the relevant equipment used in the experiment. Figure 7 shows the experimental scene with the tags placed at 60 cm intervals.

5.2. Data Collection and Preprocessing. In this paper, tags were deployed at intervals of 60 cm, 90 cm, and 120 cm in the experimental area, and the RSSI values of the three distance interval tags were collected continuously about 2000 times with an interval time of 5 seconds, respectively, while keeping the reader antenna position unchanged. The acquired dataset of tag signal intensity values is processed using Gaussian filtering [45, 46], and the filtered data are
used for the localization experiments in this paper. For the RSSI values of the same tag received by the same reader in consecutive time, there are some invalid data in the acquired data due to the interference of multipath effect and noise. In order to avoid the interference of noisy data on the localization results, a Gaussian filter is used to preprocess the RSSI values. Gaussian filter is a kind of linear filter, and the RSSI values with high probability of occurrence interval are selected as valid values by Gaussian model, and then, the arithmetic mean is calculated as the filter data for localization, so as to reduce the influence of noisy data on localization.

The received signal intensity value obeys the Gaussian distribution of \( (0, \sigma^2) \) distribution, and its probability density function is shown in

\[
F_{(RSSI)} = \frac{1}{\sigma \sqrt{2\pi}} e^{(RSSI-\mu)^2/2\sigma^2},
\]

in, \( \mu = \frac{1}{N} \sum_{z=1}^{N} RSSI_z \), and \( \sigma = \sqrt{(1/(N-1)) \sum_{z=1}^{N} (RSSI_z - \mu)^2} \), RSSI means the zth signal intensity value, and \( N \) is the number of measurements.

Then, the probability formula of interval \( (\mu - \sigma RSSI, \mu + \sigma) \) is as follows:

\[
P(\mu - \sigma RSSI, \mu + \sigma) = F(\mu + \sigma) - F(\mu - \sigma) = \phi(1) - \phi(-1) = 0.6826.
\]  

Calculate the arithmetic mean of these RSSI values:

\[
\text{RSSI} = \frac{1}{N} \sum_{z=1}^{N} RSSI_z \epsilon (\mu - \sigma, \mu + \sigma).
\]

The RSSI value of the tag numbered 18 in the 60 cm interval arrangement tag was selected for Gaussian filtering experiments. Figure 8 is the comparison of signal intensity values before and after Gaussian filtering. The blue line is the signal intensity value before filtering, and the red line is the signal intensity value after filtering. It can be seen from the figure that the filtered image is smoother, indicating that Gaussian filtering can reduce noise influence and improve the accuracy of experimental results. The signal intensity value after Gaussian filtering is used in the next step of AP clustering and positioning.

5.3. Experimental Results

5.3.1. Analysis of RSSI Clustering Results. Figures 9–11 show the AP clustering results for RSSI values. Figure 9 shows the results of clustering with tags arranged at 60 cm intervals. Figure 10 shows the clustering results for tags arranged at 90 cm intervals. Figure 11 shows the clustering results of the tags arranged at 120 cm intervals. Figure 11(a) is a three-dimensional plot of the AP clustering results; the x-axis is the label horizontal coordinate, the y-axis is the label vertical coordinate, and the z-axis is the RSSI value of the tags. Figure 11(b) is a plan view of the AP clustering results.

<table>
<thead>
<tr>
<th>Pairing 1:</th>
<th>Average value (E)</th>
<th>Number</th>
<th>Standard deviation</th>
<th>Standard error mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP-PSO</td>
<td>16.100</td>
<td>10</td>
<td>4.581</td>
<td>1.449</td>
</tr>
<tr>
<td>PSO</td>
<td>49.300</td>
<td>10</td>
<td>5.697</td>
<td>1.802</td>
</tr>
<tr>
<td>Pairing 2:</td>
<td>Average value (E)</td>
<td>Number</td>
<td>Standard deviation</td>
<td>Standard error mean</td>
</tr>
<tr>
<td>AP-PSO</td>
<td>16.100</td>
<td>10</td>
<td>4.581</td>
<td>1.449</td>
</tr>
<tr>
<td>K-means</td>
<td>68.500</td>
<td>10</td>
<td>8.396</td>
<td>2.655</td>
</tr>
</tbody>
</table>

Figure 14: Comparison of AP-PSO with other positioning methods.
5.3.2. Positioning Error Analysis. Define a variable $E$ to calculate the size of the positioning error. As shown in formula (20), $E$ is equal to the sum of the squared difference between the output position of the AP-PSO algorithm and the actual position of the tags.

$$ E = \sqrt{(x_{\text{output}} - x_{\text{actual}})^2 + (y_{\text{output}} - y_{\text{actual}})^2}. \quad (20) $$

Figure 12 shows the localization results of the improved particle swarm algorithm for three different tag intervals. As can be seen from the figure, the positioning error is the smallest when the tag interval is 60 cm, and the error is concentrated in about 20 cm. The positioning error is the largest when the tag interval is 120 cm, and the error is concentrated in about 70 cm. The positioning error is the middle when the tag interval is 90 cm, and the error is concentrated in about 50 cm. After comparing the positioning errors of three kinds of tag intervals, it can be seen that the positioning effect is best when the tag interval is 60 cm, so this paper uses the positioning results of the tag interval of 60 cm and other methods for comparative analysis.

Figure 13 shows the comparison of the localization error of the $K$-means clustering localization algorithm for different number of clusters. The blue line is the localization error when the number of clusters is 3; the red line is the localization error when the number of clusters is 4; the black line is the localization error when the number of clusters is 5; and the green line is the localization error when the number of clusters is 6. As can be seen from the figure, overall, the localization error is the largest when the number of clusters is 3 classes; the smallest when the number of clusters is 5 classes; and the localization error is between 3 and 5 classes when the number of clusters is 4 and 6 classes. The localization results of the $K$-means clustering algorithm when the number of clusters is 5 classes are chosen to compare with the localization results of the AP-PSO algorithm; i.e., the best localization results of the $K$-means clustering algorithm are chosen to compare with the proposed method in this paper.

Figure 14 shows the results of the localization error of the proposed method (AP-PSO), traditional particle swarm algorithm, and $K$-means algorithm in this paper. The RSSI data for the comparison experiments are for the tag interval of 60 cm. From the figure, it can be seen that the localization error of the AP-PSO algorithm is the smallest, and the error is concentrated between 10 cm and 25 cm; the localization error of the $K$-means algorithm is the largest, and the error is concentrated between 30 cm and 50 cm; the localization error of the traditional particle swarm algorithm is between the AP-PSO algorithm and the $K$-means algorithm, and the error is concentrated between 25 cm and 45 cm. Therefore, the proposed localization method (AP-PSO) is better than the traditional particle swarm algorithm and $K$-means algorithm in terms of localization accuracy.
To verify the effectiveness of the AP-PSO localization algorithm proposed in this paper, a t-test was chosen to compare the advantages and disadvantages of the three methods. In this paper, paired t-tests were conducted for the localization error results of AP-PSO localization algorithm, traditional particle swarm algorithm (PSO), and K-means algorithm, respectively, and two sets of t-test results were obtained, i.e., AP-PSO and PSO and AP-PSO and K-means algorithm.

**Table 5: Relative implement time of the three methods.**

<table>
<thead>
<tr>
<th>Location algorithm</th>
<th>Average computing time/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP-PSO algorithm</td>
<td>0.62</td>
</tr>
<tr>
<td>K-means clustering algorithm</td>
<td>1.03</td>
</tr>
<tr>
<td>Particle swarm algorithm</td>
<td>0.88</td>
</tr>
</tbody>
</table>

**Figure 15: Mean error comparison.**

**Figure 16: Convergence speed comparison.**
-means, and the test results are shown in Tables 2–4. Table 2 shows the statistical information of paired samples. From Table 2, it can be seen that the AP-PSO algorithm is smaller than the PSO algorithm and the K-means algorithm in terms of average localization error, standard deviation of error, and mean of standard error, which indicates that the localization results of the AP-PSO algorithm have good stability. Table 3 shows the paired sample correlation. Table 4 shows the paired sample detection results. From the significance results in Table 4, it can be seen that the positioning error of the AP-PSO positioning algorithm proposed is significantly superior to the other 2 algorithms. t-test proves the effectiveness of the AP-PSO algorithm for RFID indoor positioning, in this paper.

Figure 15 shows the comparison results of the three positioning algorithms in terms of maximum, minimum, and average values of positioning errors. From the figure, it can be seen that the maximum localization error of AP-PSO algorithm is 25 cm, the minimum error is 8 cm, and the average error is 16.1 cm; the maximum localization error of traditional particle swarm algorithm is 42 cm, the minimum error is 22 cm, and the average error is 32.9 cm; the maximum localization error of K-means clustering algorithm is 52 cm, the minimum error is 25 cm, and the average error is 36.5 cm. The AP-PSO algorithm outperforms the other two algorithms in terms of positioning accuracy, which indicates that the proposed method can improve the accuracy of RFID indoor positioning.

5.3.3. Convergence Comparison. In the traditional particle swarm algorithm, the learning factor is fixed, the computational efficiency is low, the convergence speed is slow, and it is prone to problems such as local optimization in the late stage of the search, which will affect the accuracy and efficiency of RFID localization. To address the above problems, this paper proposes the AP-PSO algorithm, which improves the learning factor of particle swarm, introduces interval cosine function, adaptively adjusts the learning factor, and improves the particle convergence speed and local search performance. Figure 16 shows the comparison between AP-PSO algorithm and traditional particle swarm algorithm in terms of convergence speed. As can be seen from the figure, the AP-PSO algorithm achieves convergence in about 35 iterations, and the traditional particle swarm algorithm achieves convergence in about 50 iterations. The convergence speed of the AP-PSO algorithm is significantly better than that of the traditional particle swarm algorithm, and the proposed improved learning factor improves the particle search speed and accelerates the convergence of the algorithm, in this paper.

5.3.4. Time Complexity Comparison. The MATLAB software was used to run the three positioning algorithms 10 times in the same hardware and software environment, and the calculation time of the algorithms was counted, and the results of the 10 times were averaged to obtain the average calculation time of the three positioning algorithms as shown in Table 5.

It can be seen from Table 5 that the average mathematical operation time of AP-PSO algorithm positioning is 0.62 s, the average mathematical operation time of the traditional particle swarm algorithm is 0.88 s, the average mathematical operation time of the K-means algorithm is 1.03 s, and the AP-PSO algorithm runs faster than the other two algorithms. The AP-PSO algorithm clusters the tags data in the first stage, which reduces the search range of the improved particle swarm algorithm and uses the learning factor to update the particle velocity and position, which improves the convergence speed of the algorithm, and the final computing time is less than the other two algorithms.

6. Conclusion

This paper studies the problem of RFID indoor positioning and proposes a positioning method based on AP clustering and an improved particle swarm algorithm (AP-PSO). This method performs AP clustering on the signal strength value RSSI value, obtains the cluster center and number, and divides the tag area. Then, the improved particle swarm algorithm is used for a location search to achieve RFID indoor positioning. The improved particle swarm algorithm introduces an interval cosine function to adjust the learning factor, in order to enhance the search performance and convergence speed of the particle algorithm. The experimental results show that the AP-PSO algorithm has certain advantages, compared with the particle swarm algorithm and the K-means clustering algorithm. The advantages relate to positioning accuracy, convergence speed, and algorithm operation efficiency. The average positioning error of the AP-PSO algorithm is 16.1 cm; the average error of the particle swarm algorithm is 32.9 cm, and the average error of K-means clustering is 36.5 cm. In terms of convergence speed, AP-PSO converges for 35 iterations, and the particle swarm algorithm converges for 50 iterations. The average computing times are 0.62 s, 0.88 s, and 1.03 s, respectively. As can be seen, the algorithm proposed in this paper improves the accuracy and efficiency of RFID positioning.

Data Availability

Experimental equipment is as follows: a reader and four antennas (the antenna model is the same), 73 passive tags, and a number of clips to hold tags; the reader uses IMPINJR420 reader, which can support four antennas, and the signal strength data is used by software tagee. Experimental area: this article uses a teaching and research room to divide a 3.6 m wide and 5 m long area as the experimental area. Because there are tables around the teaching and research room, to avoid affecting the experiment, an area is divided for related experiments. In the experiment method, the antenna is connected to the reader through a feeder, and the reader is connected to the computer through a network cable port. The four antennas are connected to a reader and placed on the four corners of the measure area. The antennas are at the same level of 1 m from the ground. On the above, the data collection interval is divided into 60 cm, 90 cm, and 120 cm. At the same time, 73 tags, 40 tags, and
30 tags are required, respectively. Each of them contains 10 tags to be measured. The tags to be measured are, respectively, clamped with gray clips. Place in the measure area, the antennas and the tags are in the same horizontal plane, and the RSSI value of the tags is continuously collected about 2000 times at an interval of 5 seconds while keeping the position of the reader antenna unchanged.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The subject is sponsored by the National Natural Science Foundation of China (No. 61872196, No. 61872194, No. 61902196, No. 62102194, and No. 62102196), the Six Talent Peaks Project in Jiangsu Province (No. RJFW-111), and the Postgraduate Research and Practice Innovation Program of Jiangsu Province (No. KYCX21_0799, No. KYCX19_0909, No. KYCX19_0911, No. KYCX20_0759, No. KYCX21_0787, and No. KYCX21_0788).

References


