applied sciences

MDPI

Article

An Anchor-Free Location Algorithm Based on Transition Coordinates

Jinzhao Fan and Sanjun Liu *

College of Intelligent Systems Science and Engineering, Hubei Minzu University, Enshi 445000, China; 18851925089@163.com

* Correspondence: liusanjunbox1@126.com

Abstract: In some location scenarios where the location information of nodes cannot be mastered in advance, the anchor-free location technology is particularly important. In order to reduce the complicated calculation and eliminate the accumulated error in the traditional anchor-free location algorithm, a new anchor-free location algorithm based on transition coordinates is proposed in this paper. This algorithm is different from the traditional methods such as minimum cost function or inverse matrix. Instead, N initial coordinates are randomly generated as the starting position of the transition coordinates, and the position increment between the transition coordinates and the real coordinates of the node is constantly modified. After K iterations, the convergent position coordinates are finally infinitely close to the real position coordinates of N nodes, and the computational complexity is less than most existing algorithms. As follows, the factors that affect the performance of the algorithm are investigated in the simulation experiment, including the topology structure, positioning accuracy and the total number of nodes, etc. The results show great advantages compared with the traditional anchor-free positioning algorithm. When the topology structure of the initial coordinates changes from a square to a random graph, the number of iterations increases by 15.79%. When the positioning accuracy increased from 1% to 1‰, the number of iterations increased by 36.84%. When the number of nodes N is reduced from 9 to 4, the number of iterations is reduced by 63.16%. In addition, the algorithm can also be extended to the field of moving coordinates or three-dimensional spatial positioning, which has broad application prospects.

Keywords: anchor-free location; transition coordinates; position increment; topology structure

check for updates

Citation: Fan, J.; Liu, S. An Anchor-Free Location Algorithm Based on Transition Coordinates. *Appl. Sci.* **2024**, *14*, 10320. https://doi.org/10.3390/app142210320

Academic Editor: Alessandro Lo Schiavo

Received: 6 October 2024
Revised: 1 November 2024
Accepted: 5 November 2024
Published: 9 November 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Many application scenarios of wireless sensor networks need to know the node location. Node location has an important reference value for information collection and data processing, so positioning is a hot topic in wireless sensor network research [1,2]. The traditional positioning methods are usually positioned with the global positioning system (GPS) or anchored points with geographic location information [3–5]. Anchored points refer to nodes [6,7] with geographic coordinates obtained in advance, and the rest nodes are called unanchored points. However, the positioning of anchor points in wireless sensor networks has many inconveniences and defects in indoor or suburban scenarios [8,9]. Anchor-free location technology [10] does not need to master the geographical location information of nodes in advance, and does not use GPS or anchor points, but calculates the location of nodes through the local distance information and connectivity between node pairs, so as to determine the relative coordinates and network topology of the entire network.

The earliest anchor-free location algorithm is anchor-free localization (AFL) proposed by Priyantha N B et al. [11], which is a completely distributed location algorithm. The location algorithm is divided into two steps: the first step is to obtain the model, obtain the collapsible network model by using the heuristic principle, and determine the rough coordinates of the unknown node by using the centroid location algorithm; the second

part corrects the error, using the particle spring model to correct and optimize the node positioning error. However, there is no clear calculation method of single jump distance in AFL algorithm. The centralized boundary discovery (CBD) algorithm proposed in [12] for unanchored point location can define the energy spent in the initialization process, but the algorithm cannot be used to calculate the energy consumed by a single node. An assumption-based coordinates (ABC) anchor-free location algorithm was proposed by Savarese, C., Rabaey, J. M. et al. [13], which is a distributed location algorithm. The coordinates of an unknown node are calculated by one estimate, and then the coordinates of the unknown node are used as the reference coordinates to calculate the coordinate position of the next adjacent unknown node, and so on, and the coordinates of all unknown nodes in the network are computed. However, its positioning error is relatively large, and it is generally used as an auxiliary positioning mechanism. Li, D. et al. proposed an improved ABC algorithm based on the dynamic Levy mechanism [14], which effectively expanded the search scope and avoided falling into local areas, but it still needs further research and testing in practical engineering applications. The above algorithms need to master the position distance information between nodes, and then establish the cost function based on it, and then find the optimal solution to get the position coordinates of nodes.

In addition, there is also a method of matrix inversion to solve the position coordinates. For example, Shang Y et al. proposed a multidimensional positioning algorithm without anchors based on the principle of multidimensional scaling (MDS), which is a centralized location algorithm [15]. The MDS algorithm makes full use of the local distance information and connectivity between unknown nodes, and can simultaneously locate multiple unknown nodes. However, the algorithm replaces the Euclidean distance with the shortest path distance between node pairs. When the shortest path distance differs greatly from the Euclidean distance, there is a large error. In addition, the accumulation and propagation errors of a centralized location algorithm will also bring some positioning errors. For irregular networks, the semi-centralized MDS method proposed in literature [16] has lower complexity and more robustness and accuracy than the centralized MDS, but it still has the problem of error accumulation. The anchor-free location protocol designed in [17] adopts the management mechanism of network clustering and combines each local coordinate map to form a set of overall coordinate system of the network. Although the protocol has good scalability, it is computationally heavy when combining each cluster plot.

The above-mentioned methods of solving position coordinates by constructing the minimum cost function or matrix inversion are very complex formulas. Therefore, when applying these algorithms for positioning, a large number of calculations need to be performed each time, and there is an error between the calculated node coordinates and the real coordinates, resulting in unstable graphics, large jumping ratios, and significant errors compared to the real position coordinates. In military exercises, this affects, for example, the positioning problem of the movement trajectory of combat personnel, frequent changes in the position coordinates of multiple personnel, and other scenarios. However, in reality, the current position of a node at any given moment should have a high correlation with its previous position, meaning that the new position should have an offset based on the original position.

Contributions to this article include the following. Firstly, an anchor-free location algorithm $\mathbf{p}_i^{k+1} = f(d_{ij}^{k+1})$ based on transition coordinates is proposed. The position coordinates $\mathbf{p}_i^{k+1}$ at time $k+1$ are related to both $\mathbf{p}_i^k$ and d_{ij}^{k+1}. In the positioning process, the transition coordinates of each node are correlated with the previous transition coordinates, which can dynamically reflect the relative motion of the transition coordinates of each node. Therefore, the algorithm proposed in this paper is very suitable for the occasions where the node coordinates change frequently or the indoor GPS signal is weak. Secondly, the proposed algorithm requires less computation in the positioning process, the positioning error can reach less than 1‰ and there is no error accumulation problem. In addition, the algorithm can also be extended to the field of moving coordinates or three-dimensional spatial positioning, which has broad application prospects.

The rest of this article includes the following content. In Section 2, we described the system model of the proposed an anchor-free localization algorithm. In Section 3, we described the specific principles and steps of the proposed algorithm. In Section 4, we compared the complexity of the proposed algorithm with existing traditional algorithms. In Section 5, we conducted extensive experiments in MATLAB R2016a to verify the accuracy and convergence of the proposed algorithm. Finally, in Section 6, we summarized the conclusions of this article and discussed future work.

2. System Model

In this paper, the distance $d'_{ij}, 1 \leq i, j \leq N, i \neq j$ between N nodes is collected in a self-organizing way. According to the arrangement and combination formula, there are a total of $N_d = N(N-1)/2$ distances, and then the distance information is transmitted to the data processing center of the system, and the processing center attempts to use the N_d distances to calculate the relative position coordinates between N nodes. In the early stage of the algorithm in this paper, only the anchor-free location problem of these nodes in the same plane is studied, and the position of N nodes is not known. According to geometric knowledge [18], if the rotation and translation of nodes are not considered, the relative position coordinates of N nodes will be unique. As shown in Figure 1, we assume that the number of nodes N = 9, so the number of distances N_d between nodes is 36, that is, the distance d'_{ij} between two nodes is represented by 36 dotted lines in the figure. For the convenience of drawing, A_1 is drawn at the origin of coordinates for all graphs that do not consider rotation and translation,A_2 is located in the positive half axis of the x-axis, and A_3 is located in the first and second quadrants. The following section gives the specific algorithm, and the meanings of each symbol are shown in Table 1.

Table 1. Notations description.

Notations	Definition
N	Number of nodes
$\mathbf{p}_i$	The actual position coordinates of node i
$\mathbf{p}_i^k$	Transition coordinates of node i at time k
$\mathbf{p}_i^{k+1}$	Transition coordinates of node i at time $k+1$
$\Delta\mathbf{p}_i^{k+1}$	The position increment of the transition coordinate of node i from time k to time $k+1$
d'_{ij}	The distance between the actual position coordinates of N nodes
d_{ij}^k	Distance between the hypothetical position of the node at time k
d_{ij}^{k+1}	Distance between the hypothetical position of the node at time $k+1$
K	Number of algorithm iterations

N nodes are set to be represented by numbered A_i and the real coordinate $\mathbf{p}_i, 1 \leq i \leq N$ of each node is undetermined, which is a problem that needs to be solved by anchor-free location technology. Traditional algorithms solve $\mathbf{p}_i$ by trigonometry or constructing a cost function and minimizing it, or inverse matrix [8–10,19], which all have the problem of many iterations and large amount of computation. In order to solve the above problems, reduce the number of iterations, and simplify the amount of computation, the anchor-free location algorithm based on transition coordinates $\mathbf{p}_i^k, 1 \leq i \leq N, 0 \leq k \leq K$ is proposed in this paper, where K represents the number of transition coordinates of nodes and is also equal to the number of iterations. When N is less than 10, K is generally less than 30, which is far lower than the number of iterations—more than several hundred times in the traditional algorithm.

The system first uses a random function to generate the initial transition coordinate $\mathbf{p}_i^0, 1 \leq i \leq N$ of N nodes, then iteratively updates the new position vector $\mathbf{p}_i^{k+1}$, $1 \leq i \leq N, 0 \leq k \leq K-1$ by adding position increment $\Delta\mathbf{p}_i^{k+1}$ to the previous transition coordinate $\mathbf{p}_i^k, 1 \leq i \leq N, 0 \leq k \leq K-1$. After K iterations, the final convergent transition

coordinate $\mathbf{p}_i^K, 1 \leq i \leq N$ will be infinitely close to the real coordinate $\mathbf{p}_i, 1 \leq i \leq N$. The iterative algorithm based on transition coordinates proposed in this paper does not construct cost function, nor does it use matrix inversion and other complex operations. The core innovation is that the part of position increment $\Delta\mathbf{p}_i^{k+1}$ only uses the difference between the transition distance d_{ij}^{k+1} and the real distance d'_{ij}. By compensating the difference, the transition coordinate $\mathbf{p}_i^k, 1 \leq i \leq N, 0 \leq k \leq K-1$ approximates the real coordinate. The approach speed is faster than the traditional scheme, and the calculation amount is greatly reduced.

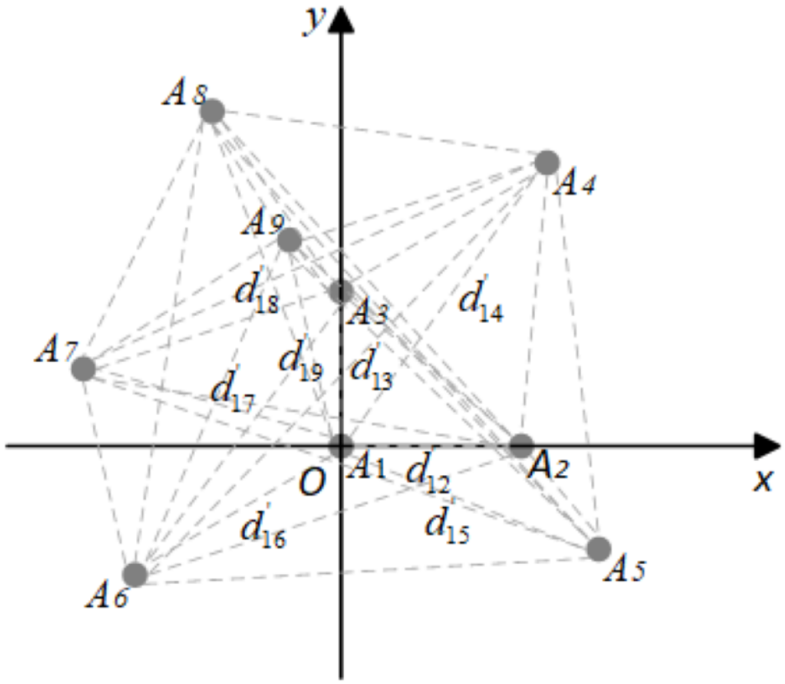


Figure 1. Schematic diagram of anchor-free positioning.

3. Algorithm Description

This paper proposes an algorithm based on transition coordinates $\mathbf{p}_i^k, 1 \leq i \leq N$, $0 \leq k \leq K$ to achieve an anchor-free location. As can be seen from the previous section, we set the changes in the transition coordinates $\mathbf{p}_i^k, 1 \leq i \leq N, 0 \leq k \leq K-1$ of each member on its horizontal and vertical axes Δx_i^{k+1} and Δy_i^{k+1} at time $k+1$, then they are written in the form of vectors $\Delta\mathbf{p}_i^{k+1} = \left(\Delta x_i^{k+1}, \Delta y_i^{k+1}\right)$, thus obtaining the transition coordinates of N members at time $k+1$:

$$\begin{aligned} \mathbf{p}_i^{k+1} &= \mathbf{p}_i^k + \Delta\mathbf{p}_i^{k+1} \\ &= \left(x_i^k, y_i^k\right) + \left(\Delta x_i^{k+1}, \Delta y_i^{k+1}\right) \\ &= \left(x_i^k + \Delta x_i^{k+1}, y_i^k + \Delta y_i^{k+1}\right) \end{aligned} \tag{1}$$

In formula (1), the left side of parentheses represents the horizontal coordinate x_i of node i at time k, adding the increment along the x-axis, and the right side of parentheses

represents the vertical coordinate y_i of node y at time k adding the increment along the y-axis. The formula for Δx_i^{k+1} and Δy_i^{k+1} is:

$$\begin{cases} \Delta x_i^{k+1} = \frac{1}{(N-1)} \sum\limits_{\substack{j=1\text{to}N \\ \text{and}j\neq i}} \left(d'_{ij} - d_{ij}^k\right) cos\alpha_{ij} \\ \Delta y_i^{k+1} = \frac{1}{(N-1)} \sum\limits_{\substack{j=1\text{to}N \\ \text{and}j\neq i}} \left(d'_{ij} - d_{ij}^k\right) sin\alpha_{ij} \end{cases} \tag{2}$$

In formula (2), α_{ij} is the angle formed between vector $\mathbf{p}_i^k, 1 \leq i \leq N, 0 \leq k \leq K$ and the positive direction of the x-axis, which ranges from 0 degrees to 360 degrees. So, formula (2) can be converted to:

$$\begin{cases} \Delta x_i^{k+1} = \frac{1}{(N-1)} \sum\limits_{\substack{j=1\text{to}N \\ \text{and}j\neq i}} \left(d'_{ij} - d_{ij}^k\right) \frac{x_i^k - x_j^k}{\sqrt{\left(x_i^k - x_j^k\right)^2 + \left(y_i^k - y_j^k\right)^2}} \\ \Delta y_i^{k+1} = \frac{1}{(N-1)} \sum\limits_{\substack{j=1\text{to}N \\ \text{and}j\neq i}} \left(d'_{ij} - d_{ij}^k\right) \frac{y_i^k - y_j^k}{\sqrt{\left(x_i^k - x_j^k\right)^2 + \left(y_i^k - y_j^k\right)^2}} \end{cases} \tag{3}$$

In Equation (3), the $d'_{ij} - d_{ij}^k$ formula is $d'_{ij} - d_{ij}^k = d'_{ij} - \sqrt{\left(x_i^k - x_j^k\right)^2 + \left(y_i^k - y_j^k\right)^2}$. The coefficient of the formula is $1/(N-1)$, which is because in addition to the member i, there are $N-1$ members, and the increment of d'_{ij} of each member relative to d_{ij}^k can cause the change in the x-axis and y-axis coordinate increment of member i. Let us assume that they have the identical effect, namely, their weighted average has the equivalent weight coefficient, so they are all equal to $1/N-1$.

From the above formula, we can calculate the distance between the transition coordinates $\mathbf{p}_i^k, 1 \leq i \leq N, 0 \leq k \leq K-1$ of N members at time $k+1$ and the expression $d_{ij}^{k+1}, i, j = 1, 2, \cdots, N, j \neq i$ is as follows:

$$d_{ij}^{k+1} = \sqrt{\left(x_i^k - x_j^k + \Delta x_i^{k+1}\right)^2 + \left(y_i^k - y_j^k + \Delta y_i^{k+1}\right)^2} \tag{4}$$

In formula (4), $\mathbf{p}_i^k = \left(x_i^k, y_i^k\right), \mathbf{p}_j^k = \left(x_j^k, y_j^k\right)$ is the position coordinate of node i, j at time k, and $\Delta x_i^{k+1}, \Delta y_i^{k+1}$ is the position increment of node j at time $k+1$ relative to node i, which is regarded as an infinitesimal quantity of higher order and obtained by using the total differential formula of multiple functions:

$$\begin{aligned} d_{ij}^{k+1} &= d_{ij}^k + \frac{\mathbf{p}_i^k - \mathbf{p}_j^k}{d_{ij}^k} \bullet \Delta \mathbf{p}_i^{k+1} \\ &= d_{ij}^k + \mathbf{e}_{ij} \bullet \Delta \mathbf{p}_i^{k+1} \end{aligned} \tag{5}$$

In formula (5), the circle $\bullet$ represents the dot product of the vector, that is, the multiplication and addition of the same components of the two vectors. $\mathbf{e}_{ij} = (\mathbf{p}_i^k - \mathbf{p}_j^k)/d_{ij}^k$ represents the unit vector from the position vector $\mathbf{p}_j^k$ to $\mathbf{p}_i^k$, whose modulus is equal to 1. $\mathbf{e}_{ij} \bullet \Delta \mathbf{p}_i^{k+1}$ represents the increment of the vector $\Delta \mathbf{p}_i^{k+1}$ in the $\mathbf{e}_{ij}$ direction by the position of the member $\mathbf{p}_i^k$, the magnitude of which is equal to the projection of the position increment vector in the $\mathbf{e}_{ij}$ direction.

We know that the d_{ij}^{k+1} between the transition coordinates $\mathbf{p}_i^{k+1}$ of N members at time $k+1$ becomes larger or smaller, which is completely caused by the member $\mathbf{p}_i$ or $\mathbf{p}_j$ moving a certain distance along the direction of $\mathbf{e}_{ij}$ at $k+1$, that is, the member $\mathbf{p}_i$ or $\mathbf{p}_j$ does

not move along the direction perpendicular to $\mathbf{e}_{ij}$. It can be expressed by mathematical formula as:

$$\Delta \mathbf{p}_i^{k+1} = \left(d'_{ij} - d_{ij}^k\right)\mathbf{e}_{ij} \tag{6}$$

When you decompose $\mathbf{e}_{ij}$ along the x-axis and y-axis, there is:

$$\mathbf{e}_{ij} = cos\alpha_{ji} \bullet \mathrm{e}_x + sin\alpha_{ji} \bullet \mathrm{e}_y \tag{7}$$

$\mathbf{e}_x$ and $\mathbf{e}_y$ represent unit vectors in the x-axis and y-axis directions, and their modulus is equal to 1. Substituting Equation (7) into Equation (6) gives:

$$\Delta \mathbf{p}_i^{k+1} = \left(d'_{ij} - d_{ij}^k\right)cos\alpha_{ji} \bullet \mathrm{e}_x + \left(d'_{ij} - d_{ij}^k\right)sin\alpha_{ji} \bullet \mathrm{e}_y \tag{8}$$

The algorithm proposed in this article can ensure reliable convergence in the vast majority of scenarios, except for some extreme scenarios such as nodes on a straight line or overlapping node coordinates where convergence may not be possible. However, the possibility of the above extreme situations occurring is relatively small, so it can be said that the algorithm proposed in this article has good convergence.

As is shown in Algorithm 1, an anchor-free location algorithm based on transition coordinates is given to realize the specific steps of positioning.

Algorithm 1 An anchor-free location algorithm based on transition coordinates

1: Set the real position coordinates of the moving member $A_i, 1 \leq i \leq N$.
2: Set the assumed position coordinates the moving member $P_i^0, 1 \leq i \leq N$ and the starting position of the transition coordinates.
3: Calculate the real distance d'_{ij} between each moving member through the distance formula between two points.
4: Calculate the distance between the transition coordinates of each moving member in the same way, and use formula (3) to calculate the offset of the transition coordinates of each moving member on its horizontal and vertical axes.
5: Formula (1) is used to obtain the new position coordinates of the transition coordinates of each moving member, and formula (9) is used to find the error proportion of the distance between the transition coordinates of each moving member and the actual coordinates of each moving member.
6: The new position coordinate of the obtained transition coordinate of the moving member is used as the starting position of the next iteration update, which is returned to step 2, and the iteration is stopped until the set distance error ratio is met.

4. Computational Complexity

Computation complexity is an important factor to consider when choosing a technique for the unanchored location. The higher the accuracy of the positioning required, the more times the algorithm has to be calculated and the longer the algorithm takes to calculate. Similarly, the more nodes are located, the more times the algorithm is computed and the longer the algorithm takes to compute. The computational complexity of the existing centralized algorithm is generally $O(N^3)$, the computational complexity of the semi-centralized algorithm is generally $O(Nk^3)$, and the computational complexity of the distributed algorithm is generally $O(NL)$, while the complexity of the algorithm proposed in this paper is $O((N-1)L)$. The symbols N, k, and L in the above computational complexity represent the number of nodes, the number of neighbors, and the number of iterations, respectively, where the number of nodes is greater than the number of neighbors [20]. Table 2 summarizes the computational complexity of the existing algorithms and the algorithms proposed in this paper. Compared with the existing localization algorithms, the proposed algorithm has less computational complexity and can quickly locate the location coordinates of nodes.

Table 2. The computational complexity of different localization algorithms.

Approaches	Computation Complexity
Centralized	$O(N^3)$
Semi-centralized	$O(Nk^3)$
Distributed	$O(Nk^3)$
The algorithm proposed in this paper	$O((N-1)L)$

5. Simulation Result

In order to verify the performance of the algorithm of the unanchored location, MATLAB R2016a software is used to simulate the algorithm. The positioning area is set to a two-dimensional plane of 100 m × 100 m, the number of nodes is set to N, and the real position of each member is randomly generated by the system in $[-50, 50]$, one of which is its horizontal coordinate and the other is its vertical coordinate. In order to better observe the relative position graph formed by each member, we adjusted the position coordinate of the first member to the origin, the position of the second member to the positive half axis of the x-axis, and the position of the third member to the positive half axis of the y-axis.

After the completion of each iteration update, the ratio of the sum of distance errors between the transition coordinates of all nodes and the real coordinates to the sum of distances between the real coordinates of each node is called error proportion (EP). The value of the EP is that the positioning accuracy needs to be achieved in the positioning process. We give the EP formula as follows:

$$\mathrm{EP} = \frac{\sum\limits_{\substack{i,j=1\mathrm{to}N \\ \mathrm{and}i\neq j}} |d_{ij} - d'_{ij}|}{\sum\limits_{\substack{i,j=1\mathrm{to}N \\ \mathrm{and}i\neq j}} d'_{ij}} \tag{9}$$

where N is the total number of nodes, $|d_{ij} - d'_{ij}|$ represents the absolute value of the distance error between the transition coordinates of the node and the real coordinates, and d'_{ij} represents the distance between the real coordinates of the nodes.

The authors of [11] proposed the global energy ratio (GER) index, which considers the distance between nodes and neighbors and non-neighbors from the perspective of the whole network, reflecting not only the side length error but also the overall network structure. Let the real coordinate distance between node N_i and N_j be r_{ij}, and the distance between the coordinates after their positioning is completed be $\hat{d}_{ij}$, and set $\hat{e}_{ij} = (\hat{d}_{ij} - r_{ij})/r_{ij}$. If the network size is N, GER is defined as follows:

$$\mathrm{GER} = \frac{\sqrt{\sum\limits_{i,j,i<j} \hat{e}_{ij}^2}}{N(N-1)/2} \tag{10}$$

In this paper, the factors that affect the performance of the algorithm mainly include the topological structure of the initial coordinates, the positioning accuracy, and the number of nodes N. After reading a lot of the literature, the simulation conditions set for our first experiment were that the topology structure of the initial coordinates was square, the positioning accuracy was 1%, and the number of nodes N = 9. This experiment was taken as a control experiment for the rest of the simulation experiments in this paper, and the simulation effect was shown in Figure 2.

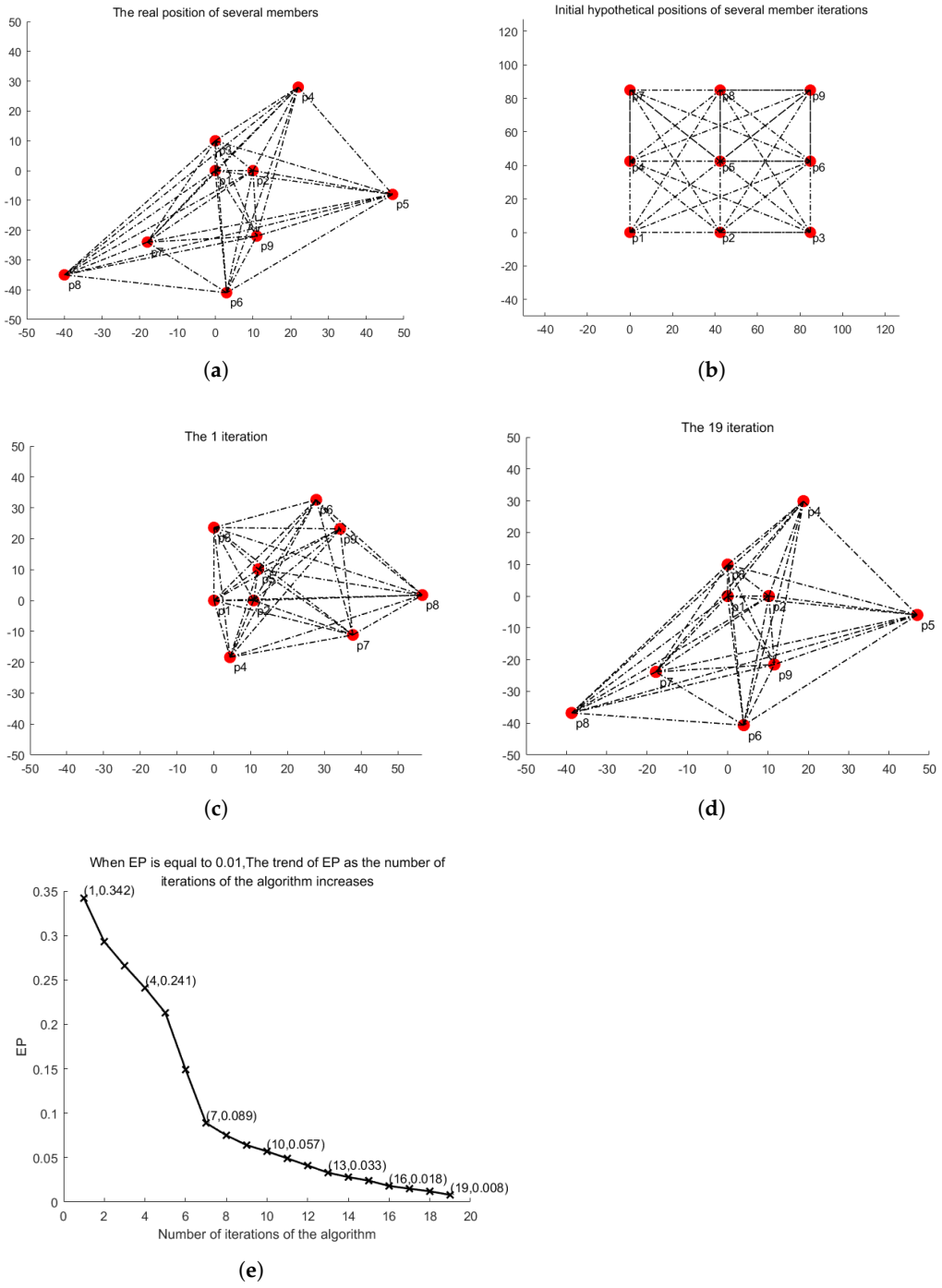


Figure 2. The effect of the first experiment of the algorithm: (**a**) the real position several members; (**b**) initial hypothetical positions of several member iterations; (**c**) the first iteration; (**d**) the 19th iteration; (**e**) when EP is equal to 0.01, the trend of EP as the number of iterations of the algorithm increases.

As can be seen from Figure 2, the initial position coordinates of 9 nodes are generated by a random function and serve as the starting position of transition coordinates, and the initial error EP is 0.342. Then, by constantly correcting the position increment between transition coordinates and real coordinates through the algorithm proposed in this paper, it can be seen that EP is less than 0.1 when it is equal to 0.089 after 7 iterations. After the 19th iteration, the positioning error EP is equal to 0.008 and less than 0.01 to meet the error conditions set by the simulation, and finally, the relative position graph formed by

the transition coordinates of these 9 nodes is obtained. After comparing with the relative position graph formed by the real position as coordinates, the graph obtained by the proposed algorithm can accurately represent the relative position graph formed by the real coordinates of the 9 nodes, and the effect is excellent.

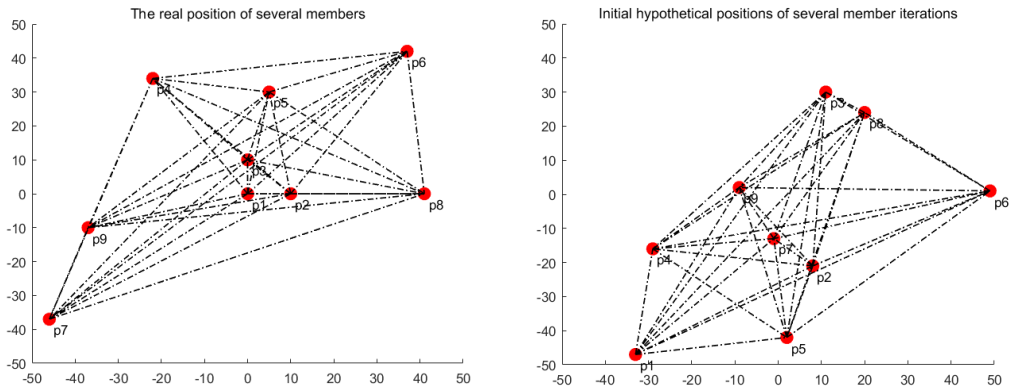
5.1. Relationship Between Topology Structure and Number of Iterations

In this paper, the topology structure of the initial coordinates of each member has a great impact on the performance of the positioning algorithm, so we need to carry out research on the topology structure of the initial coordinates. Through research, it is found that the topology structure of the initial coordinates of each member has the following characteristics:

1. The topology structure formed by the initial coordinates of each member cannot be at the same point, and it is obvious that the real coordinates of each member cannot be at the same position in real life;
2. The topology structure formed by the initial coordinates of each member cannot all be on the same straight line, otherwise the transition coordinates of each member can only move in the horizontal direction, resulting in algorithm failure;
3. The topology structure formed by the initial coordinates of each member should not be too close to each other, otherwise the transition coordinates of each member will be confused together, and the relative position graph formed by them cannot be observed well.

Next, we will change the initial topology of each member in the first experiment by randomly generating the initial coordinates of each member instead of distributing them according to a square pattern. The simulation effect is shown in the following Figure 3.

From Figure 3, we can see that when the topology of the initial coordinates is changed from square to random topology, the required number of iterations is increased by 3 times from 19 to 22 times, which is an increase of 15.79% compared with the first experiment. It can be seen that the algorithm has a good adaptability to the convergence of different topologies. In the existing node positioning problem, when the topology structure is square or network topology, the precision and convergence speed of the node positioning results will be better, so the choice of the topology structure of the initial coordinates is very important in this paper. The algorithm proposed in this paper can satisfy various types of topologies in different localization problems and has broad application prospects.



(**a**) (**b**)

Figure 3. *Cont.*

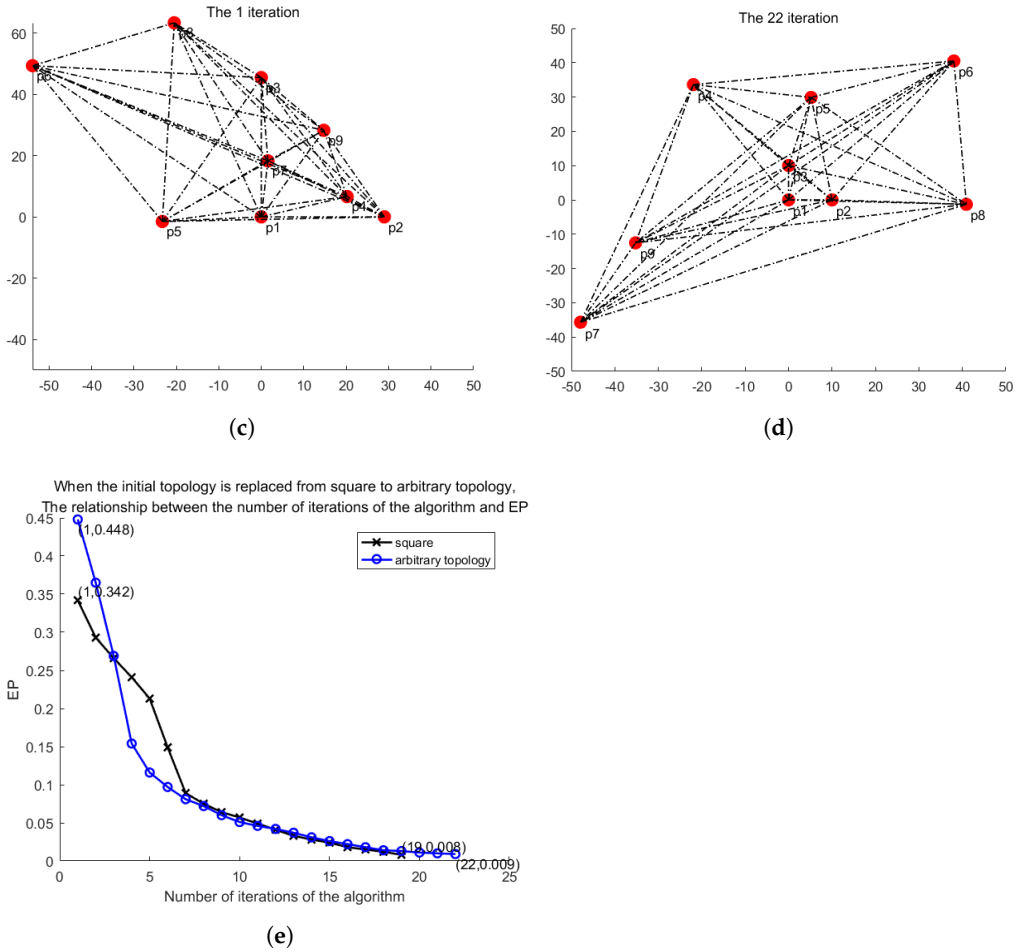
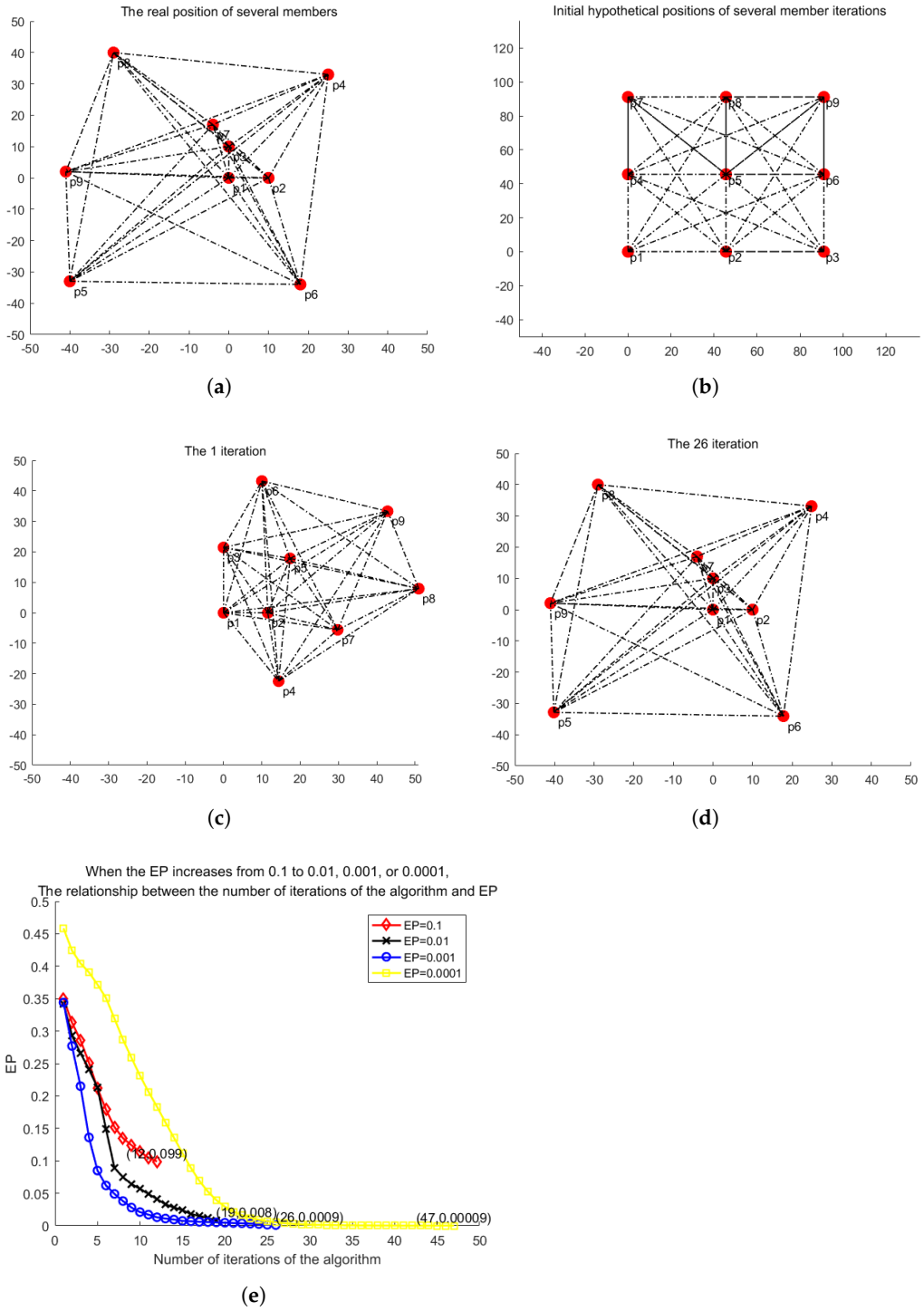


Figure 3. The topological structure of the initial coordinates and the number of iterations: (**a**) the real position several members; (**b**) initial hypothetical positions of several member iterations; (**c**) the first iteration; (**d**) the 22nd iteration; (**e**) when the initial topology is replaced from square to arbitrary topology, the relationship between the number of iterations of the algorithm and EP.

5.2. The Relationship Between Positioning Accuracy and Number of Iterations

The positioning accuracy of the proposed algorithm can be changed according to the actual application requirements, and the positioning accuracy is one of the most important factors to evaluate the quality of an algorithm. The positioning accuracy set in the first experiment is 1%, which can meet most of the positioning requirements under normal circumstances. Next, we improve the positioning accuracy to 1‰, that is, the error ratio is 0.001, and the other simulation conditions remain unchanged, and analyze the relationship between the positioning accuracy of the proposed algorithm and the number of iterations. The simulation effect is shown in Figure 4.

As can be seen from Figure 4, when the positioning accuracy of the algorithm is increased from 10% to 1%, 1‰, and 0.1‰, the number of iterations increases from 12 by 7, 14, and 35 times, respectively, that is, the number of iterations increases by 58.33%, 116.66%, and 291.66%, respectively. Therefore, the proposed algorithm in this paper can achieve high positioning accuracy, because the proposed algorithm does not generate error accumulation in the positioning process, only by constantly correcting the position increment between the transition coordinates and the real coordinates, so the algorithm can quickly and accurately locate the position coordinates of each member during positioning.



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 4. The relationship between positioning accuracy and the number of iterations: (**a**) the real position several members; (**b**) initial hypothetical positions of several member iterations; (**c**) the 1st iteration; (**d**) the 26th iteration; (**e**) when the EP increases from 0.1 to 0.01, 0.001, 0.0001, the relationship between the number of iterations of the algorithm and EP.

5.3. Relationship Between the Number of Nodes N and the Number of Iterations

The problem considered in this paper is to locate the relative position graph between multiple members, and the randomly generated node coordinates should ensure that the connection degree is greater than the specified minimum connection degree. Usually the minimum connection degree is set as 3, that is, $N \geq 3$. Next, we will change the value of

the number of nodes N in the first test, and the remaining simulation conditions remain unchanged. Taking $N = 6$ as an example, we will show the iterative process and give the relationship between the error proportion and the number of iterations when N takes different values. Figure 5 shows the simulation results.

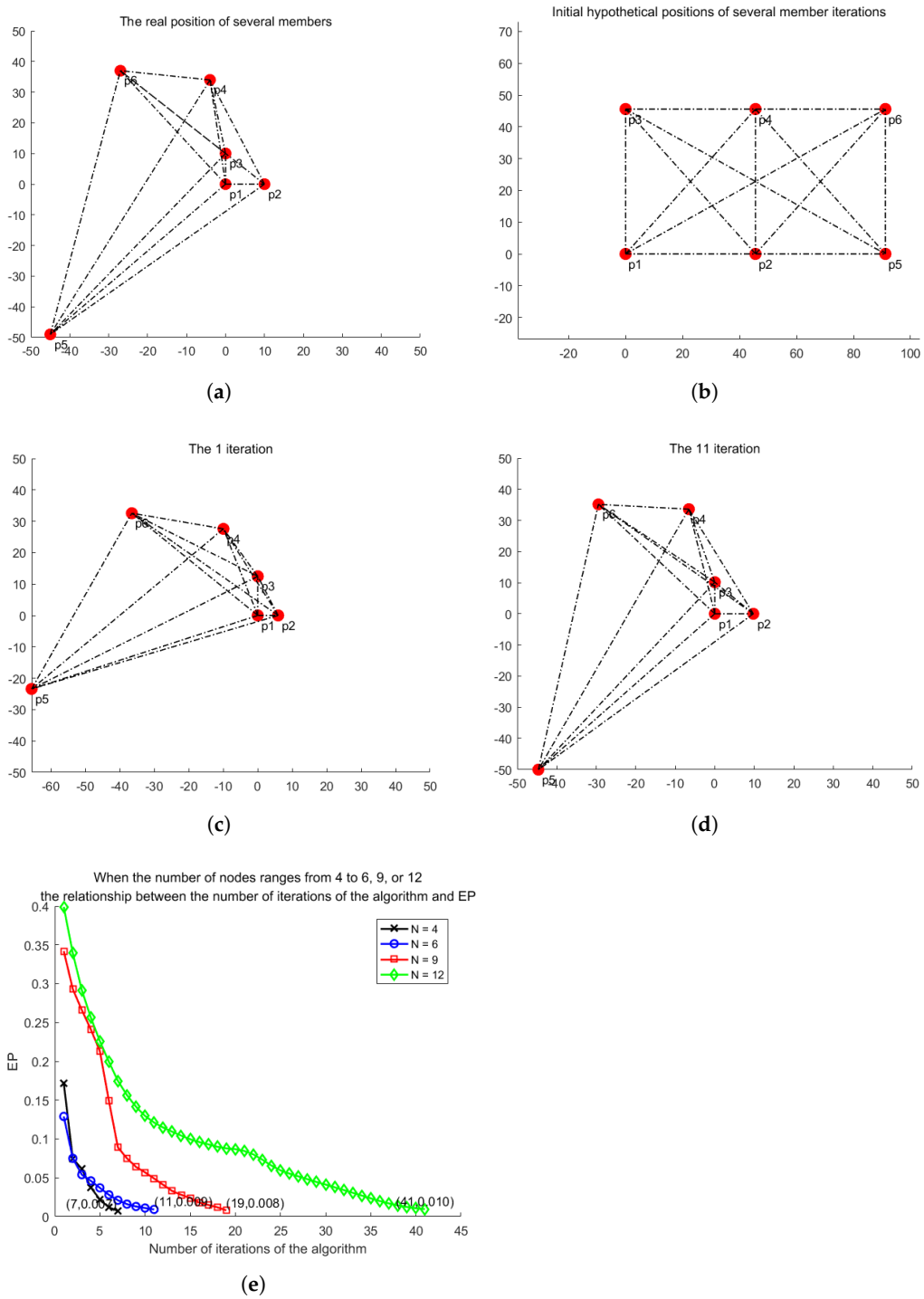


Figure 5. The relationship between the number of nodes N and the number of iterations: (**a**) the real position several members; (**b**) initial hypothetical positions of several member iterations; (**c**) the 1st iteration; (**d**) the 11th iteration; (**e**) when the number of nodes ranges from 4 to 6, 9, or 12, the relationship between the number of iterations of the algorithm and EP.

From Figure 5, we can see that when the number of nodes N is 4, the number of iterations is 7. When increasing from 4 to 6, the number of iterations increases by 4; when

increasing from 6 to 9, the number of iterations increases by 8; when increasing from 9 to 12, the number of iterations increases by 22. With the increase in the number of nodes N, the number of iterations required by the algorithm in this paper also gradually increases, but the number of iterations increased each time by the algorithm, that is, the amount of computation required by the algorithm did not increase much with the increase in the number of nodes. Therefore, the algorithm can be applied to positioning in various complex scenarios and quickly obtain the position coordinates of nodes.

As can be seen from Figure 6, the positioning error of the anchor-free location algorithm based on transition coordinates proposed in this paper decreases with the increase in the number of nodes. Compared with the classical algorithm, the AFL, R-AFL, and MDS-MAP algorithms have the smallest positioning error and the best positioning performance. Secondly, the positioning error of R-AFL algorithm is better than that of traditional AFL and MDS, because the coordinate system established by R-AFL in the positioning process is more accurate and the adopted algorithm optimizes the position of nodes in the network, so it has better positioning performance. The AFL algorithm performs better than MDS when the number of nodes is small, but its performance is surpassed by MDS when the number of nodes increases gradually. In general, the proposed algorithm has some advantages compared with the existing algorithms.

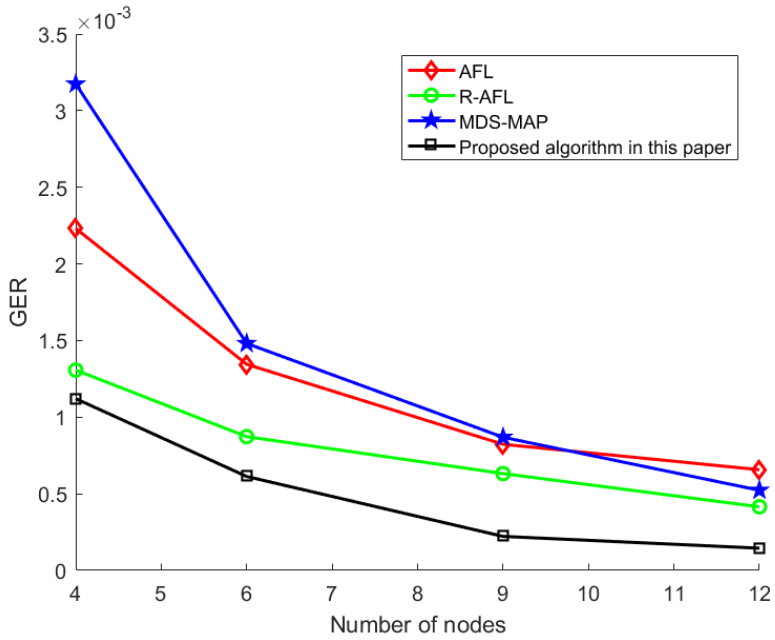


Figure 6. GER error of four different positioning algorithms with different numbers of nodes.

6. Conclusions

In this paper, we propose an anchor-free location algorithm based on transition coordinates. The algorithm does not use the traditional methods based on cost function minimization or matrix inversion, but simply gives the difference of position vector according to the difference between the temporary distance and the real distance, and iterates gradually to approach the real coordinates. The computational complexity of this algorithm is much less than that of the traditional algorithm, and it can guarantee its convergence and achieve high precision for most topologies, and can meet the needs of most positioning scenarios, especially in the case of weak GPS signals or short-time position changes multiple times.

In the simulation experiment, we investigated some factors that may affect the performance of the algorithm, including the initial topology of the transition coordinates, the positioning accuracy required by the algorithm, and the total number of nodes required by the algorithm. We also list the places that attention should be paid to in the application

of the algorithm, such as where the real coordinate of the node cannot be in the same position or a straight line, the distance between each node cannot be too low, or the effect of the algorithm may be poor. We also compare the computational complexity between the proposed algorithm and the existing centralized algorithm, semi-centralized algorithm, and distributed algorithm in the experiment. It can be seen that the computational complexity of the proposed algorithm is small, and the relative position graph between nodes can be obtained at a faster speed. At the same time, the global energy ratio (GER) shows great advantages over the classical AFL, R-AFL, and MDS-MAP algorithms. In the future, the goal of our work is to consider how to apply it to the field of moving coordinates or three-dimensional spatial positioning, how to combine the proposed algorithm with the anchored location algorithm, and how to optimize the convergence rate of the algorithm so as to further reduce the computation amount or improve the accuracy of the algorithm.

Author Contributions: Conceptualization, S.L.; methodology, J.F. and S.L.; validation, J.F.; resources, S.L.; data curation, J.F.; writing—original draft preparation, J.F.; writing—review and editing, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Professor Huang Shuanglin's Hubei Provincial Natural Science Foundation (2024AFB872).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: We choose not to disclose data due to privacy reasons.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GPS	Global positioning system
AFL	Anchor-free localization
CBD	Centralized boundary discovery
ABC	Assumption-based coordinates
MDS	Multidimensional scaling
R-AFL	Refined anchor-free localization
MDS-MAP	Multidimensional scaling-map
EP	Error proportion
GER	Global energy ratio

References

1. Zhao, S.; Zhang, X.-P.; Cui, X.; Lu, M. A new toa localization and synchronization system with virtually synchronized periodic asymmetric ranging network. *IEEE Internet Things J.* **2021**, *8*, 9030–9044. [CrossRef]
2. Li, M.; Wang, W.; Zhang, J.; Zhang, Q. Incentivizing wifi-based multilateration location verification. *IEEE Internet Things J.* **2021**, *9*, 3083–3096. [CrossRef]
3. Zhang, S.; Xu, Y.; Tang, B.; Yang, Y.; Sun, Y. Efficient solution resilient to noise and anchor position error for joint localization and synchronization using one-way sequential toas. *Appl. Sci.* **2024**, *14*, 6069. [CrossRef]
4. Zhao, Q.; Xu, Z.; Yang, L. An improvement of dv-hop localization algorithm based on cyclotomic method in wireless sensor networks. *Appl. Sci.* **2023**, *13*, 3597. [CrossRef]
5. Su, Y.; Guo, L.; Jin, Z.; Fu, X. A mobile-beacon-based iterative localization mechanism in large-scale underwater acoustic sensor networks. *IEEE Internet Things J.* **2020**, *8*, 3653–3664. [CrossRef]
6. Banavar, M.K.; Wickramasinghe, S.; Achalla, M.; Sun, J. Ordinal unloc: Target localization with noisy and incomplete distance measures. *IEEE Internet Things J.* **2021**, *8*, 17111–17119. [CrossRef]
7. Liu, X.; Chen, H.-X.; Liu, B.-Y. Dynamic anchor: A feature-guided anchor strategy for object detection. *Appl. Sci.* **2022**, *12*, 4897. [CrossRef]
8. Zhao, S.; Zhang, X.-P.; Cui, X.; Lu, M. A closed-form localization method utilizing pseudorange measurements from two nonsynchronized positioning systems. *IEEE Internet Things J.* **2020**, *8*, 1082–1094. [CrossRef]
9. Sadowski, S.; Spachos, P.; Plataniotis, K.N. Memoryless techniques and wireless technologies for indoor localization with the internet of things. *IEEE Internet Things J.* **2020**, *7*, 10996–11005. [CrossRef]

10. Bi, S.; Cui, J.; Ni, W.; Jiang, Y.; Yu, S.; Wang, X. Three-dimensional cooperative positioning for internet of things provenance. *IEEE Internet Things J.* **2022**, *9*, 19945–19958. [CrossRef]
11. Priyantha, N.B.; Balakrishnan, H.; Demaine, E.; Teller, S. Anchor-free distributed localization in sensor networks. In Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, Los Angeles, CA, USA, 5–7 November 2003; pp. 340–341.
12. Simek, M.; Komosny, D.; Burget, R.; Moravek, P.; Silva, R. Centralized boundary discovery algorithms for anchor-free localization in wireless sensor networks. In Proceedings of the 2009 International Conference on Ultra Modern Telecommunications & Workshops, St. Petersburg, Russia, 12–14 October 2009; pp. 1–7.
13. Savarese, C.; Rabaey, J.M.; Beutel, J. Location in distributed ad-hoc wireless sensor networks. In Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, Salt Lake City, UT, USA, 7–11 May 2001; pp. 2037–2040.
14. Li, D.; Chen, L.; Hu, J.; Wu, H. Research on uwb positioning based on improved abc algorithm. In Proceedings of the 2021 4th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), Yibin, China, 20–22 August 2021; pp. 81–85.
15. Shang, Y.; Ruml, W.; Zhang, Y.; Fromherz, M.P. Localization from mere connectivity. In Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing, Annapolis, MD, USA, 1–3 June 2003; pp. 201–212.
16. Wang, X.; Qiu, J.; Fan, J.; Dai, G. Mds-based localization scheme for large-scale wsns within sparse anchor nodes. In Proceedings of the 2015 IEEE International Conference on Communications (ICC), London, UK, 8–12 June 2015; pp. 6609–6614.
17. Youssef, A.; Agrawala, A.; Younis, M. Accurate anchor-free node localization in wireless sensor networks. In Proceedings of the PCCC 2005. 24th IEEE International Performance, Computing, and Communications Conference, Phoenix, AZ, USA, 7–9 April 2005; pp. 465–470.
18. Kazuma, S.; Kazushi, S.; Akihiro, K.; Shunsuke, S.; Ryusuke, F. Development of a behavioral trajectory measurement system (bucket-antam) for organisms moving in a two-dimensional plane. *Artif. Life Robot.* **2022**, *27*, 698–705.
19. Huang, S.-C.; Wang, Y.-K. Mobile anchor route scheduling with an iterative sensor positioning algorithm in wireless sensor networks. *Appl. Sci.* **2022**, *13*, 22. [CrossRef]
20. Saeed, N.; Nam, H.; Al-Naffouri, T.Y.; Alouini, M.-S. A state-of-the-art survey on multidimensional scaling-based localization techniques. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3565–3583. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.