

# A Optimized 3D DV-Hop Localization Algorithm Based on Hop Count and Differential Evolution Methods

Kui Li\*, Tengxiao Zhang

College of Electrical & Information Engineering, Southwest Minzu University, Chengdu, 610200, China

---

**Abstract:** The location problem is a fundamental problem in underwater wireless sensor networks. This paper focuses on the distance vector hopping (DV-Hop) localization algorithm, which is most widely used in underwater wireless sensor networks (UWSNs). The algorithm does not require distance measurement and is simple to implement, but the research on DV-Hop algorithm in 3D space is not mature, which leads to large positioning errors. Based on the error analysis of the 3D DV-Hop algorithm, this paper proposes a 3D differential evolutionary hop count DV-Hop (DEHDV-Hop) algorithm to reduce the localization error. We define a continuous hop value based on the number relationship of adjacent nodes and solve the upper bound of the hop count under this definition. The discrete values of the global hop count are converted to continuous values using the broadcasted node information. The concept of trust is introduced by analyzing the error between the actual and estimated distances between anchor nodes. The method of obtaining the average hop distance of unknown nodes in the original algorithm is replaced so that the estimated distance is calculated using the new hop count and hop distance. Finally, the hop count information is segmented, a new fitness function is constructed, and the coordinates of the unknown nodes are solved iteratively using a differential evolutionary algorithm.

**Keywords:** Underwater wireless sensor network, 3D DV-Hop, Trust degree, Differential evolutionary algorithm.

---

## 1. Introduction

At present, countries all over the world are paying more and more attention to the rights and interests of the sea, and the proportion of the marine economy in the global economy is getting higher and higher [1], and each country the enthusiasm of developing and using the ocean is growing year by year, and underwater wireless sensor networks (UWSNs) have become the focus of research [2]. Underwater node localization is a fundamental method and key technology for underwater wireless sensor networks to realize applications in many fields [3]. Obtaining data information about the location of target nodes within a specific water area is crucial and of great research value. The localization technology of single sensor is difficult to achieve precise positioning in the monitoring area, while the underwater wireless sensor network technology that connects multiple sensor nodes to the network precisely achieves information sharing and fusion among nodes, which better improves the accuracy of target positioning. In addition, in underwater wireless sensor networks, the limited computing power and energy of nodes requires the design of node localization algorithms with the performance index of saving communication energy. Therefore, the improvement of underwater node localization technology is beneficial to the overall application research of underwater wireless sensor networks with long-term significance, which can ensure the reliability of localization accuracy and improve the stability of the system, as well as solve the communication problems caused by nodes with low energy and limited computing and communication capabilities.

In recent years, with the development of the three major technologies of sensors, communication and computers, UWSN research based on these technologies has started to rise from two-dimensional space to three-dimensional space. Since it is much more difficult and complex to deploy and control wireless networks in three-dimensional space than in

two-dimensional space, UWSN technology research for three-dimensional space is still a hot topic [4]. In recent years, some swarm intelligence algorithms have been widely used in UWSNs. In order to reduce the localization error, the localization algorithms are combined with swarm intelligence algorithms to implement a variety of localization techniques, and the common swarm intelligence optimization algorithms include genetic algorithm (GA) [5], particle swarm optimization (PSO) [6], and differential evolution algorithm (DE) [7], etc. Gaurav and Ashok [8] proposed a GA-based improved 3D GAIDV-Hop algorithm, by updating the correction factor to modify the average hop distance of anchor nodes, and further optimized the number of hops by the row search algorithm, and defined coplanarity to reduce the effect of coplanar anchor nodes on localization. The proposed localization algorithm is optimized to locate the unknown nodes using GA algorithm. Santar proposed a new PSO [9] based improved PSODV-Hop that reduces the localization error by using a two-dimensional hyperbolic method to calculate the unknown node positions, and finally uses a PSO algorithm to correct the estimated positions. Although a large number of research algorithms already exist to solve the problem of localization, there are still certain drawbacks making it possible to continue to study and optimize the algorithms, which is an ongoing researchable topic and has important research implications.

## 2. DV-Hop Localization Algorithm

In UWSNs, when the nodes are far away from each other, ranging-based localization algorithms are very inaccurate for both angle and signal strength measurements and cause node energy wastage. Therefore, it is of broader market and application value to study ranging-free localization algorithms with high energy utilization, efficiency and accuracy. Among them, DV-Hop algorithm as the most widely used method based on ranging-free algorithm has become a significant research content in UWSN localization

algorithm research.

## 2.1. DV-Hop Algorithm Principle Introduction

The core idea of the 3D DV-Hop algorithm [11] is divided into four main steps, and the detailed process is as follows.

### 2.1.1. Get the Minimum Number of Hops

The anchor node sends a packet containing ID, location and hop count information to the neighbor node, where the initial value of hop count is 0. After the packet is delivered to the neighbor node, it is recorded and the hop count is incremented by 1, and then the packet is forwarded to other neighbor nodes. In the process of broadcasting information, if a node in the network receives the packet with the same ID number several times then it compares with the original recorded data and keeps the minimum number of hops. The above process is repeated so that all nodes in the network can get the minimum hop value.

### 2.1.2. Calculation of the Average Jump Distance of Anchor Nodes

Based on the known anchor node positions, the true distance between anchor nodes can be calculated. The average hop distance of the anchor nodes can be obtained by equation (1) combined with the minimum hop count information. Then the hop distance information of the anchor nodes is redelivered using information broadcast. The average hop distance of unknown nodes is determined by the anchor node closest to it.

$$\text{Hopsiz}e c_i^B = \frac{\left( \sum_{j \neq i} \sqrt{(x_i^B - x_j^B)^2 + (y_i^B - y_j^B)^2 + (z_i^B - z_j^B)^2} \right)}{\sum_{j \neq i} H_{ij}^B} \quad (i, j=1, 2, \dots, m) \quad (1)$$

where the coordinates of anchor node  $X_i^B$  and anchor node  $X_j^B$  are  $(x_i^B, y_i^B, z_i^B)$  and  $(x_j^B, y_j^B, z_j^B)$ ,  $H_{ij}^B$  is the number of hops between  $X_i^B$  and  $X_j^B$ ,  $m$  is the number of anchor nodes, and  $\text{Hopsiz}e c_i^B$  denotes the average hop distance of  $x_i^B$ . As shown in Figure 1,  $X_1^B$ ,  $X_2^B$ ,  $X_3^B$ , represent three anchor nodes,  $X_k^U$  ( $k = 1, 2, \dots, n$ ) are unknown nodes, and  $n$  is the number of unknown nodes. Assuming that  $X_1^B$  to  $X_2^B$  has 50 m away and the number of jumps is 3, and  $X_1^B$  to  $X_3^B$  has 120m away and the number of jumps is 6, the average jump distance of  $X_1^B$  is  $(50 + 120)/(3+6) \approx 18.89\text{m}$ , Similarly, the average jump distance of  $X_2^B$  and  $X_3^B$  can be calculated as 14.375 m and 16.82 m, respectively.

### 2.1.3. Calculation of the Estimated Distance

After step B, the estimated distance  $d_{ik}^{BU}$  between the unknown node  $X_k^U$  and the anchor node  $X_i^B$  is calculated using equation (2).

$$d_{ik}^{BU} = \text{Hopsiz}e c_i^B \times h_{ik}^{BU} \quad (i=1, 2, \dots, m; \quad k=1, 2, \dots, n) \quad (2)$$

where  $h_{ik}^{BU}$ ,  $d_{ik}^{BU}$  denotes the minimum number of hops and the estimated distance between  $X_k^U$  and  $X_i^B$ , respectively.

### 2.1.4. Calculation of the Location of Unknown Nodes

After the calculation by step C, when the unknown node receives not less than 4 anchor nodes information, the location of the unknown node  $X_k^U$  can be solved using the least squares method in three dimensions.

## 2.2. DV-Hop Error Source Analysis

### 2.2.1. Hop Count Error

In the DV-Hop algorithm, if two nodes can communicate directly and a node is within the communication range of another node, the number of hops is denoted as 1 hop. As shown in Fig. 2, let the communication range of nodes are both spheres of radius  $R$ . As can be seen from the figure, the number of hops from  $X_k^U$  to  $X_1^B$  and the hops from  $X_k^U$  to  $X_2^B$  are both 1 hop, and the estimated distances will be equal to the subsequent calculations, however, the actual distances are significantly different. Therefore, the main error of DV-Hop come from the inaccuracy of the hop counts, and the error generated by the hop count affects the calculation of the average hop distance and thus the accuracy of the estimated distance. If the error in hop counts is small enough, the error in estimating the position of unknown nodes will be reduced, so a more accurate method of hop counts calculation and distance estimation needs to be found.

### 2.2.2. Average Jump Distance Error and Least Squares Positioning Error

The error of the average hop distance is mainly affected by the degree of tortuosity of the minimum transmission path between nodes. The greater the degree of tortuosity, the greater the difference between the estimated and actual values of the average hop distance of the anchor nodes, resulting in inaccurate estimation of the distance between the nodes and thus an increase in the localization error. In addition, the average hop distance of unknown nodes in DV-Hop is determined based on the information of 1 anchor node, while the distribution of nodes in UWSN is characterized by a huge number and random unevenness, so more anchor node information should be fully utilized to calculate the average hop distance of unknown nodes. In the final localization process, due to the iterative calculation error of the least squares method itself, coupled with the fact that there is also an error in the estimated distance, further processing will result in the accumulation of errors and large localization errors. Therefore, in this paper, DE is used instead of least squares to calculate the coordinates of the unknown nodes.

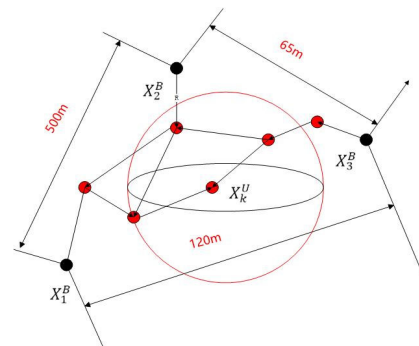


Figure 1. DV-Hop average hop distance calculation chart

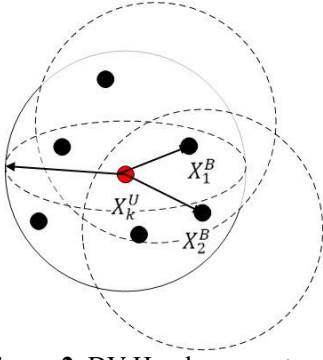


Figure 2. DV-Hop hop count error

### 3. DEHDV-Hop Algorithm

From the analysis of the error sources of DV-Hop algorithm, it is known that the error of 3D DV-Hop mainly comes from three parts. Based on this, this paper proposes the DEHDV-Hop algorithm, which firstly redefines the calculation method of hop counts based on the number of nodes shared among neighboring nodes and the number of neighboring nodes of the nodes themselves, gives the range of hop counts values, and proves it theoretically. Then the concept of trust degree is proposed to redefine the average hop distance from unknown nodes based on the error between the actual and estimated distance about anchor nodes. Finally, the hop counts information is segmented, a new objective function is constructed based on the distance information between nodes, and the DE algorithm is used to optimize the position of the unknown nodes, and then a more accurate position is obtained.

#### 3.1. Optimize the Number of Hops

As shown in Fig 3, the communication radius of nodes in UWSN is  $R$ ,  $X_i$  is any node in the network,  $X_j$  is any neighbor node of  $X_i$ . Let solid dots denote the neighbor nodes shared by two nodes, hollow dots denote the remaining neighbor nodes of two nodes, and  $d_{ij}$  denotes the distance between two nodes. In order to reduce the error caused by the number of hop, this paper defines the continuous value of the number of hop and improves the number of hop by using the ratio of the number of node. Define the continuous hop value of the communication range of neighboring nodes  $X_i$  and  $X_j$ . Assuming that  $N_i$  denotes the number of nodes  $X_i$  and neighboring nodes ( $N_i = |X_i| + 1$ ),  $|X_i|$  denotes the number of neighboring nodes of node  $X_i$ , and  $N_{ij}$  denotes the number of nodes  $X_i$ ,  $X_j$  and their common neighboring nodes, the continuous hop value is expressed as

$$H(i, j) = 1 - \left( \frac{N_{ij}}{N_i} \right)^2 \quad (3)$$

Since the nodes are deployed randomly, the number of hop  $H(i, j)$  from node  $X_i$  to node  $X_j$  calculated in equation (3) may not be equal to  $H(j, i)$ , only if  $N_i = N_j$ ,  $H(i, j) = H(j, i)$ , but this case is a coincidental one. Therefore, the average of  $H(i, j)$  and  $H(j, i)$  is taken as the value of consecutive jump of node  $X_i$  and node  $X_j$ .

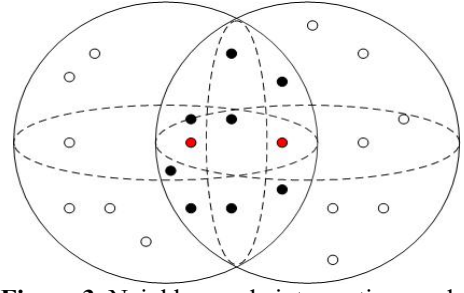


Figure 3. Neighbor node intersection model

From Fig. 3, it can be seen that the number of nodes is related to the distance  $d_{ij}$  between nodes, when the closer the neighboring nodes are, the closer  $N_{ji}$  is to  $N_i$ , the closer  $H_{ij}$  is to 0. Conversely, when  $d_{ij}$  is larger, the difference between  $N_{ji}$  and  $N_i$  is larger and the number of hops between nodes is closer to 1. Therefore, the distance between nodes indirectly affects the number of hops between nodes. In the network, the value range of  $i \in [0, R]$  and the hop count range  $H_{ij} \in [0, 1]$ .

#### 3.2. Correction of Average Jump Distance

To address the problem of error in the average hop distance selection in the traditional DV-Hop algorithm, this section introduces the concept of trust degree to update the unknown nodes so as to obtain the average hop distance value. The real distance between each anchor node in UWSN can be calculated by the Euclidean distance algorithm, and the estimated distance can be obtained from equation (2). The values of the actual distance and the estimated distance are not necessarily identical, and the following is a method to update the unknown nodes according to the error relationship, so as to obtain the average hop distance. Since the average hop distance calculated by each anchor node is different, two different estimated distances are generated between anchor nodes  $X_i^B$  and  $X_j^B$ . In order to balance the error generated by the estimated distance, the estimated distance  $\hat{d}_{ij}^B$  takes the average value of the two distances, as shown in equation (4).

$$\hat{d}_{ij}^B = (\text{Hopsize}c_i^B \times H_{ij}^B + \text{Hopsize}c_j^B \times H_{ij}^B) / 2 \quad (4)$$

The actual distance between the anchor node  $X_i^B$  and the anchor node  $X_j^B$  is found by the Euclidean distance formula and is denoted as  $d_{ij}^B$ . Combining the errors of  $\hat{d}_{ij}^B$  and  $d_{ij}^B$ , the average hop distance error  $e_{ij}^B$  caused by the distance error from  $X_i^B$  to  $X_j^B$  can be calculated according to Equation (5).

$$e_{ij}^B = \frac{|\hat{d}_{ij}^B - d_{ij}^B|}{H_{ij}^B} \quad (5)$$

The hop distance error between  $X_i^B$  and the rest of the anchor nodes is summed up and averaged, which is the average hop distance error  $E_{ij}^B$  of  $X_i^B$ .

$$E_{ij}^B = \sum_{j=0, j \neq i}^m c_{ij}^B / (m - 1) \quad (6)$$

The trust level of each anchor node is the proportion of the inverse of the average hopping distance error of that anchor node to the inverse of the average hopping distance error of all anchor nodes.

$$\theta_i^B = \frac{1/E_i^B}{\sum_{i=1}^m (1/E_i^B)} \quad (7)$$

The average hop distance of the updated unknown node is expressed as equation (8)

$$Hopsizc' = \sum_{i=1}^m \theta_i^B \times Hopsizc_i^B \quad (8)$$

Finally, the average number of hops of the corrected unknown nodes is multiplied by the improved number of hops to obtain the distance between any two nodes in the network, which is the estimated distance  $d_{ij}$ .

### 3.3. Localization Process

The localization process of DV-Hop is calculated using the least squares method, however, the unknown node coordinates obtained by this method have a large error. Therefore, in order to obtain more accurate position coordinates, the DE algorithm is introduced in this paper to solve for the position of the unknown nodes. The DE algorithm [12] is a swarm intelligence optimization algorithm that usually uses stochastic search techniques. It has three main advantages over other swarm intelligence optimization algorithms: fewer parameters, fast convergence, and robustness. The algorithm consists of three processes: mutation, crossover and selection. The mutation process is done by using the differences between the selected individuals in the population, and then the original individuals are crossed with the mutated individuals produced by the mutation process to obtain the candidate individuals. If the candidate individual is better than the original individual in terms of fitness, the candidate individual replaces the original individual and becomes an individual in the new population.

*A. Initialization:* The network environment of the localization area is initialized and set, and the initial population  $X_0 = [X_1^0, X_2^0, \dots, X_{NP}^0]$  is generate according to equation (9), where  $X_i^0 = [X_{i1}^0, X_{i2}^0, \dots, X_{iD}^0]^T$ , NP represents the population size.

$$X_{i,j}^0 = X_{min,j} + rand(0,1) * (X_{max,j} - X_{min,j}), j = 1, 2, \dots, D \quad (9)$$

where  $X_{min,j}$ ,  $X_{max,j}$  indicate the minimum and maximum values of the boundary of the j-th component of the search vector, respectively, and D indicates that each individual. There are D-dimensional vectors.

*B. Mutation:* Each individual in the population can be selectively mutated. Any three individuals  $X_{r1}^t$ ,  $X_{r2}^t$ ,  $X_{r3}^t$  in the population are selected for mutation.

$$V_i^{t+1} = X_{r1}^t + F(X_{r2}^t - X_{r3}^t) \quad (10)$$

where  $V_i^{t+1}$  denotes the population obtained after mutation, t denotes the number of population generations, and F is the scaling factor.

*C. Crossover:* The mutant population is crossed over with the original population according to equation (11).

$$U_{i,j}^{t+1} = \begin{cases} x_{i,j}^t & rand(j) > CR \text{ or } j = rand(i) \\ V_{i,j}^{t+1} & rand(j) \leq CR \text{ or } j \neq rand(i) \end{cases} \quad (11)$$

where  $U_{i,j}^{t+1}$  denotes the population obtained after crossover and CR is the crossover probability.

*C. Selection:* selection of individuals with higher fitness into the next generation according to the principle of elimination in biology.

$$x_i^{t+1} = \begin{cases} x_i^t & f(U_i^{t+1}) \geq f(x_i^t) \\ U_i^{t+1} & f(U_i^{t+1}) < f(x_i^t) \end{cases} \quad (12)$$

where  $f(U_i^{t+1})$  and  $f(x_i^t)$  are the adaptation values for  $U_i^{t+1}$  and  $x_i^t$ , respectively. When the fitness of the test individual  $U_i^{t+1}$  is better than  $x_i^t$ , the test individual replaces the original individual, otherwise the original individual is retained.

In order to reduce the complexity of the fitness function, this paper improves the traditional DE algorithm. By using the hop counts and distance information from the unknown node to the anchor node to optimize the fitness function, the computational effort of the algorithm is effectively reduced and the running time is decreased by improving the fitness function. The original fitness function in the DE algorithm is calculated using the estimated distances from the unknown nodes to all anchor nodes in the network.

$$f(x_k^U, y_k^U, z_k^U) = \frac{1}{m} \sum_{i=1}^m \left( \sqrt{(x_i^B - x_k^U)^2 + (y_i^B - y_k^U)^2 + (z_i^B - z_k^U)^2} - d_{ik}^{BU} \right)^2 \quad (13)$$

where the coordinates of the anchor node  $x_i^B$  and the unknown node  $x_k^U$  are  $(x_i^B, y_i^B, z_i^B)$  and  $(x_k^U, y_k^U, z_k^U)$ .  $d_{ik}^{BU}$  denotes the estimated distance from  $x_i^B$  to  $x_k^U$  and m is the number of anchor nodes.

Although a better localization effect can also be achieved by using the calculation of equation (13), the computational effort of the algorithm will also increase a lot, so a new method of constructing the fitness function is proposed in this paper. Considering that in the range of 1 hop, the number of hops calculated by the method in Eq. (8) is more accurate than DV-Hop, and it will be closer to the real distance in the calculation of the estimated distance. On the contrary, as the number of hops becomes larger, the error accumulation in the estimated distance due to the accumulation of hops will be larger. The adaptation function of this paper differs from that of the original DE algorithm by segmenting the hop count information of unknown nodes and anchor nodes, giving priority to the information of anchor nodes with less than 1 hop to reduce the computation of the DE algorithm, and the adaptation function is:

$$f(x_k^U, y_k^U, z_k^U) = \frac{1}{N_k} \sum_{i=1}^{N_k} \left( \sqrt{(x_i^B - x_k^U)^2 + (y_i^B - y_k^U)^2 + (z_i^B - z_k^U)^2} - d_{ik}^{BU} \right)^2 \quad (14)$$

where  $N_k$  denotes the number of anchor nodes with less than 1 hop for the k-th unknown node.

If the unknown node  $x_k^U$  has no anchor node with less than 1 hop, the information of anchor nodes with less than t (t is

the number of hops from the unknown node to the nearest anchor node,  $t > 1$ , and  $t$  is taken as an integer) hops is used, and  $N_k^p$  is the number of  $p$ -hops nodes of the  $k$ -th unknown node. Since the accumulation of the number of hops can cause excessive distance error, a weight factor is used to correct the distance error accumulated by the excessive number of hops. In this regard, a new objective function is given as follows:

$$\frac{1}{N_k^p} \sum_{i=1}^{N_k^p} \left( \frac{1}{H_{ij}} \right)^2 \left( \sqrt{(x_i^B - x_k^U)^2 + (y_i^B - y_k^U)^2 + (z_i^B - z_k^U)^2} - d_{ik}^{BU} \right)^2 \quad (15)$$

Using the DE algorithm means that the localization problem is transformed into a function minimization optimization problem. Through iteration, the smaller the individual's fitness. The smaller the value of the degree function, the more accurate the localization of the unknown node.

## 4. Experimental Simulation Verification

In this paper, the DEHDV-Hop algorithm is experimentally validated by simulation on MATLAB 2018b, and all nodes are randomly deployed in the corresponding 3D network monitoring area. The node distribution model is shown in Figure 4. The specific parameters of the simulation are shown in Table 1.

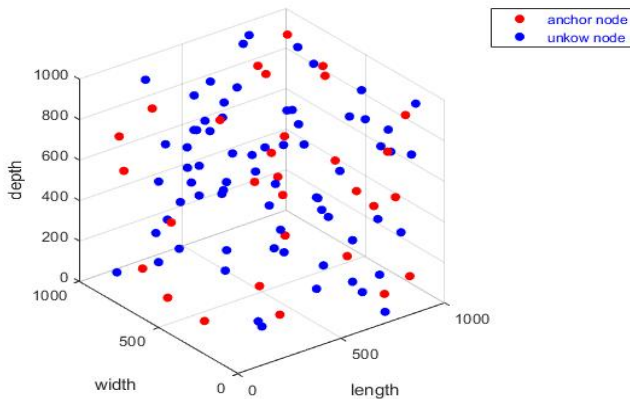


Figure 4. Node Distribution Model

Table 1. Simulation parameter setting

Simulation parameters	Value
Simulation area/ $m^3$	1000*1000*800
Communication range/m	25~50
Total number of nodes	0~200
Anchor node ratio	15%~40%
Number of experiments	100
Crossover probability CR	0.8
Scaling factor F	0.6
Number of iterations	100

The following metrics are used as criteria to evaluate the performance of the algorithm for localization.

A. Average localization error (ALE): It is mainly used to evaluate the localization performance of the algorithm as the total number of nodes, the proportion of anchor nodes, and other different parameters change. The calculation formula is

as follows:

$$ALE = \frac{\sum_{i=1}^n \sqrt{(x_k^U - \hat{x}_k^U)^2 + (y_k^U - \hat{y}_k^U)^2 + (z_k^U - \hat{z}_k^U)^2}}{n} \quad (16)$$

where the estimated and actual coordinates of the unknown node  $x_k^U$  are  $(x_k^U, y_k^U, z_k^U)$ ,  $(\hat{x}_k^U, \hat{y}_k^U, \hat{z}_k^U)$ , and  $n$  is the number of unknown nodes.

B. Normalized Average Positioning Error (Nor-ALE): It mainly evaluates the association between the communication radius  $R$  and the performance of the positioning algorithm as shown in equation (17).

$$Nor_{ALE} = ALE/R \quad (17)$$

C. Localization error percentage (LEP): It mainly evaluates the percentage of errors in locating the nodes, and the nodes that are incorrectly located are those whose distance from the actual location to the estimated location of the unknown node is greater than half of the communication range. The proportion can be expressed as the ratio of the number of nodes incorrectly located ( $n_{wrong}$ ) to the number of unknown nodes ( $n$ ).

$$LEP = n_{wrong}/n \quad (18)$$

We simulate the algorithm under the conditions of varying communication range, anchor node ratio, and total number of nodes, illustrate the results of each improvement step, and compare it with GADV-Hop [13] and PSODV-Hop [14].

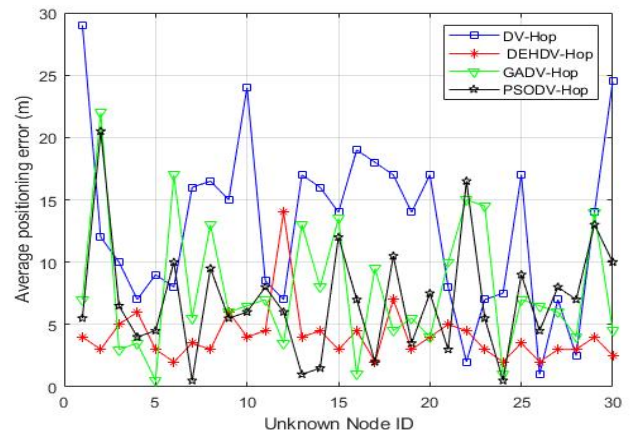
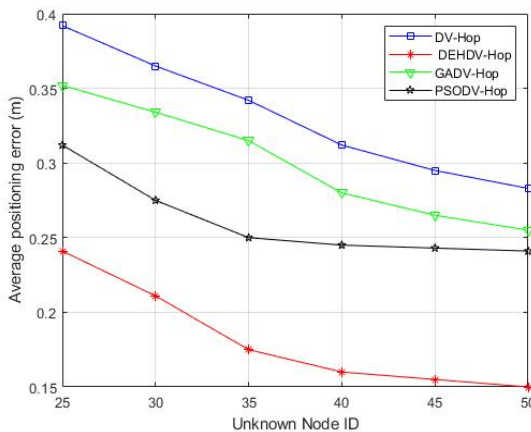


Figure 5. Individual node error plot for DV-Hop vs. comparison algorithm

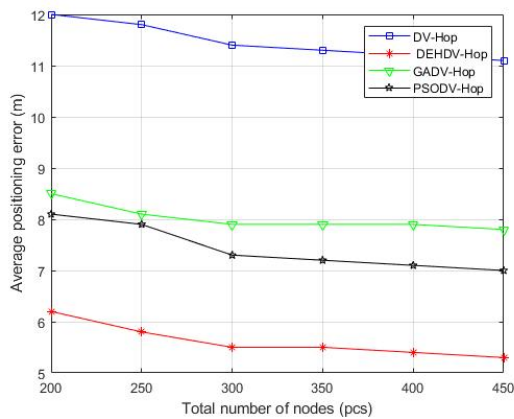
100 nodes are deployed in the UWSN, of which 30% are anchor nodes. As can be seen from Figure 5, the fold of the DEHDV-Hop algorithm is almost entirely below the DV-Hop algorithm. When the node ID is 10, the average localization error produced by the DV-Hop algorithm is 24.26m, while that of DEHDV-Hop is only 4.38m, and the localization error of DEHDV-Hop is reduced by 19.88m. The localization error of DEHDV-Hop algorithm is compared with that of GADV-Hop and PSODV-Hop algorithms, although there is a certain ID node of the average localization error of PSODV-Hop is smaller than that of DEHDV-Hop, but on the whole, the localization error of DEHDV-Hop for a single node is more than 95% smaller than that of the comparison algorithms.

The communication radius of a node is an important

variable that affects the localization error of the algorithm. In order to investigate the effect of the variation of the communication radius on the normalized average localization error, the simulation parameters are set to 100 sensor nodes, the proportion of anchor nodes is 30%, and the communication distance varies from 25m to 50m. Figure 6 shows the variation of normalized average localization error of the four algorithms. All four algorithms have a tendency to decrease the localization error as the communication radius of nodes increases, and it is worth noting that the localization error of DEHDV-Hop has been smaller than that of GADV-Hop and PSODV-Hop, which indicates that the algorithm proposed in this paper has a good correction effect in dealing with the variable factor of communication radius.



**Figure 6.** Effect of communication radius on normalized mean positioning error



**Figure 7.** Effect of total number of nodes on positioning error

In the simulation area, the variation of the total number of nodes has a significant impact on the localization error. Figure 7 shows the trend of the average localization error during the change of the total number of nodes from 200 to 450. The average localization error of all four algorithms decreases gradually as the total number of nodes increases, because as the total number of nodes increases, more and more nodes in the network can communicate with each other, and a large amount of node location information can be collected, which reduces the localization error. In conclusion, the DEHDV-Hop algorithm not only makes the localization error decrease, but also makes the localization error more stable during the change of the total number of nodes.

## 5. Conclusion

The article firstly introduces the principle of the DV-Hop algorithm in 3D space and gives a detailed analysis of the causes of its error. Secondly, the hop values between nodes are improved by redefining the hop count according to the neighbor node ratio. For the error between the actual and estimated distance of anchor nodes, the concept of trust degree is introduced to update the average hop distance between unknown nodes. The DE algorithm is used to construct a new fitness function using distance information so as to optimize the unknown node positions, and a theoretical derivation process is made, and finally the proposed algorithm is validated by simulation. Theoretical and MATLAB simulation experiments show that the DEHDV-Hop algorithm has a smaller localization error compared with the comparison algorithm under various parameter conditions.

## References

- [1] Qu F, Wang S, Wu Z, et al. A survey of ranging algorithms and localization schemes in underwater acoustic sensor network [J]. *Wireless Communication Over Zigbee for Automotive Inclination Measurement China Communications*, 2016, 13 (3): 66-81.
- [2] Heidemann J, Ye W, Wills J. Research challenges and applications for underwater sensor networking [C]. *Wireless Communications and Networking Conference*. Las Vegas: IEEE, 2006: 228-235.
- [3] Pompili D, Akyildiz I F. Overview of networking protocols for underwater wireless communications [J]. *IEEE Communications Magazine*, 2009, 47 (1): 97-102.
- [4] Kumari J, Kumar P, Singh S K. Localization in three-dimensional wireless sensor networks: a survey[J]. *The Journal of Supercomputing*, 2019, 75: 5040-5083.
- [5] Gong D W, Wang G X, Sun X Y, Han Y Y. A set-based genetic algorithm for solving the many-objective optimization problem[J]. *Soft Computing* 2015, 19(6): 1477-1495.
- [6] Souravlias D, Parsopoulos K E, et al. Particle swarm optimization with neighborhood -based budget allocation[J]. *International Journal of Machine Learning and Cybernetics*, 2016, 7(3): 451-477.
- [7] Mohamed A W, Mohamed A K, et al. Adaptive guided differential evolution algorithm with novel mutation for numerical optimization[J]. *International Journal of Machine Learning and Cybernetics*, 2017, 10(2): 253-277.
- [8] Sharma G, Kumar A. Improved range-free localization for three-dimensional wireless sensor networks using genetic algorithm[J]. *Computers & Electrical Engineering*, 2018, 72: 808-827.
- [9] Singh S P, Sharma S C. A PSO used improved localization algorithm for wireless sensor network[J]. *Wireless Personal Communications*, 2018, 98: 487-503.
- [10] Singh P, Khosla A, Kumar A, et al. 3D localization of moving target nodes using single anchor node in anisotropic wireless sensor networks[J]. *AEU- International Journal of Electronics and Communications*, 2017, 82: 543-552.
- [11] Liu Zhiliang, Liu Shuan. Elliptical ranging-based DV-hop localization algorithm for 3D underwater wireless sensor networks [J]. *Computer Engineering and Design*, 2017, 38(12): 3218-3223.
- [12] Cui L, Xu C, Li G, et al. A high accurate localization algorithm with DV-Hop and differential evolution for wireless sensor network[J]. *Applied Soft Computing*, 2018: 39-52.

[13] Peng B, Li L. An improved localization algorithm based on genetic algorithm in wireless sensor networks[J]. *Cognit Neurodyn* 2015, 9(2): 249–256.

[14] Xue D. Research on range-free location algorithm for wireless sensor network based on particle swarm optimization[J]. *EURASIP Journal on Wireless Communications and Networking*, 2019(221): 1-8.