Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

WSNs Node localization algorithm based on Multi-Hop Distance Vector and Error Correction

Ke Zhang
Wuhan University of Technology - Mafangshan Campus: Wuhan University of Technology
Guang Zhang
Wuhan University of Technology
Xiuwu Yu (yxw_usc@163.com)
University of South China https://orcid.org/0000-0003-0887-8830
Shaohua Hu
Wuhan University of Technology - Mafangshan Campus: Wuhan University of Technology
Youcui Yuan
Wuhan University of Technology - Mafangshan Campus: Wuhan University of Technology

Research Article

Keywords: Wireless Sensor Networks(WSNs), Range-free, Node localization, Error correction, Local similarity

Posted Date: April 4th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1406414/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

WSNs Node localization algorithm based on Multi-Hop Distance Vector and Error Correction

Ke Zhang[1], Guang Zhang[1], Xiuwu Yu[2*], Shaohua Hu[1], Youcui Yuan[1]

(1 School of Safety Science and Emergency Management, Wuhan University of Technology, Wuhan 430070 China. 2 School of Resource & Environment and Safety Engineering, University of South China, Hengyang 421001, China)

E-mail: Ke Zhang (zhangkeblue@163.com)

Abstract: Wireless sensor networks (WSNs) have broad application prospects in various industries, and node localization technology is the foundation of WSN applications. Recently, many range-free node localization algorithms have been proposed, but most of them suffer from low accuracy. In order to improve the localization accuracy, in this paper we proposed the node localization algorithm based on multi-hop distance vector and error correction(MDV-EC). In terms of distance estimation, firstly the MDV-EC algorithm calculates the neighbor distance according to node neighbor relationship, then estimates the distance between unknown node and anchor node in multi-hop manner, and finally calibrates the distance refer to distance correction coefficient. In view of similarity of localization errors of nodes in similar regions, an error correction scheme is also investigated, which corrects the node initial estimated locations of nodes refer to the localization error vector of nearby anchor node. Simulation results show that our proposed MDV-EC has better performance than the other two algorithms in terms of node localization accuracy, and the error correction scheme can effectively reduce the localization errors.

Keywords: Wireless Sensor Networks(WSNs), Range-free, Node localization, Error correction, Local similarity

1 Introduction

Wireless sensor networks (WSNs) are composed of many sensor nodes with wireless communication capability, which is a key technology for the Internet of Things (IoT) and has broad application prospects in many fields[1]. Node localization is one of the critical issues of WSNs system, and the location information of nodes is crucial for WSNs, as the sensor data without location information will become meaningless[2]. While traditional localization technologies such as GPS can provide good location information, equipping all nodes with GPS chips will undoubtedly increase the cost of system deployment. In addition, sensor nodes have limited energy and usually cannot be charged in a timely manner. Equipping the nodes with GPS will consume a lot of energy and reduce the lifetime of the system. Therefore, in practice only a few nodes called anchor nodes are equipped with GPS to obtain their location, while other nodes apply the localization algorithm to estimate their locations with the assistance of anchor nodes[3].

In recent years, researchers have proposed many node localization algorithms, which can be roughly divided into two categories: range-based and range-free localization algorithms. The ranging-based localization algorithm uses ranging techniques to obtain the distance between nodes and estimate the position based on the coordinates of the anchor nodes and distance estimation information. Popular ranging techniques, such as time of arrival (TOA), time difference of arrival (TDOA), angle of arrival(AOA) and received signal strength(RSS)[4-5]. Among them, TOA, TDOA and AOA have good ranging accuracy, but these methods require additional hardware, which raises the system deployment cost and is not suitable for large-scale deployment. RSS-based ranging is used to estimate the distance based on the propagation attenuation characteristics of wireless signals in the environment. This method has gained the attention of many researchers because it does not require additional hardware and has the advantage of low cost. However, the robustness of RSS-based ranging technique is poor, and its ranging accuracy is easily affected by environmental factors. Compared with ranging-based localization algorithms, ranging-free algorithms use network connectivity to estimate the distance between nodes or estimate node locations directly based on connectivity, which are less affected by the environment and more robust. Classical rang-free localization algorithms such as centroid algorithm[6], approximate Point-In-Triangulation(APIT)[7], and Distance Vector Hop(DV-Hop)[8]. The most popular range-free algorithms are the centroid algorithm, approximate point-in-triangle (APIT), and distance vector hopping (DV-Hop). DV-Hop has gained wide attention because of its simplicity and high robustness. The DV-Hop algorithm is a typical distance vector based node localization algorithm, which estimates the distance between the unknown node and the anchor node by exploring network connectivity, and then calculates the estimated location of the unknown node according to the estimated distance and the location of the anchor node. Although the DV-Hop algorithm has the advantages of computational simplicity and environmental adaptability, its localization accuracy is far from satisfactory. Therefore, many researchers have proposed a number of improved algorithms for DV-Hop. An improved algorithm named DV-MaxHop is proposed in Reference [9]. The DV-MaxHop defines the maximum hop count to constrain the information propagation in the network, ignores the distal anchor nodes with large hop counts, and obtains a more accurate average hop value to improve the distance estimation accuracy. To enhance the location estimation accuracy, a heuristic algorithm is introduced to solve the weighted objective function, and find the optimal estimated location. The simulation results show that the proposed method has better localization accuracy than DV-Hop. A centroid DV-hop localization with selected anchors and inverse distance weighting schemes(SIC-DV-Hop) was proposed in [10]. The SIC-DV-Hop employs an inclusive check rule to select appropriate anchor points to avoid inconsistencies and introduces a distance weighting scheme to improve the location estimation accuracy. Paper [11] proposed the DANS algorithm, in which the authors argue that different anchor nodes provide different reliability of information, and by referring to the information of some anchor nodes, better localization accuracy can be obtained. The DANS algorithm investigates a binary particle swarm algorithm to select reliable anchor nodes that provide distance estimation information and determine the final estimated position, avoiding the interference of unreliable anchor nodes with a large range of errors and improving the localization accuracy. The hop-count based distance estimation method estimates the distance in terms of integer hops, which leads to a large error in distance estimation. Therefore, paper [12] proposed an RSS-based hop count estimation method that uses node RSS information to serialize the hop estimation. In addition to improving the node localization accuracy from improving the distance estimation accuracy, some improved works also improve the accuracy from the location estimation aspect. These approaches first transform the location estimation problem into an optimization problem and apply heuristic optimization algorithms, such as particle swarm optimization and genetic algorithms, to iteratively find the best estimated coordinates[13]. Although these improved algorithms provide a significant improvement in accuracy compared to the traditional DV-Hop algorithm, estimating the distance between nodes based on the number of hops and the average hop distance often leads to large estimation errors and results in large distance estimation errors and low localization accuracy. In view of this, some researchers try to estimate the distance refer to the node connectivity and in a multi-hop manner. Paper [14] proposed a localization algorithm called LEAP, which first investigates the functionre lationship between distance and neighbor information, then estimate distance between nodes according to this function, and finally calculates the estimated location of the unknown node based on the estimated distance. paper [15] analyzes the relationship between the distance between neighboring nodes and the number of common neighboring nodes, and proposes a distance vector localization algorithm based on regular neighbor distance(DV-RND). The DV-RND estimates the distance of neighbor nodes based on the neighbor relationship, then gets the estimated distance between the unknown nodes and each anchor, and finally calculates the estimated location. To improve the performance of the algorithm for localization in isotropic deployment, paper [16] combines the Max-hop method with the ranging method in LEAP, thus reducing the influence of isotropic environments on ranging and enhancing the ranging accuracy of every hop progress. An accurate multi-hop node localization algorithm is proposed in [17]. The proposed algorithm obtains the estimated distance between nodes based on the neighbor information and solves the location estimation problem by the hyperbolic estimation method, while geometric constraints are applied to further reduce the error in location estimation.

For the distance vector node localization algorithm, the estimated distances between nodes are related to the network connectivity of the path region, and neighboring nodes have a large overlap area with the path region of each anchor node, which leads to similar ranging and localization errors of neighboring nodes. Most distance vector localization algorithms, however, estimate distances or calibrate distances from a global perspective, ignoring local differences in node distributions, resulting in large distance estimation errors and low localization accuracy. From this aspect, in order to improve the node localization accuracy, the MDV-EC is proposed in this paper. The MDV-EC algorithm consists of two parts: initial location estimation and error correction. In the initial position estimation phase, the distance between neighboring nodes is first estimated based on the neighbor relationship, the estimated distance between the unknown node and the anchor node is obtained by exploring the shortest path between nodes, and finally, the initial estimated position of the node is obtained by the least squares method. In the error correction phase, all anchor nodes calculate their estimated positions according to the node position estimation method, and the error vector of each anchor node is obtained with their estimated and real positions. Given the similarity of the error vectors, the error vectors of the anchor nodes are used to correct the initial estimated locations of the unknown nodes around the anchor nodes, further reducing the localization errors. The simulation results demonstrate that the localization accuracy of our proposed MDV-EC algorithm outperforms DV-Hop and DV-RND in different scenarios, and the error correction method of the proposed algorithm can further improve the localization accuracy.

The rest of this paper is organized as follows. Section 2 illustrates the network model and problem statement, Section 3 describes our proposed MDV-EC localization algorithm, Section 4 evaluates the performance of the proposed algorithm, and the last section concludes the paper and discusses future research directions.

2 Network model and problem statements

In the real deployment scenario, the communication coverage area of nodes is not a standard circle considering environmental factors. To simulate the signal propagation in the real scenario, this paper adopts the irregularity degree model (DOI) to simulate the coverage area of the nodes in the network [18]. According to the DOI model, nodes have different signal strengths in different directions, and the signals in adjacent directions have a certain correlation. Dividing the space into 360 directions, the signal strength of each direction can be calculated by equation (1).

$$P_r = P_t - P(d_0) - 10 \times \eta \times log_{10}(\frac{d_i}{d_0}) \times K_i \tag{1}$$

Where P_r represents the signal strength received by the node, P_t represents the signal strength of the message sent, η represents the signal attenuation constant (path loss exponent), d_0 represents the reference distance, $P(d_0)$ is the received signal strength at the reference distance, and K_i is the direction parameter of the DOI model, the value is calculated from equation (2).

$$K_i = \begin{cases} 1 & if \ \ i = 0 \\ K_{i-1} + rand_i & else, i \in [1,2,\ldots,360] \end{cases} \tag{2}$$

Where $rand_i$ is the direction random parameter, $rand_i \in$[-DOI, DOI]. The DOI value controls the stability of the signal; a larger value means a more unstable signal and a smaller value means a more stable signal. When DOI is equal to 0, the coverage area of the signal is a standard circle. To describe the model more specifically, we set the communication radius R=30 ,and plot the signal coverage areas with DOI=0.01 and DOI=0.02, respectively, as shown in figure 1.

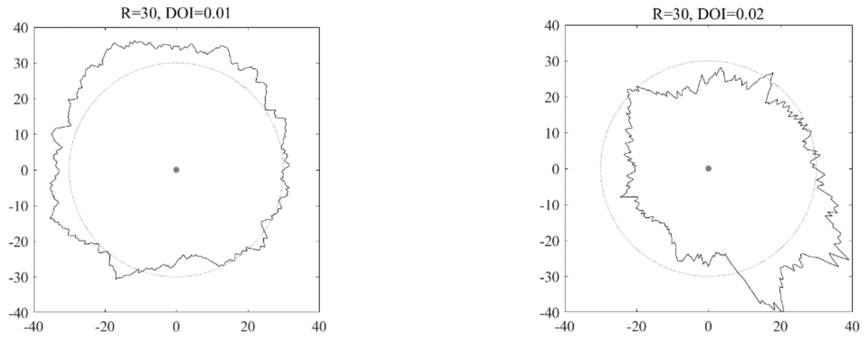


Figure 1 DOI model

In this paper, we assume that WSNs system satisfy the following conditions.

1. All nodes are randomly and uniformly arranged in deployment area, forming a connectivity graph in which any two nodes can communicate in a single-hop or multi-hop manner.

2. Some of the nodes in the network have determined and known locations, and they are called anchor nodes.

3. All nodes in the network have the same communication radius R. The communication model of all nodes is the DOI model.

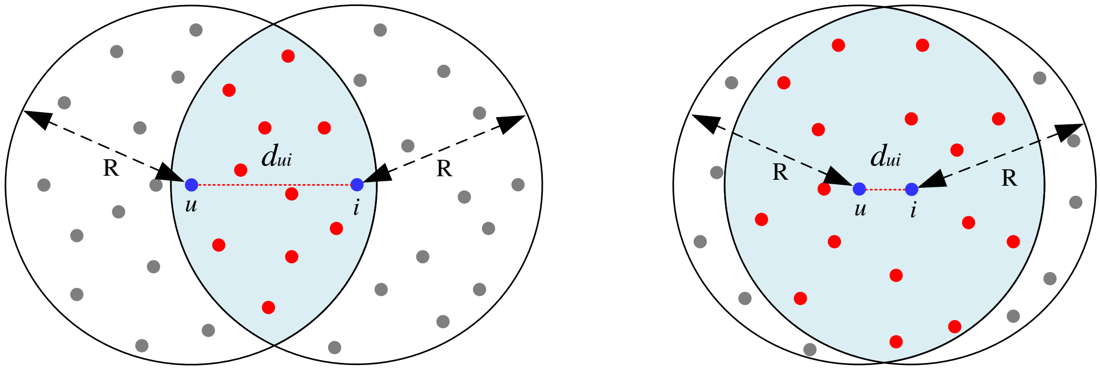
4. The connectivity of any two nodes in the network is symmetric, i.e., when node i can communicate directly with node j, then node j must be able to communicate directly with node i.

The node localization problem in this paper is to obtain the geographic location of all sensor nodes with the assistance of anchor nodes and the network connectivity information.

3 Proposed node localization algorithm

3.1 Neighbor distance estimation

As shown in figure 2, node u and i are two neighboring nodes, and their communication radius is R. The common communication coverage area is the light blue part. In a network with uniform random deployment of sensor nodes, the closer two adjacent nodes are, the larger their communication common coverage area is and the more nodes fall into the common coverage area. Similarly, the farther two nodes are from each other, the smaller their communication common coverage area is and the fewer nodes fall into the common coverage area. As a result, we can estimate the distance of two adjacent nodes by exploring their neighbor information.



(a) (b)

Figure 2 Neighbor distance estimation

In random uniform deployment network, according to the geometric relationship, the distance d_{ui} between two neighboring nodes u, i is a function of the common coverage area $A(d_{ui})$, as shown in equation (3) [19,20].

$$A(d_{ui}) = 2R^2\arccos(\frac{d_{ui}}{2R}) - \frac{d_{ij}}{2}\sqrt{4R^2 - d_{ui}^2} \tag{3}$$

Let $x=d_{ui}/2R$ and bring it into equation (3) to obtain the following equation.

$$\frac{A(d_{ui})}{R^2} = 2\arccos(x) + 2x\sqrt{1-x^2} \tag{4}$$

The Taylor expansion of the right part of the equation (4) at $x= 0$ gives formula (5).

$$\begin{cases} \arccos(x) = \frac{\pi}{2} - x - \frac{1}{6}x^3 - \frac{3}{40}x^5 \ldots \\ x\sqrt{1-x^2} = x - \frac{1}{2}x^3 - \frac{1}{8}x^5 \ldots \end{cases} \tag{5}$$

Bring euquation (5) into equation (4), we get equation (6).

$$\frac{A(d_{ui})}{R^2} = \pi - 4x + \frac{2}{3}x^3 + \frac{1}{10}x^5 \ldots \tag{6}$$

Since nodes u and i are each other's neighbor nodes, their distance d_{ui} is smaller than the communication radius R. Because $x \le d_{ui}/2R$, we have $x \le 0.5$. Since x is less than 0.5, the value of the higher order term of x is very small. Therefore, ignoring the x^3 and higher terms in equation (6), taking $x=d_{ui}/2R$ into equation (6) and simplifying it, we obtain equation (7) as follows.

$$d_{ui} = \frac{\pi R^2 - A(d_{ui})}{2R} \tag{7}$$

According to equation (7), we can estimate the neighbor node distance d_{ui} based on the communication radius R and the common coverage area $A(d_{ui})$. The node communication radius R is predefined, and the common coverage area $A(d_{ui})$ can be estimated from the neighbor node relationship. Next, we estimate the common coverage area $A(d_{ui})$ of the communication range of the two neighboring nodes u, i based on their neighbor relationship. The node density of $A(d_{ui})$ can be obtained by dividing the number of common neighbor nodes of u, i by the area of $A(d_{ui})$. Similarly, the node density of the communication coverage area can be obtained by dividing the number of neighbor nodes by the area of the node communication range. Since the nodes are randomly and uniformly arranged, the node density is the same in all regions.

According to this feature, the node density of the common communication coverage area is the same as the node density within the communication range of each node, which leads to equation (8).

$$\frac{A(d_{ui})}{CN_{ui}} = \frac{\pi R^2}{M_u} \tag{8}$$

Where CN_{ui} represents the number of common neighbor nodes, M_u represents the number of neighbors of node u. It is worth noting that the number of both CN_{ui} and M_u contain node u, i itself. From equation (8), we know that the estimated coverage area $\hat{A}(d_{ui})$ of two nodes u, i can be calculated according to the number of neighboring nodes of u, i and the communication radius R, as shown in equation (9).

$$\hat{A}(d_{ui}) = \frac{CN_{ui}}{M_u}\pi R^2 \tag{9}$$

Equation (9) uses the number of neighboring nodes of u and the number of common neighbor nodes of u, i to estimate the common coverage area $\hat{A}(d_{ui})$. Similarly, we can estimate the $\hat{A}(d_{ui})$ according to the number of neighboring nodes of i and the number of common neighboring nodes. To avoid estimation bias, we take the average of the two estimation methods as the final estimate of $\hat{A}(d_{ui})$, as shown in equation (10).

$$A(d_{ui}) = \frac{\hat{A}(d_{ui})+\hat{A}(d_{iu})}{2} = \frac{(NB_u+NB_i)CN_{ui}}{2NB_uNB_i}\pi R^2 \tag{10}$$

To simplify the presentation, we define the intersection degree I_{ui} associated with nodes u and i as shown in equation (11).

$$I_{ui} = \frac{(NB_u+NB_i)CN_{ui}}{2NB_uNB_i} \tag{11}$$

Taking equation (11) into equation (10) yields equation (12).

$$A(d_{ui}) = \pi R^2 I_{ui} \tag{12}$$

Simultaneous equations (7) and (12), after eliminates $A(d_{ui})$ we can obtain the functions of d_{ui} and I_{ui} as shown in equation (13).

$$d_{ui} = \frac{(1-I_{ui})\pi R}{2} \tag{13}$$

According to equation (13), we can use the intersection degree I_{ui} of two nodes to calculate the estimated distance d_{ui} of two nodes, the I_{ui} can be obtained from the adjacency information. Referring to equation (13), all nodes can estimate the distance to their neighbor nodes.

3.2 Preliminary location estimate

After initialization, all nodes obtain their neighboring information by data exchange and calculate the estimated distance to neighbor nodes according to the method illustrated in Section 3.1. After the neighbor distance estimation, all anchor nodes flood their information to the whole network, and all nodes get the estimated distance to each anchor node during flooding. To improve the accuracy of distance estimation, each anchor node calculates its distance correction coefficient(*DCC*) based on the anchor position information and the estimated distance. The unknown node corrects the estimated distance from the anchor node based on *DCC*. The preliminary unknown estimates of nodes are as follows.

Suppose there are n anchor nodes in the deployment area and the set of anchor nodes is [$A1$, $A2$, ... , An]. After deployment, all nodes broadcast a "*hello*" message containing their ID number and receive "*hello*" messages from neighboring nodes. In this way, node neighbor node information is collected. Then, exchange neighbor node information with all neighboring nodes and estimate the distance between the node itself and its neighboring nodes. After neighboring node distance estimation, all anchor nodes broadcast "*flood message*" message and flood it to the whole network. The format of the "*flooding message*" is *flooding_msg*$\{Ai, d_{ac}, N_{ID}\}$, where Ai represents the ID of the anchor node to which the flooding message belongs, d_{ac} represents the cumulative distance of the message propagation, i.e., the cumulative estimated distance of the message from the anchor node, and N_{ID} represents the set of relay nodes. Before the flooding starts, the node initializes the estimated multi-hop distance [d_{A1}, d_{A2}, … d_{Ai}] with each anchor node, where all elements are initialized to null. When unknown node k receives the "*flooding_msg*" message belonging to anchor node Ai, it first refers the estimated distance d_t between the node itself and the relay node and calculates the cumulative estimated distance FD_{Ai} of anchor node Ai in the "*flooding_msg*" propagation path according to equation (14).

$$FD_{Ai} = d_{ac} + d_t \tag{14}$$

After updating the cumulative distance, the node checks whether there is multi-hop estimated distance information of anchor node Ai in memory, if there is no such anchor node information, it saves the updated FD_{Ai} as the shortest multi-hop estimated distance d_{Ai} of anchor node Ai, updates d_{ac} in "*flooding_msg*" as d_{Ai}, and N_{ID} is updated to the set of neighbor nodes of the receiving node itself, and forwards it after the update. If a node has multi-hop estimated distance information of anchor node Ai in its memory, then it compares FD_{Ai} with d_{Ai} and keeps the minimum value. In this way, all nodes can get their shortest estimated distances from all anchor nodes. The flowchart of this process is shown in figure 3.

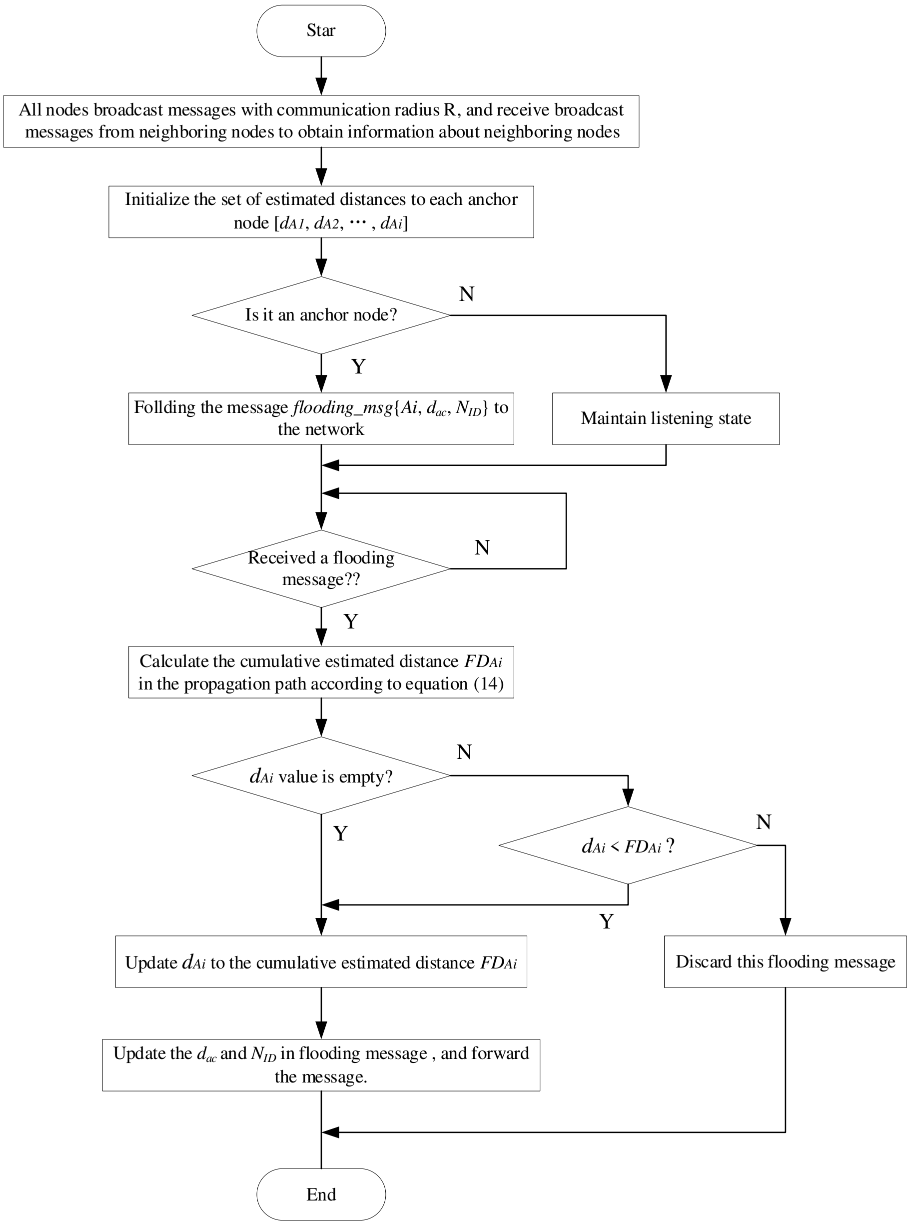


Figure 3 Network flooding process

After the network flooding, all nodes get the estimated distance from all anchor nodes, and all anchor nodes calculate their respective *DCC* based on the estimated distance from other anchor nodes according to equation (15).

$$DCC(A_i) = \frac{\sum_{i\in AN, i\neq Ai} \sqrt{(x_{Ai}-x_i)^2+(y_{A_i}-y_i)^2}}{\sum_{i\in AN, i\neq Ai} d_{ki}} \tag{15}$$

Where (x_{Ai}, y_{Ai}) and (x_i, y_i) represent the location of anchor node *Ai*, and *i* respectively, d_{ki} represents the shortest estimated distance between unknown node *k* and anchor node *i*. It is worth noting that each anchor node gets its own correction coefficient by the above method instead of getting the average correction coefficient of the network as a whole so that the local characteristics of the respective anchor node can be preserved. After all anchor nodes obtain their *DCC* values, they flood the *DCC* value to the whole network. Therefore, the *DCC* values of all anchor nodes are available for all nodes. Each node uses *DCC* to correct the estimated distance and obtains the final estimated distance according to equation (16).

$$\hat{d}_{ki} = DCC(Ai)\times d_{ki} \tag{16}$$

Where, d_{ki} and $\hat{d}_{ki}$ represent estimated distance and final estimated distance respectively. After obtaining the final estimated distances to all anchor nodes, the preliminary estimated positions of the nodes are obtained by using the least squares method. The initial location estimation procedure is as follows.

$$\begin{cases} (x_{A1} - x_k)^2 + (y_{A1} - y_k)^2 = \hat{d}_{k1}^2, \\ (x_{A2} - x_k)^2 + (y_{A2} - y_k)^2 = \hat{d}_{k2}^2, \\ \quad\quad\quad \cdots, \\ (x_{An} - x_k)^2 + (y_{An} - y_k)^2 = \hat{d}_{kn}^2 \end{cases} \tag{17}$$

Where (x_{Ai}, y_{Ai}) represents the coordinate of the anchor node Ai, $\hat{d}_{ki}$ is the final estimated distance between the unknown node k and the anchor node Ai, and (x_k, y_k) is the coordinate of node k. The first n-1 terms of equation (17) are subtracted from the last term to obtain equation (18). The equation (18) can be written in the form of ***AX***=***B***, where ***A***, ***B***, and ***X*** are shown in equation (19)-(21).

$$\begin{cases} -2(x_1 - x_{An})x_k - 2(y_{A1} - y_{An})y_k = d_{1k}^2 - d_{Ank}^2 - x_1^2 + x_{An}^2 - y_1^2 + y_{An}^2 \\ -2(x_2 - x_{An})x_k - 2(y_{A2} - y_{An})y_k = d_{2k}^2 - d_{Ank}^2 - x_2^2 + x_{An}^2 - y_2^2 + y_{An}^2 \\ \cdots \\ -2\left(x_{A(n-1)} - x_{An}\right)x_k - 2\left(y_{A(n-1)} - y_{An}\right)y_k = d_{A(n-1)k}^2 - d_{Ank}^2 - x_{A(n-1)}^2 + x_{An}^2 - y_{A(n-1)}^2 + y_{An}^2 \end{cases} \tag{18}$$

$$\boldsymbol{A} = -2\begin{bmatrix} x_1 - x_{An}, & y_1 - y_{An} \\ x_2 - x_{An}, & y_2 - y_{An} \\ \cdots & \\ x_{A(n-1} - x_{An}, & y_{A(n-1)} - y_{An} \end{bmatrix} \tag{19}$$

$$\boldsymbol{B} = \begin{bmatrix} d_{1k}^2 - d_{ANk}^2 - x_1^2 + x_{AN}^2 - y_1^2 + y_{AN}^2 \\ d_{2k}^2 - d_{ANk}^2 - x_2^2 + x_{AN}^2 - y_2^2 + y_{AN}^2 \\ \cdots \\ d_{(AN-1)k}^2 - d_{ANk}^2 - x_{(AN-1)}^2 + x_{AN}^2 - y_{(AN-1)}^2 + y_{AN}^2 \end{bmatrix} \tag{20}$$

$$\boldsymbol{X} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} \tag{21}$$

The linear equation ***AX***=***B*** is solved by the least-squares method to obtain the solution of ***X*** as in shown in formula (22).

$$\boldsymbol{X} = (\boldsymbol{A}\boldsymbol{A}^{\boldsymbol{T}})^{-1}\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{B} \tag{22}$$

3.3 Location error vector correction

The distance vector range-free localization algorithms estimate the distance between the unknown node and the anchor node by investigating the network connectivity, then calculate the unknown node location according to the estimated distance. For these localization algorithms, there is similarity in the network connection between two nodes with similar locations. Therefore, there is also similarity in the distance estimation errors between neighboring nodes and anchor nodes, which leads to similarity in their localization errors. To illustrate this phenomenon, we set 200 nodes to be randomly deployed in a 100m×100m square area, of which 20 are anchor nodes. Apply two representative distance vector range-free localization algorithms DV-Hop and DV-RND to locate unknown nodes in the network. The localization error vectors of all sensor nodes are shown in Figure 4.

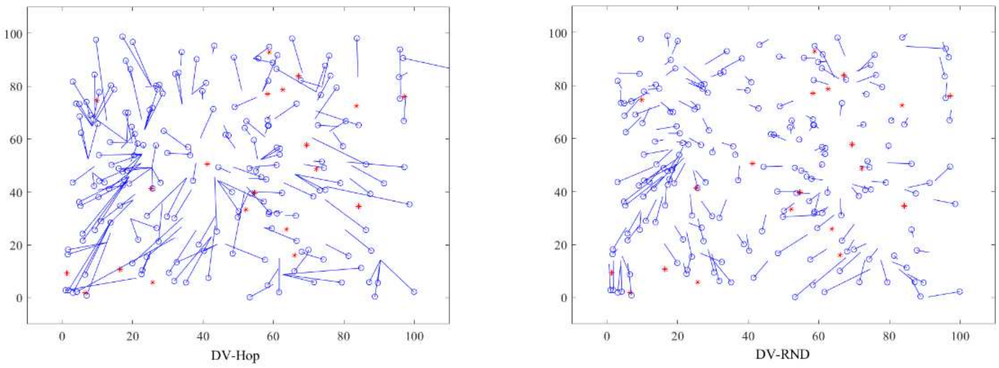


Figure 4 Localization error description

In Figure 4, the blue circle represents the true unknown node location, the red asterisk represents the anchor node location, and the blue line represents the error vector of the node. One end of the blue line connects the real position of the node and the other end is the estimated position of the node. It can be seen from Figure 4 that the error vectors of geographically close nodes have significant similarity in error values and directions. Based on the above observations, we propose the following error correction method.

Define the location error vector (LEV) of the anchor node as the vector of the estimated position of the anchor node pointing to the true position. The direction of LEV is from the estimated location to the true location. Before calculating the anchor node LEV, the anchor node itself is considered as an unknown node, and the position of itself is estimated refer to the coordinate information and estimated distance of all anchor nodes (including itself), where the estimated distance from itself is set to 0. The LEV of the anchor node i is $\boldsymbol{V}(i)$, which is determined by equation (23).

$$\boldsymbol{V}(i) = (\hat{x}_{Ai} - x_{Ai},\ y_{Ai} - \hat{y}_{Ai}) \tag{23}$$

Where $(\hat{x}_{Ai}, \hat{x}_{Ai})$ represents the estimated position of anchor node i and (x_{Ai}, y_{Ai}) represents the true position of anchor node i. Considering that the magnitude and direction of the localization error between sensor nodes in close proximity are highly correlated, based on this property, we can correct the initial estimated position of the unknown node using the LEV of the anchor node. Assuming the initial estimated position of unknown node k is $\boldsymbol{L_p}(k)=(\hat{x}_k, \hat{y}_k)$ whose nearest anchor node is anchor node i, and the LEV of this anchor node is $\boldsymbol{V}(i)$, then the final estimated position of unknown node k $\boldsymbol{L_F}(k)$ is calculated by equation (24).

$$\boldsymbol{L_F}(k) = \boldsymbol{L_p}(k) + c^{\hat{d}_{ik}} \times \boldsymbol{V}(i) \tag{24}$$

Where c is the error correction factor, which is a constant between [0, 1], and $\hat{d}_{ik}$ is the estimated distance between the unknown node k and its reference anchor node i.

Since the error correction vector of a node is negatively correlated with the distance between two nodes, i.e., The closer the nodes are, the higher the similarity of the error vectors, while the longer distant they are, the lower the similarity in error vectors. Therefore, the magnitude of the correction value is determined by the distance between the unknown node and its reference node; the closer the distance, the larger the correction value, and the farther the distance, the smaller the correction value, in order to avoid over-correction. When the distance between node i and anchor node k is 0, the correction value is equal to $\boldsymbol{V}(i)$. The pseudo-code of our proposed localization error vector correction method is shown below.

```
Algorithm 1: Location error vector correction
Input: Unknown nodes set UN, Node Initial estimation location, estimated distance D, Anchor node set AN.
Output:Final estimation location L_f.
For anchor node i in AN
    Initialize the estimated distance between the anchor node i and itself to 0.
    Estimate the distance between anchor node i and other anchor nodes.
    Calculate the estimated location of anchor node i.
    Calculate anchor node i's location error vector V(i) by equation (23).
End
For node k in UN
    Find the reference anchor k by arg(k) min(d_ik | d_ik ∈ D and i ∈ AN).
    Calculate the final estimate location L_f(k) = L_p(k) + c^(d̂_ik) × V(i).
End
```

4 Experimental results and discussion

4.1 Parameter setting and evaluation metrics

We deploy 200 sensor nodes in a $100m\times 100m$ square area to test the performance of the localization algorithm and compare the proposed MDV-EC with two distance vector localization algorithms, DV-Hop and DV-RND. We also test the effectiveness of the proposed error correction method by comparing the performance of MDV (the proposed algorithm without distance vector correction) with our proposed MDV-EC algorithm. The default parameter settings in the simulation are shown in Table 1.

Table 1 Default parameter settings

Parameters	Value
Deployment area	$100m\times100m$
Node number (N)	200
Anchor number (N_a)	20
Communication radius (R)	30
DOI	0.01
Error correction factor (c)	0.95

In this paper, we use the average localization error (*ALE*) and average localization accuracy (*ALA*) to evaluate the performance of the localization algorithm. The smaller *ALE* and *ALA* represent the higher accuracy of the localization algorithm, which can be calculated by equation (25), (26) respectively.

$$ALE = \sum_{i \in N, i \notin Na} \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \tag{25}$$

$$ALA = \frac{ALE}{R} \tag{26}$$

Where $(\hat{x}_{Ai}, \hat{x}_{Ai})$ represents the final estimated position of unknown node i, (x_{Ai}, y_{Ai}) represents the real position of unknown node i, and R is the communication radius of the node.

4.2 Performance evaluation

Figures 5 and 6 test the effect of error correction factor c on the average localization error of the proposed MDV-EC for different node densities and different number of anchor nodes respectively. The correction factor is a constant between [0, 1]. When the correction factor is small, the correction value of the node position is smaller, and the smaller correction value has less effect on the node position estimation and the error variation is smaller. In order to distinguish the impact of the correction coefficient c on the average positioning error effectively, the range of the error correction coefficient c is set to [0.75, 1] in Figure 5 and 6.

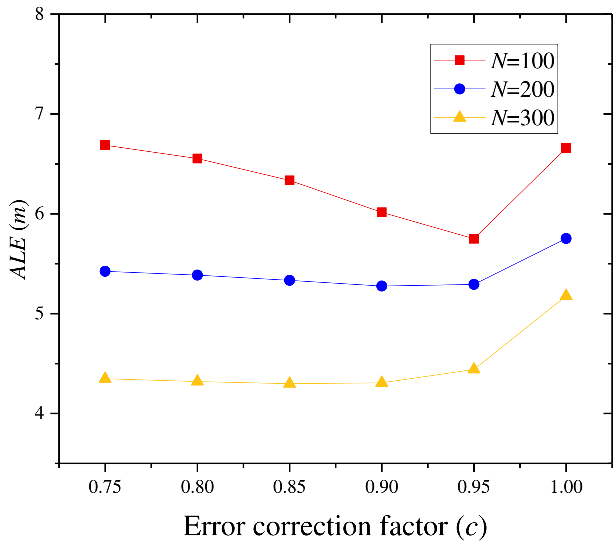


Figure 5 effectiveness of c under different node density

As we can see in figure 5, the *ALE* of the proposed MDV-EC algorithm firstly decreases and then increases as c increases in different node densities. The main reason for this phenomenon is that when c is small, the correction vector value of each unknown node is small, and the correction has less influence on the estimated position. With the continuous increase of c, the value of unknown nodes correction vectors gradually increases and the localization error of the estimated location gradually decreases, which indicates the validity of our proposed error correction. The algorithm achieves optimal localization accuracy when *c* increases to about 0.95, and the *ALE* of the algorithm increases rapidly as *c* continues to increase to 1. This is because when *c* equals to 1, the error correction vector of each unknown node is equal to the *LEV* of nearby anchor node, and using this correction vector to correct the initial location will lead to over-correction, increase the localization error. In different node density scenarios, the higher the node density, the higher the localization accuracy, so according to figure 5 the algorithm has the highest node localization accuracy at *N*=300. When *N* is 100 and 200, the best localization accuracy is achieved at *c*=0.95, and the *ALE* of the MDV-EC is 5.32*m* and 5.21*m*, respectively; when *N*=300, the best localization accuracy is achieved at *c*=0.9, and the *ALE* of the MDV-EC is 4.76*m*. According to the figure, the localization accuracy is more sensitive to the error correction factor in the low node density scenarios. In the case of low node density deployment, the error of the node is larger, so the error vector of the anchor node is also larger, leading to a larger base of the reference correction vector, and therefore the *ALE* is more sensitive to *c*.

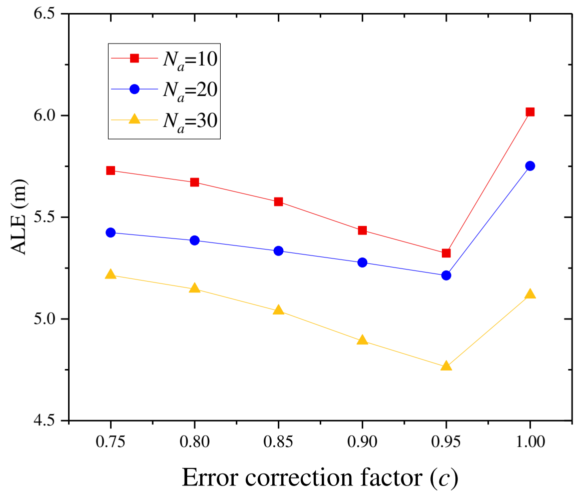


Figure 6 effectiveness of *c* under different anchor number

Figure 6 compares the impact of the error correction factor *c* on *ALE* for different number of anchor node scenarios. According to the figure, in all cases, the *ALE* values gradually decrease as the correction factor *c* increases from 0.75 to 0.95, while the *ALE* values increase as *c* increases from 0.95 to 1. Under the condition of the same correction factor, the more the number of anchor nodes in the network, the higher the localization accuracy. There are two main aspects leading to this phenomenon. On one hand, the more anchor nodes, the more provided localization information, which is beneficial to localization accuracy; on the other hand, the more anchor nodes, the closer the average distance between the unknown node and the reference node, and the higher the reliability of the reference node. In addition, the figure shows that the *ALE* is more sensitive to *c* in high anchor node density scenario. This is mainly because in high anchor node density scenario, the distance between the unknown node and its reference node is shorter, the correction vector of the unknown node is larger, and therefore the value of *ALE* is more sensitive to the change of *c*. The closer the average distance between the unknown node and its reference node, the more reliable the error vector will be.

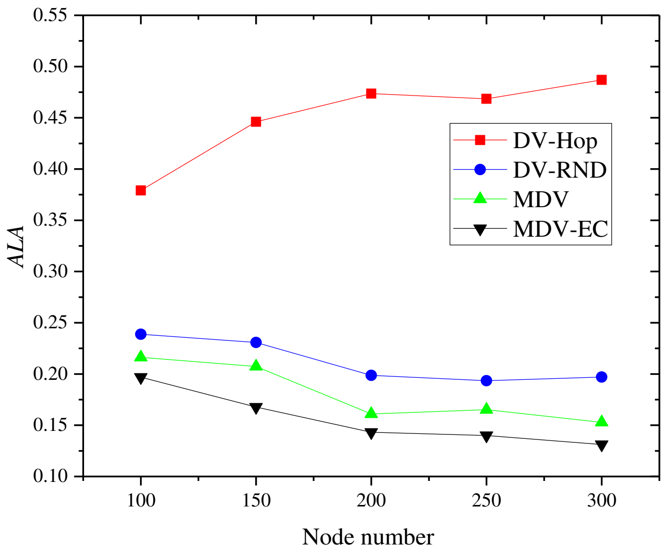


Figure 7 performance comparison in different node density

Figure 7 compares the average localization accuracy of the proposed MDV-EC algorithm with the DV-Hop, DV-RND, and MDV algorithms in different node densities. It can be seen from the figure that the *ALA* value of the DV-Hop algorithm gradually increases with the increase of node density, while the *ALA* of the proposed MDV-EC, DV-RND, and MDV algorithms gradually decreases with the increase of node density. Among all the algorithms, the proposed MDV-EC has the highest localization accuracy, which is better than MDV and DV-RND, where MDV has better localization accuracy than DV-RND. Therefore, the proposed MDV-EC outperformed the distance vector localization algorithms DV-Hop and DV-RND, meanwhile the proposed error correction method can make full use of the similarity of local errors to further reduce the node localization errors and improve the localization accuracy.

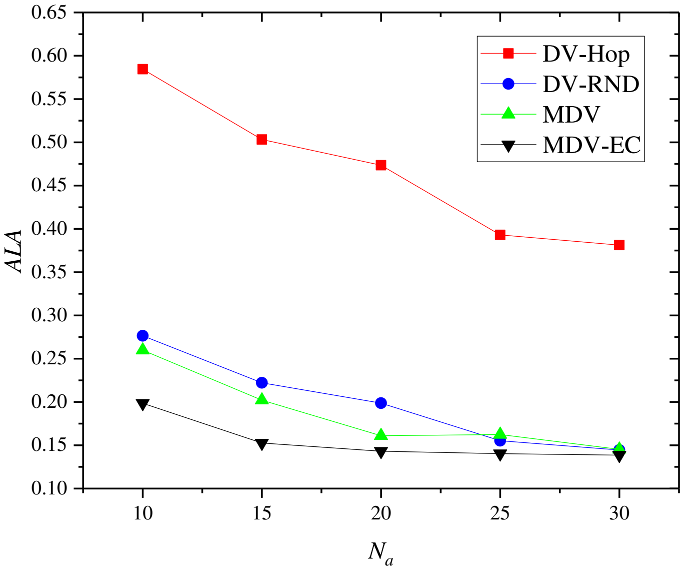


Figure 8 Performance comparison with different number of anchor nodes

Figure 8 compares the localization accuracy of the proposed MDV-EC algorithm with the DV-Hop, DV-RND, and MDV algorithms for scenarios with different numbers of anchor nodes. As can be seen from the figure, the *ALA* values of all four algorithms gradually decrease with the increase of the number of anchor nodes, where the proposed MDV-EC algorithm has the highest localization accuracy and the classical DV-Hop algorithm has the worst localization accuracy. When the number of anchor nodes is less than or equal to 20, the MDV algorithm outperformed DV-RND, while when the number of anchor nodes is greater than 20, the accuracy of DV-RND is superior to the proposed MDV algorithm. Therefore, in practical application scenarios, we can improve the localization accuracy by increasing the number of anchor nodes

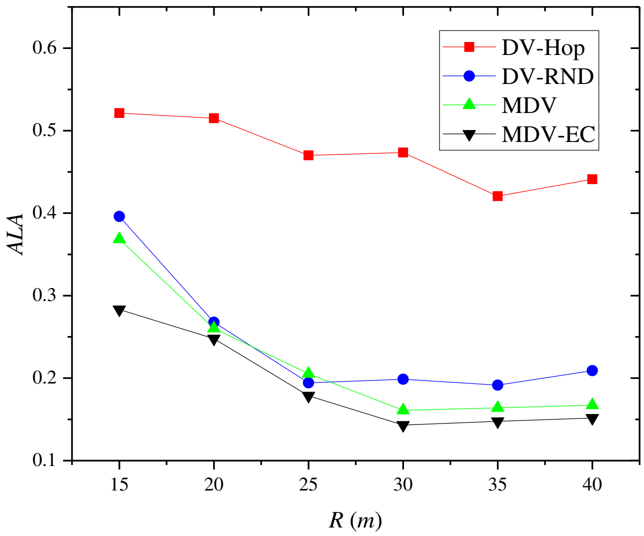


Figure 9 performance comparison in different communication radius

Figure 9 compares the *ALA* of the four algorithms at different communication radius. It can be seen from the figure that the localization accuracy of the proposed MDV-EC algorithm is optimal in all scenarios. The *ALA* of DV-Hop slowly decreases as the communication radius *R* increases, while the *ALA* of DV-RND, MDV and MDV-ECD decrease firstly and then stabilize as the communication radius *R* increases. When *R* is less than 30*m*, the *ALA* of the proposed MDV and MDV-EC decreases with the increase of *R*, and when *R* is greater than 30*m*, the two algorithms have less variation. This is because when the communication radius is small, the connectivity of the network is poor and the average positioning accuracy of all nodes is large. When *R* increases, the connectivity of the network is improved, thus enhancing the accuracy of distance estimation and localization accuracy. As *R* continues to increase, the communication coverage of the sensor nodes in the network becomes too large and the nodes are affected by the deployment area boundaries during distance estimation. While higher network connectivity contributes to the accuracy of distance estimation, larger *R* also makes distance estimation more negatively influenced by boundaries, which overall leads to a flat *ALA*.

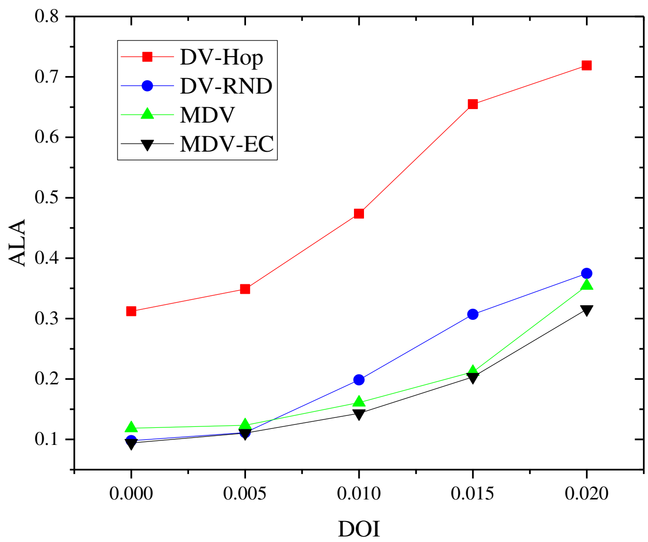


Figure 10 Performance comparison under different signal noise

Figure 10 compares the localization accuracy of the four algorithms in different signal noise environments. As can be seen from the figure, the *ALA* of the four algorithms increases with the increase of signal noise, and the best accuracy in all experimental scenarios is the proposed MDV-EC algorithm, while the worst accuracy is DV-Hop. When the signal noise in the network environment is 0, i.e., the communication coverage area of every node is a standard circle, the *ALA* of the proposed MDV-EC algorithm is slightly better than the DV-RND algorithm, both around 0.1, while the *ALA* of the MDV algorithm without the error correction is about 0.12. With the increase of signal noise in the network, the *ALA* value of DV-RND is gradually larger than MDV, while the *ALA* of the proposed MDV-EC algorithm is always the smallest. Therefore, the proposed MDV-EC has stronger environmental robustness and good localization performance in different noisy environments. In addition, the proposed error correction method can further reduce the node localization errors in different noisy environments.

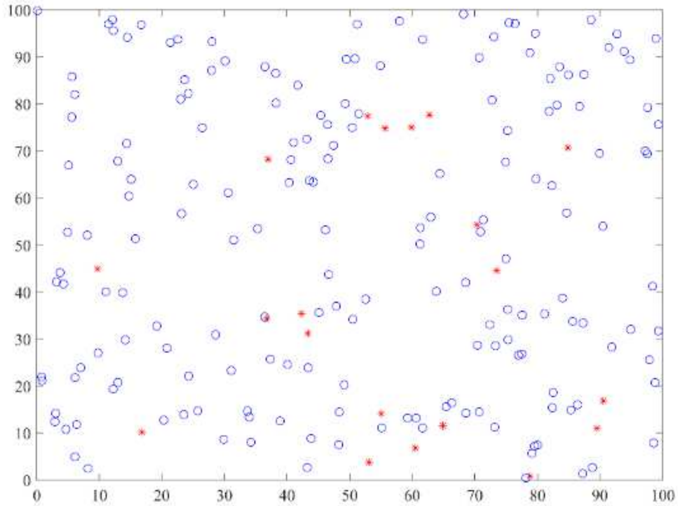


Figure 11 Node distribution

Figure 11 illustrates the random distribution of 200 sensor nodes, including 20 anchor nodes. The blue circles represent unknown nodes and the red stars represent anchor nodes. Set communication radius R=20m, other parameters are preset values. The data of Figure 12 and Figure 13 are obtained in this node deployment of Figure 11.

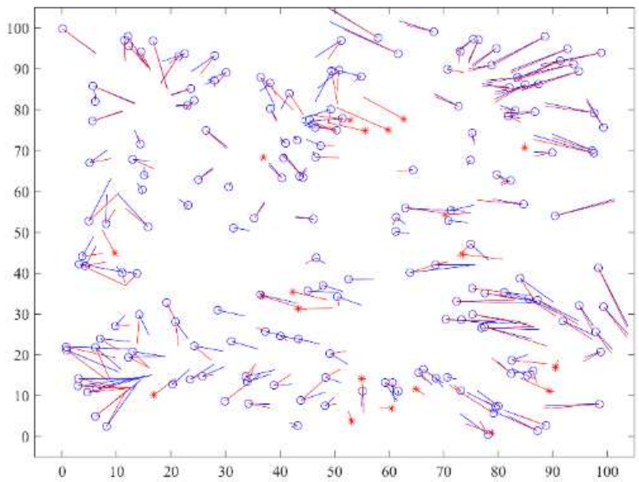


Figure 12 Localization error vector graph

Figure 12 compares the error vectors of the MDV algorithm without the error correction scheme and the MDV-EC algorithm with the error correction scheme. The blue line represents the error vector of MDV, the red line is the error vector of MDV-EC, and the red line segments on the anchor nodes represent the *LEVs* of the respective anchor nodes. It can be seen that there is a strong similarity in the magnitude and direction of the localization error between neighboring nodes. The main reason for this phenomenon is the similar node distribution of neighboring nodes in the distance estimation, which leads to strong similarity in their ranging and localization errors. Furthermore, according to Figure 12, the error vector length of MDV-EC of most nodes (red line segment) is shorter than MDV (blue line segment), which indicates the localization error of the proposed MDV-EC is smaller than MDV, and the proposed error correction method is effective in reducing the localization error of nodes by exploring the local similarity of localization errors.

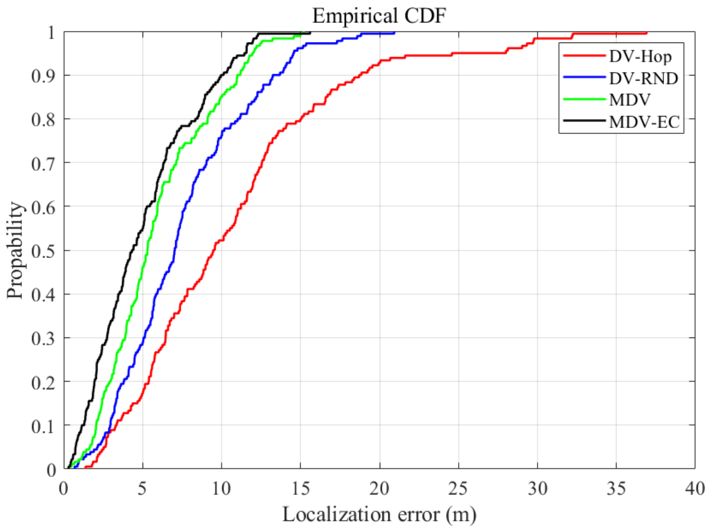


Figure 13 localization error CDF comparison

Figure 13 compares the cumulative probability curves (CDF) of localization errors for the four algorithms. As shown in the figure, the CDF curve of the proposed MDV-EC is always at the highest position, followed by MDV, DV-RND and DV-Hop algorithms. Among them, the MDV-EC algorithm has 90% of the node localization errors within 10*m*, and the maximum value of the localization error is about 15*m*, which is better than the performance of the MDV algorithm. While about 75% of the nodes in DV-RND have errors within 10*m*, and the number of node localization errors within 10m in DV-Hops algorithm accounts for about 52%. Therefore, the proposed MDV-EC algorithm outperforms MDV, DV-RND, and DV-Hops in terms of localization accuracy.

5 Conclusion

In order to enhance the localization accuracy for WSNs, a multi-hop distance-vector localization algorithm named MDV-EC is proposed in this paper. The proposed MDV-EC estimates the distance between neighboring nodes based on the neighbor information and corrects the estimated distance with reference to the local path correction coefficient, rather than corrects the estimated distance from a global point view. By investigating the local similarity characteristics of the localization error in the distance vector localization algorithm, an error correction method is applied to further reduce the localization error. The error correction method uses the estimated and true positions of the anchor nodes to calculate the localization error vector for each anchor node. Then correct the initial estimated position by referring the localization error vector of nearby anchor node. The simulation results demonstrate that the proposed MDV-EC algorithm outperformed DV-Hop and DV-RND in terms of localization accuracy. And the proposed error correction method is effective to reduce the localization error. In the future research, we will focus on exploring the quantization relationship between the local error similarity and the location error vector and investigating a better error correction method to reduce the node localization error.

Acknowledgment

This work was supported by the Key Research and Development Projects of Hunan Province(2018SK2055); Ministry of Emergency Management of the People's Republic of China Key Technologies for Safety Production and Major Accidents Prevention and Control (Hunan-0001-2018AQ); Hunan Provincial Natural Science Foundation of China(2021JJ50093).

Conflict of interest

There is no conflict of interest exists in the submission of this manuscript, and all authors have approved the manuscript that is enclosed.

References

[1] K. Zhang, G. Zhang, X. Yu, S. Hu, Boundary-Based Anchor Selection Method for WSNs Node Localization, Arab. J. Sci. Eng. 46 (2021) 3779–3792. https://doi.org/10.1007/s13369-020-05286-9.

[2] X. Liu, F. Han, W. Ji, Y. Liu, Y. Xie, A Novel Range-Free Localization Scheme Based on Anchor Pairs Condition Decision in Wireless Sensor Networks, IEEE Trans. Commun. 68 (2020) 7882–7895. https://doi.org/10.1109/TCOMM.2020.3020553.

[3] J. Liu, Z. Wang, M. Yao, Z. Qiu, VN-APIT: virtual nodes-based range-free APIT localization scheme for WSN, Wirel. Networks. 22 (2016) 867–878. https://doi.org/10.1007/s11276-015-1007-z.

[4] T.J.S. Chowdhury, C. Elkin, V. Devabhaktuni, D.B. Rawat, J. Oluoch, Advances on localization techniques for wireless sensor networks: A survey, Comput. Networks. 110 (2016) 284–305. https://doi.org/10.1016/j.comnet.2016.10.006.

[5] F. Khelifi, A. Bradai, A. Benslimane, P. Rawat, M. Atri, A Survey of Localization Systems in Internet of Things, Mob. Networks Appl. 24 (2019) 761–785. https://doi.org/10.1007/s11036-018-1090-3.

[6] H. Zhang, Z. Wang, T.A. Gulliver, Two-stage weighted centroid localization for large-scale wireless sensor networks in ambient intelligence environment, J. Ambient Intell. Humaniz. Comput. 9 (2018) 617–627. https://doi.org/10.1007/s12652-017-0458-8.

[7] T. He, C. Huang, B.M. Blum, J.A. Stankovic, T. Abdelzaher, Range-Free Localization Schemes for Large Scale Sensor Networks, Proc. Annu. Int. Conf. Mob. Comput. Networking, MOBICOM. (2003) 81–95. https://doi.org/10.1145/938985.938995.

[8] L.S. Zaremba, W.H. Smoleński, Optimal portfolio choice under a liability constraint, Ann. Oper. Res. 97 (2000) 131–141. https://doi.org/10.1023/A.

[9] F. Shahzad, T.R. Sheltami, E.M. Shakshuki, DV-maxHop: A Fast and Accurate Range-Free Localization Algorithm for Anisotropic Wireless Networks, IEEE Trans. Mob. Comput. 16 (2017) 2494–2505. https://doi.org/10.1109/TMC.2016.2632715.

[10] J. Wang, A. Hou, Y. Tu, H. Yu, An improved DV-Hop localization algorithm based on selected anchors, Comput. Mater. Contin. 65 (2020) 977–991. https://doi.org/10.32604/cmc.2020.011003.

[11] Y. Cao, Z. Wang, Improved DV-Hop Localization Algorithm Based on Dynamic Anchor Node Set for Wireless Sensor Networks, IEEE Access. 7 (2019) 124876–124890. https://doi.org/10.1109/access.2019.2938558.

[12] X. Huang, Multi-node topology location model of smart city based on Internet of Things, Comput. Commun. 152 (2020) 282–295. https://doi.org/10.1016/j.comcom.2020.01.052.

[13] Ouyang, Aijia & Lu, Yinsheng & Liu, Yanmin & Wu, Meng & Peng, Xuyu. (2021). An Improved Adaptive Genetic Algorithm based on DV-Hop for Locating Nodes in Wireless Sensor Networks. Neurocomputing. 458. 10.1016/j.neucom.2020.04.156.

[14] Y. Wang, X. Wang, D. Wang, D.P. Agrawal, Range-free localization using expected hop progress in wireless sensor networks, IEEE Trans. Parallel Distrib. Syst. 20 (2009) 1540–1552. https://doi.org/10.1109/TPDS.2008.239.

[15] [1] G. Wu, S. Wang, B. Wang, Y. Dong, S. Yan, A novel range-free localization based on regulated neighborhood distance for wireless ad hoc and sensor networks, Comput. Networks. 56 (2012) 3581–3593. https://doi.org/10.1016/j.comnet.2012.07.007.

[16] W. Wen, X. Wen, L. Yuan, H. Xu, Range-free localization using expected hop progress in anisotropic wireless sensor networks, Eurasip J. Wirel. Commun. Netw. 2018 (2018). https://doi.org/10.1186/s13638-018-1326-8.

[17] X. Yan, J. Cao, L. Sun, J. Zhou, S. Wang, A. Song, Accurate Analytical-Based Multi-Hop Localization with Low Energy Consumption for Irregular Networks, IEEE Trans. Veh. Technol. 69 (2020) 2021–2033. https://doi.org/10.1109/TVT.2019.2957390.

[18] Zhou, G., He, T., Krishnamurthy, S., & Stankovic, J. A. (2006). Models and solutions for radio irregularity in wireless sensor networks. ACM Transactions on Sensor Networks, 2(2), 221–262. doi:10.1145/1149283.1149287.

[19] S. Zaidi, A. El Assaf, S. Affes, N. Kandil, Accurate Range-Free Localization in Multi-Hop Wireless Sensor Networks, IEEE Trans. Commun. 64 (2016) 3886–3900. https://doi.org/10.1109/TCOMM.2016.2590436.

[20] G. Wu, S. Wang, B. Wang, Y. Dong, S. Yan, A novel range-free localization based on regulated neighborhood distance for wireless ad hoc and sensor networks, Comput. Networks. 56 (2012) 3581–3593. https://doi.org/10.1016/j.comnet.2012.07.007.