International Journal of Mechanical and Electrical Engineering
ISSN: 3005-9615 (Print), ISSN: 3005-7132 (Online) | Volume 4, Number 3, Year 2024
DOI: https://doi.org/10.62051/ijmee.v4n3.01
Journal homepage: https://wepub.org/index.php/IJMEE/index

# DV-Hop Localization Algorithm for Underwater Wireless Sensor Networks based on Squirrel Algorithm Optimization

Wenkang Zhang, Yuming Mao *, Huizhong Hu

College of Information Science and Electrical Engineering, Shandong Jiaotong University, Jinan, China

* Corresponding Author: Yuming Mao

## ABSTRACT

Aiming at the problem of large errors in the positioning algorithm of underwater wireless sensor networks, an underwater DV-Hop positioning algorithm optimized based on the squirrel algorithm is proposed. The estimated position of the unknown nodes in the DV-Hop algorithm is optimized by the squirrel algorithm, and optimization is carried out with the squirrel algorithm, the number of hops between nodes is optimized using the hop count adjustment factor, the beacon nodes that will lead to a large error are removed by the use of the covariance degree, and the average hopping distance of beacon nodes is optimized using the weighted processing method, and the average value of the improved average hopping distance is taken to be the average hopping distance of each unknown node, to Improve the localization accuracy of DV-Hop algorithm. The simulation results show that the improved algorithm improves the positioning accuracy by 34.02% and 9.75% compared with the traditional 3DDV-Hop and the hopping distance optimized 3DDV-Hop, respectively.

## KEYWORDS

Wireless Sensor Networks; Squirrel Algorithm; DV-Hop Localization Algorithm; Average Hop Distance.

## 1. INTRODUCTION

With the deep development and utilization of marine resources, the intelligent ocean system has become a research focus, and sensing the marine environment is one of the important supporting technologies. Currently, Underwater Wireless Sensor Networks (UWSNs) are an important means to sense the marine environment [1,2], and node information without location is invalid, so the localization technology of underwater nodes is the focus of research in various countries [3,4].

The ocean environment is complex and variable, and the underwater nodes are mainly based on acoustic communication, which is very susceptible to the high noise of the ocean environment and the high delay of the channel. The localization of underwater nodes faces severe challenges, therefore, improving the accuracy of node localization algorithms is a hot spot in current research. Ranging-based and non-ranging-based localization algorithms are two major types of underwater wireless sensor networks [5]. Ranging-based localization algorithms mainly include time-of-arrival (TOA)-based localization algorithms, angle-of-arrival (AOA)-based localization algorithms and signal strength (RSSI)-based localization algorithms, etc [6]. While non-ranging based localization algorithms mainly use the connectivity between nodes to estimate the position, such as center of mass method, convex planning, APIT, DV-HOP, etc [7]. In underwater node localization, DV-Hop is a non-ranging based localization algorithm, which does not need to measure the exact distance between nodes, but only the number of hops and the average hop distance between nodes [8], and therefore, has low hardware requirements and simple and easy to implement, which makes it widely used [9]. In the case of uneven distribution of nodes, the presence of obstacles and signal interference, the traditional 3D DV-Hop algorithm suffers from large localization errors [10]. In order to improve the localization accuracy, some scholars have proposed a variety of improved algorithms based on DV-Hop, but the localization accuracy of these algorithms is still unsatisfactory, and there is a large room for improvement [11-13].

# 2. RELATED THEORIES

## 2.1. DV-Hop Localization Algorithm

Niculescu et al. proposed the DV-Hop localization algorithm [14]. The DV-Hop localization algorithm is applied in 3D space with the following main steps:

1) Calculation of minimum hop count. Within the communication range, each beacon node in the wireless sensor network broadcasts its own information. When a node receives information from a beacon node, it saves the information and adds 1 to the hop value.

2) Hop Distance Estimation. The unknown nodes use the hop count estimation and the location information of the beacon node to estimate the distance between the nodes. By broadcasting the hop count estimation and the location information of the beacon node to the whole network, other nodes receive the information and can use it to calculate the average hop distance between nodes. The calculation is shown in equation (1):

$$HopSize_i = \frac{\sum_{i \neq j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}}{\sum_{i \neq j} h_{ij}} \tag{1}$$

Where the coordinates of the beacon node and the beacon node are ($x_i$, $y_i$, $z_i$) and ($x_j$, $y_j$, $z_j$) respectively, and the minimum hop count between two beacon nodes i, j is $h_{ij}$. all beacon nodes are able to obtain the average hop distance from themselves to other nodes and broadcast this information to the network, and the node, after receiving the information, retains the information from the nearest beacon node, and calculates the beacon node's distance from all unknown nodes distance. $HopSize_i$ is the average hop distance of node i and the distance between $d_{ij}$ unknown nodes and beacon nodes is. The calculation is shown in equation (2):

$$d_{ij} = HopSize_i * h_{ij} \tag{2}$$

3) Unknown node location coordinate calculation. When two or more beacon node distance information is collected by an unknown node, the least squares method is utilized to determine its coordinate position. This method estimates the position of the unknown node by minimizing the error between the known distance information and the predicted distance value to determine the coordinate position of the unknown node.

## 2.2. Error Analysis of DV-Hop Algorithm

There are three main factors that affect the localization error of the DV-Hop algorithm:

1) The hop count problem between nodes. Within the communication range of beacon nodes, neighboring nodes are recorded as one hop regardless of the distance, but in the underwater environment most nodes are randomly distributed, and the complexity of the underwater environment may lead to unstable communication distance between nodes, which in turn affects the accuracy of the hop count.

2) Average hop distance problem. The average distance is calculated by the correction value of the unknown node and the distance between the unknown node and the beacon node. In the underwater

environment, if a beacon node fails, resulting in an error in the average hop distance of that node, it will affect the unknown nodes within the range of that node, which in turn affects the localization accuracy of the DV-Hop algorithm.

3) The problem of node coordinate calculation method. In the case of a common line between nodes, it is relatively easy to use the least squares method to calculate the position of the node with a small error, but the distribution of nodes in the underwater environment is relatively random, and this method will cause the accumulation of errors.

### 2.3. Squirrel Algorithm

Squirrel Search Algorithm (SSA) is derived from the natural dynamic foraging behavior of squirrels and proposed by Mohit Jain et al. 2018. Squirrels are a diverse group of arboreal and nocturnal gliding rodents. The algorithm builds an optimization model by simulating the gliding mechanism of squirrels, which is an algorithm that balances search and convergence [15]. The steps of the algorithm are as follows:

1) Random initialization. Determine the spatial dimension. There are n squirrels in the forest. The position of each squirrel can be specified by a vector, and the positions of all squirrels can be represented by Equation (3):

$$FS = \begin{pmatrix} FS_{1,1} & FS_{1,2} & \cdots & \cdots & FS_{1,d} \\ FS_{2,1} & FS_{2,2} & \cdots & \cdots & FS_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ FS_{n,1} & FS_{n,2} & \cdots & \cdots & FS_{n,d} \end{pmatrix} \tag{3}$$

where $FS_{i,j}$ denotes the jth dimension of the first squirrel. Equation (4) is used to assign the initial position of each squirrel in the forest.

$$FS_{i,j} = FS_L + rand(0,1) * (FS_U - FS_L) \tag{4}$$

$FS_U$ , $FS_L$ represent the upper and lower boundaries in the jth dimension of the ith squirrel, respectively, as uniformly distributed random numbers between [0,1].

2) Adaptation value evaluation. The fitness of each squirrel is calculated by defining the position of the individual squirrel. Store the corresponding values in the following array:

$$f = \begin{pmatrix} f_1([FS_{1,1}, FS_{1,2}, \cdots, \cdots, FS_{1,d}]) \\ f_1([FS_{2,1}, FS_{2,2}, \cdots, \cdots, FS_{2,d}]) \\ \vdots \\ f_1([FS_{n,1}, FS_{n,2}, \cdots, \cdots, FS_{n,d}]) \end{pmatrix} \tag{5}$$

In the fitness function, each squirrel's fitness values reflect the abundance of food resources in its environment, and these fitness values can be viewed as descriptions of the food source situation, categorized into three levels: common trees indicate relatively scarce or undesirable food, oaks indicate moderate food sources, and hickory trees represent the most desirable food source environment. The same is true for their survival probability.

3) Sorting, declaration and random selection. The location adaptation values of each squirrel are stored in an array in an ascending order. The squirrel with the smallest adaptation value is perched in the hickory tree, the next three squirrels with the next lowest adaptation values have chosen the oak tree as a perch and are moving continuously in the direction of the hickory tree, while the remaining squirrels are scattered in the common tree. By random selection, some squirrels had already taken their daily energy requirement and started to move in the direction of the hickory tree. Some of the squirrels that failed to ingest the energy they needed flew toward the oak tree, hoping to gain energy there. Predators affect the foraging behavior of squirrels, and this natural behavior was modeled by

using a position updating mechanism, adjusted by incorporating the probability ($P_{dp}$) of predator presence. The model takes into account the strategic actions that squirrels take when they perceive a potential threat to ensure their survival and safety.

4) Generate new coordinate positions. During the dynamic foraging process of squirrels, if there are no predators, they can glide through the forest easily and efficiently in search of their favorite food. Once a predator is present, squirrels are immediately alert and seek hidden locations only in small areas nearby to avoid potential dangers. Dynamic foraging behavior can lead to the following three scenarios:

(1) A squirrel in an oak tree may choose to move toward a hickory tree while searching for food. Once the movement is complete, the determination of the new location can be calculated using equation (6).

$$FS_{at}^{t+1} = \begin{cases} FS_{at}^{t} + d_g * G_c * (FS_{ht}^{t} - FS_{at}^{t}), R_1 \geq P_{dp} \\ RandomLocation, otherwise \end{cases} \tag{6}$$

where $d_g$ is the random gliding distance, t denotes the current number of iterations, $FS_{ht}^{t}$ is the position of the squirrel in the hickory tree, and $R_1$ denotes a random number generated in the range [0,1]. The balance between the global and local aspects of $G_c$ is achieved through the sliding constants in the mathematical model. In this model, the value of $G_c$ was set to 1.9, a value that was derived by analyzing and justifying a large body of literature.

(2) When squirrels are located in a common tree, they may choose to move toward the oak tree. The new position after moving can be found by equation (7):

$$FS_{nt}^{t+1} = \begin{cases} FS_{nt}^{t} + d_g * G_c * (FS_{at}^{t} - FS_{nt}^{t}), R_2 \geq P_{dp} \\ RandomLocation, otherwise \end{cases} \tag{7}$$

where $R_2$ is a random number that takes values between [0,1] and $FS_{nt}^{t}$ denotes the position of the squirrel in the common tree.

(3) In the face of food shortages, some squirrels that have already ingested acorns in common trees may make the decision to migrate to hickory trees. The goal is to store the pecans. The new location after the move can be found by equation (8):

$$FS_{nt}^{t+1} = \begin{cases} FS_{nt}^{t} + d_g * G_c * (FS_{ht}^{t} - FS_{nt}^{t}), R_3 \geq P_{dp} \\ RandomLocation, otherwise \end{cases} \tag{8}$$

where the probability of a predator appearing is $P_{dp}$, assumed to be $P_{dp}$ = 0.1 in all cases. $R_3$ is a random number in the range [0,1].

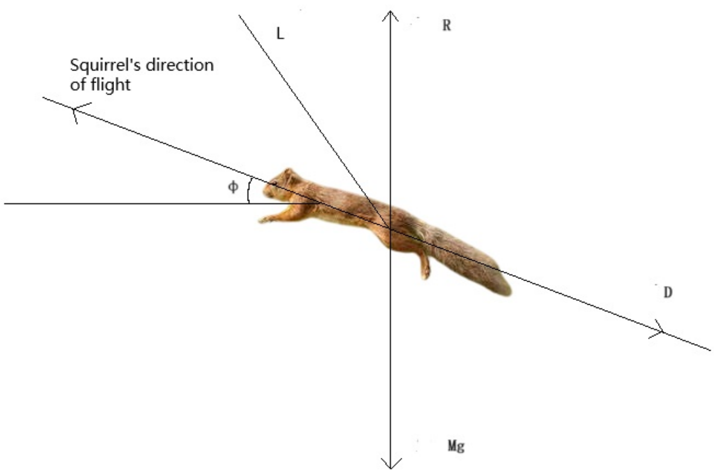


**Figure 1.** Squirrel sliding state

5) Gliding Aerodynamics. Gliding aerodynamics involves the gliding mechanism of the squirrel, in which the sum of the lift force L and the drag force D produces a combined force R, which is equal in magnitude and opposite in direction to the gravitational force exerted on the squirrel, suggesting that the squirrel is subjected to a combined force of zero during gliding, and is in the state of gliding in a straight line at an angle to the horizontal plane at a constant velocity V, allowing it to move efficiently among the trees or in the rest of the environment. The squirrel gliding state is shown in Fig. 1.

The glide ratio is defined as in equation (9):

$$\frac{L}{D}=\frac{1}{tan\phi} \tag{9}$$

During gliding, in order to increase the length of the glide path to improve the glide efficiency, it can be realized by decreasing the glide angle $\phi$. The downward deflection of the air impingement membrane produces the lift force L, which is calculated as follows:

$$L=\frac{1}{2}\rho C_L V^2 S \tag{10}$$

where the gliding speed of the squirrel V = 5.25 m/s, the surface area of the squirrel's wingspan membrane $S=0.0154m^2$, $C_L$ is the coefficient of lift, which is a random number between [0.675,1.5], and the air density (= 1.204 $kg/m^{-3}$). $C_D$ is called the coefficient of frictional drag, which is generally taken as = 0.6. the drag force D, which is the air resistance that the squirrel suffers while gliding, can be obtained from Eq. (11) is obtained:

$$D=\frac{1}{2\rho V^2 S C_D} \tag{11}$$

From Eq. (9), the angle $\phi$ can be calculated from Eq. (12):

$$\phi=\arctan\left(\frac{L}{D}\right) \tag{12}$$

$h_g$ denotes the amount of height reduction that occurs after the squirrel glides, and $d_g$ is the random glide distance, which is calculated according to Equation (13):

$$d_g=\frac{h_g}{tan\phi} \tag{13}$$

Assuming a height reduction of 8 meters, it is necessary to use natural real measurements as parameters to calculate the glide distance $d_g$, including the lift coefficient $C_L$ and the drag coefficient $C_D$, the values of which are based on actual measurements and tests. In a single glide, usually squirrels glide a horizontal distance of 5 meters to 25 meters. In the proposed SSA algorithm model, the horizontal gliding distance is from 9 meters to 20 meters. When the value of $d_g$ is large, it may introduce large perturbations in the equation, which can negatively affect the performance of the algorithm and lead to a decrease in localization accuracy or unstable path planning. Therefore, a parameter called scale factor sf is introduced, sf=18 provides a sufficient range of perturbations such that floating in the interval [0.5,1.11] provides satisfactory performance.

6) Seasonal monitoring conditions. Foraging activities of squirrels are affected by seasonal changes. During the cold season, they forage at high cost. Due to their small size and high body temperature, this requires more energy consumption to maintain body temperature. And squirrels also face the threat of natural predators during foraging, which increases their foraging risk. Squirrels are usually less active in winter climates compared to fall to reduce the costs and risks involved in the foraging process. The steps for modeling seasonal monitoring conditions are as follows:

(1) Calculate the seasonal constant $S_c$. This is obtained by equation (14):

$$S_c^t = \sqrt{\sum_{z=1}^{3}\sum_{k=1}^{d}\left(FS_{at,k}^{t,z} - FS_{ht,k}\right)^2} \tag{14}$$

Where: t is the current number of iterations.

(2) Check the seasonal variation condition $S_c^t < S_{min}$. calculate the minimum value of the seasonal constant, $S_{min}$, from equation (15):

$$S_{min} = \frac{10E^{-6}}{(365)^{\frac{t}{t_m}/2.5}} \tag{15}$$

where the maximum value of the current number of iterations is $t_m$. In heuristic algorithms, the balance between global and local search processes is crucial. larger values of $S_{min}$ algorithms are more inclined to perform global optimization, while smaller values of $S_{min}$ algorithms are more inclined to perform local optimization. Therefore, when choosing the parameter $S_{min}$, a careful balance between the global and local search processes is needed to ensure that the algorithm can obtain the best solution in a reasonable amount of time.

If the seasonal change conditions are met, signaling the end of winter, surviving squirrels can feel a significant reduction in foraging costs, and as the weather warms, food resources become more abundantly available. They will forage in new directions, and the squirrel's random glide equation is as follows:

$$FS_{nt}^{new} = FS_L + Lévy\left(FS_{i,U} - FS_{i,L}\right) \tag{16}$$

$Lévy$ denotes the Levy distribution, which helps the algorithm to explore a better and more efficient search space, jumping out of the local optimal solution and improving the global optimization seeking ability of the algorithm. The calculation is shown in equation (17):

$$Levy = 0.01\frac{r_a * \sigma}{|r_b|^{\frac{1}{\beta}}} \tag{17}$$

β is a constant, here taking the value of 1.5, $r_a$ and $r_b$ are two normally distributed random numbers on the interval [0,1], and σ is computed as shown in Eq. (18):.

$$\sigma = \left(\frac{\Gamma(1+\beta) * \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) * \beta * 2^{\left(\frac{\beta-1}{2}\right)}}\right)^{\frac{1}{\beta}} \tag{18}$$

Included among these, $\Gamma(x) = (x-1)!$

7) Checking algorithm stopping conditions. The squirrel algorithm checks to see if a stopping condition is satisfied. This condition is usually based on the maximum number of iterations $t_m$. In each loop, the algorithm increases the value of the current number of iterations t. When the value of t equals $t_m$, the algorithm stops iterating, outputs the optimal solution, and ends the run.

# 3. DV-HOP ALGORITHM OPTIMIZED BASED ON SQUIRREL ALGORITHM

## 3.1. Hop Count Optimization

In order to solve the influence of the hop count problem between neighboring nodes on the localization accuracy, the hop count adjustment coefficient is introduced to correct the original hop count obtained by the nodes. The coefficient consists of an exact hop count and a deviation coefficient.

$$H_{ij} = \frac{D_{ij}}{R} \tag{19}$$

Where $D_{ij}$ is the actual distance between the beacon nodes and the communication radius is R. The deviation factor is calculated as shown in equation (20):

$$M_{ij} = \frac{|H_{ij}-h_{ij}|}{|h_{ij}|} \tag{20}$$

The deviation coefficient is an important indicator of the difference between the estimated hop value in the first step of the DV-Hop algorithm and the exact hop value between nodes. The larger the value of the deviation coefficient, the larger the error between the estimated hop value and the actual hop value between the nodes, in order to reduce the error, the hop information needs to be corrected, which can be obtained from equation (21):

$$c_{ij} = 1 - M_{ij}^2 \tag{21}$$

The corrected inter-node hopping value $\overline{h}_{ij}$ can be obtained from Eq. (22):

$$\overline{h}_{ij} = c_{ij}h_{ij} \tag{22}$$

### 3.2. Average Hopping Distance Optimization

In order to reduce the impact of the average hop distance between nodes on the localization accuracy, the beacon nodes with large localization errors are excluded, the beacon nodes that will lead to large errors are removed using the covariance degree, the nodes that are not reliable enough or are in a special position are excluded, and then the average hop distance of the beacon nodes is optimized using a weighted processing method, and the average value of the improved average hop distance is taken as the average hop distance of each unknown node, and the maximum ideal for node i hop count is shown in Equation (23):

$$S_i = \max\left(\frac{P_{k1}}{R}, \frac{P_{k2}}{R}, \frac{P_{k2}}{R}, \frac{P_{k4}}{R}\right) \tag{23}$$

where the distances from the four vertices of the rectangular distribution range to node k are respectively: $P_{k1}$,$P_{k2}$,$P_{k3}$,$P_{k4}$. The localization algorithm may have a large error when the number of hops between beacon nodes is too large. Therefore, the average hop distance of a node is calculated by involving another beacon node as well. The difference between the actual minimum number of hops and the ideal number of hops between two nodes is found by Equation (24):

$$\delta_{ij} = \left|\frac{D_{ij}}{R} - h_{ij}\right| \tag{24}$$

The minimum number of hops between two beacon nodes i and j in the above equation is $h_{ij}$. $W_{ij}$ denotes the weight of node j to participate in the computation of the average hop distance of node i, which is calculated as follows:

$$W_{ij} = \frac{1/\delta_{ij}}{\sum_{l\neq i}^{m} 1/\delta_{ij}} \tag{25}$$

In the above equation, m denotes the total number of nodes involved in calculating the average hop distance of node i. The larger the difference $\delta_{ij}$ between the ideal number of hops and the actual minimum number of hops between two nodes, the smaller $W_{ij}$, thus reducing the impact of the nodes with excessive errors on the localization results.Q is the improved average hop distance, which is calculated by the following formula:

$$Q = \sum_{j\neq i}^{m} W_{ij} \times \frac{D_{ij}}{h_{ij}} \tag{26}$$

### 3.3. Unknown Node Coordinate Position Optimization

In the traditional DV-Hop localization algorithm, the least squares method causes error accumulation and affects the localization accuracy when the node dispersion is more random and the spacing calculation error is large in the real situation. To solve this problem, squirrel algorithm is used instead of least squares.

The DV-Hop localization algorithm for underwater wireless sensor networks optimized based on the squirrel algorithm can be divided into the following steps:

1) Network initialization. Establish network topology and determine node locations, and construct fusion coefficients for fusing ranging information from multiple nodes to improve localization accuracy.

2) Squirrel population initialization, the algorithm generates an initial squirrel population for each node, designs squirrel fitness values, and evaluates the fitness of the nodes based on their location and ranging information. Subsequently, the adaptation values of the squirrels are sorted, and the optimal node is selected as the target, which simulates the gliding mechanism of the squirrels. The seasonal monitoring condition is introduced to prevent the algorithm from falling into local optimality and further improve the quality of the solution.

3) Stopping condition of the algorithm. Determine whether the algorithm has reached a predetermined number of iterations or error range, and complete the localization task and output the results through the termination of the algorithm. Squirrel foraging strategy. Monitor the conditions according to the acclimatization value season to adapt to the changes in the environment.

4) DV-Hop Localization. According to the optimal solution in the iterative process or the final converged solution, the optimized node position is obtained as input to execute the DV-Hop localization algorithm. The algorithm estimates the absolute position of the node by the number of hops and hop distances between the nodes.

## 4. EXPERIMENTS AND SIMULATIONS

Simulation experiments were conducted using MATLAB R2022b software to verify the performance of the DV-Hop positioning algorithm for optimizing underwater wireless sensor networks based on the squirrel algorithm. The experimental area is set as a square area of 100m×100m×100m. The positioning accuracies of the traditional 3DDV-Hop algorithm, the hopping distance optimized 3DDV-Hop algorithm and the ASSA-3DDV-Hop algorithm are compared respectively. The normalized localization error of the algorithm is:

$$ALE = \frac{\sum_{i=n}^{n}\sqrt{(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2+(z_i-\hat{z}_i)^2}}{N\times R} \tag{27}$$

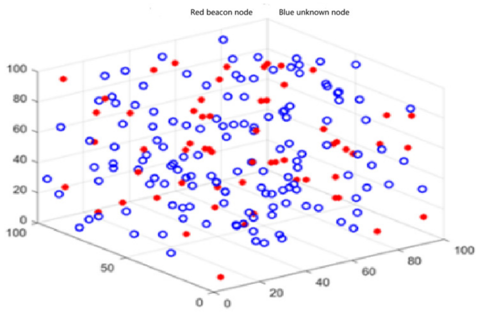


**Figure 2.** Underwater random node distribution scenario

Where R is the communication radius of the node, N is the number of unknown nodes. $(x_i, y_i, z_i)$ are the coordinates of the unknown nodes in the localization algorithm, and $(\hat{x}_i, \hat{y}_i, \hat{z}_i)$ are the actual coordinates of the unknown nodes. Fig. 2 shows the simulated distribution of underwater wireless sensor nodes.

## 4.1. Algorithm Parameterization

In the experimental network environment, Table 1 shows the wireless sensor node parameter settings and Table 2 shows the ASSA algorithm parameter settings. The parameters in the original paper are maintained in the comparison with other algorithms.

**Table 1.** WSN parameter settings

| Parameter | Value |
|---|---|
| **Boundary deployment size/m$^3$** | 100×100×100 |
| **Total number of nodes(N)** | 150,200,250,300,350,400 |
| **Communication range (R)/m** | 45,50,55,60,65,70,75,80 |
| **Percentage of beacon nodes (p)** | 20,30,40,50,60,70,80,90 |

**Table 2.** ASSA parameter setting

| Parameter | Value |
|---|---|
| **Number of squirrels/pc** | 50 |
| **Hickory tree/tree** | 1 |
| **Oak tree/pc** | 3 |
| **Common tree/tree** | 46 |
| **Probability of predator occurrence($P_{dp}$)** | 0.1 |
| **Slide coefficient($G_c$)** | 1.9 |
| **Air density($\rho$)/kgm$^{-3}$** | 1.204 |
| **Velocity(V)/ms$^{-1}$** | 5.25 |

## 4.2. Simulation Analysis

### 4.2.1. Analysis of Beacon Node Proportion and Mean Localization Error.

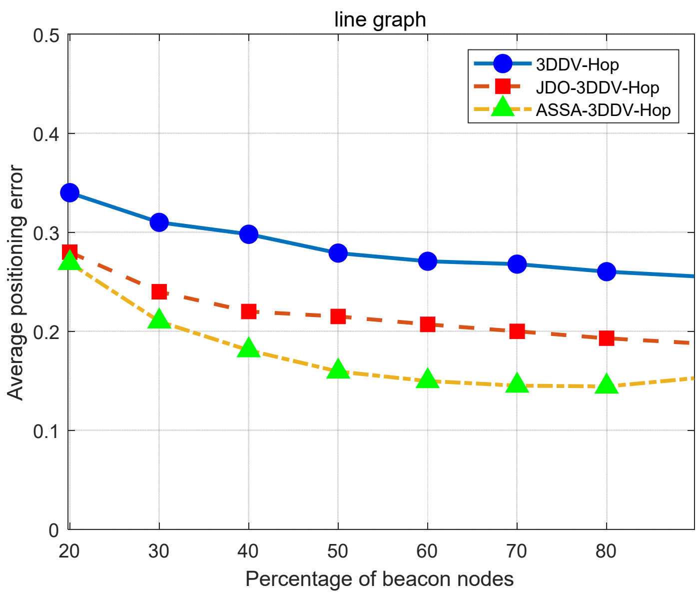


**Figure 3.** Effect of beacon node share on average localization error

In the set simulation area, the communication radius of the nodes is set to R=60m, the total number of nodes is set to 200, and the percentage of beacon nodes is set to increase from 20% to 90%. The nodes are randomly distributed in the monitored area, and the traditional 3DDV-Hop algorithm, JDO-3DDV-Hop algorithm and ASSA-3DDV-Hop algorithm are compared respectively. The experiment is shown in Fig. 3.

According to the above figure, the average localization error of the algorithm shows a gradual decrease in the case of an increase in the number of beacon nodes. In the 3D DV-Hop algorithm, the increase of beacon nodes leads to a decrease in the number of hops between nodes, which affects the localization results. With the participation of more beacon nodes, the number of hops between nodes decreases and the localization accuracy of the algorithm increases. Under the condition of the same number of beacon nodes, the ASSA-3DDV-Hop algorithm has better localization accuracy compared to the traditional 3DDV-Hop algorithm and JDO-3DDV-Hop algorithm. Under the condition of different proportions of beacon nodes, the average localization error of ASSA-3DDV-Hop algorithm is reduced by 32.14% and 11.02% compared with the traditional 3DDV-Hop algorithm and JDO-3DDV-Hop algorithm, respectively.

### 4.2.2. Beacon Node Communication Range and Average Localization Error Analysis.

In the set simulation area, the number of beacon nodes is fixed to 60, the node communication range R is set to increase from 45m to 80m, and 200 nodes are randomly distributed in the monitored area. The experiment is shown in Fig. 4.

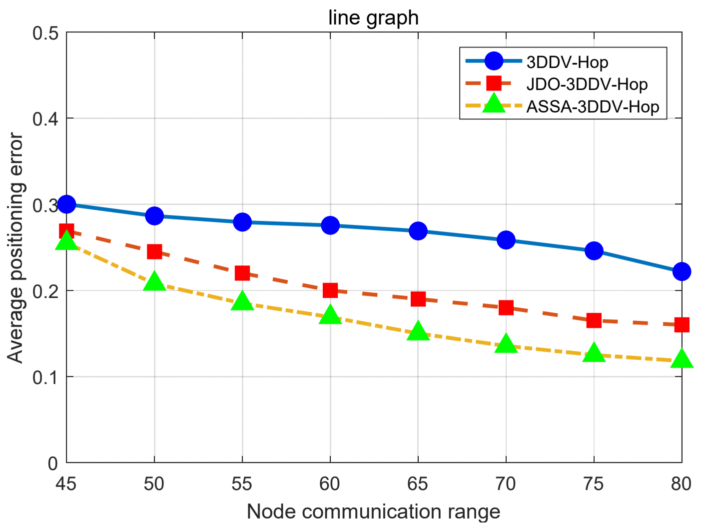


**Figure 4.** Effect of communication range of beacon nodes on average localization error

The highest localization accuracy of the ASSA-3DDV-Hop algorithm is seen when the communication range R takes a fixed value, and the average localization error of the algorithm decreases gradually as the communication range R increases. The nodes have to invest more energy to maintain the communication connection, which is a serious challenge for sensor nodes relying on battery power. Especially in the case of underwater monitoring where the environment is harsh and cannot be recharged in time, the node's endurance will be seriously affected. Reducing the localization error by sacrificing the communication energy may not pay off. Even if the communication energy is sacrificed, the improvement in positioning accuracy may not be significant, while the range of the node is seriously compromised. Under different communication range conditions, the ASSA-3DDV-Hop algorithm improves the localization accuracy by 31.04% and 7.42% compared to the traditional 3DDV-Hop algorithm and JDO-3DDV-Hop algorithm, respectively.

### 4.2.3. Number of Nodes and Average Localization Error Analysis.

In the set simulation area, the communication range R is set to 60m, the proportion of beacon nodes is set to 20%, and 150, 200, 250, 300, 350, and 400 nodes are randomly distributed in the monitored area, respectively, and the experiment is shown in Fig. 5.

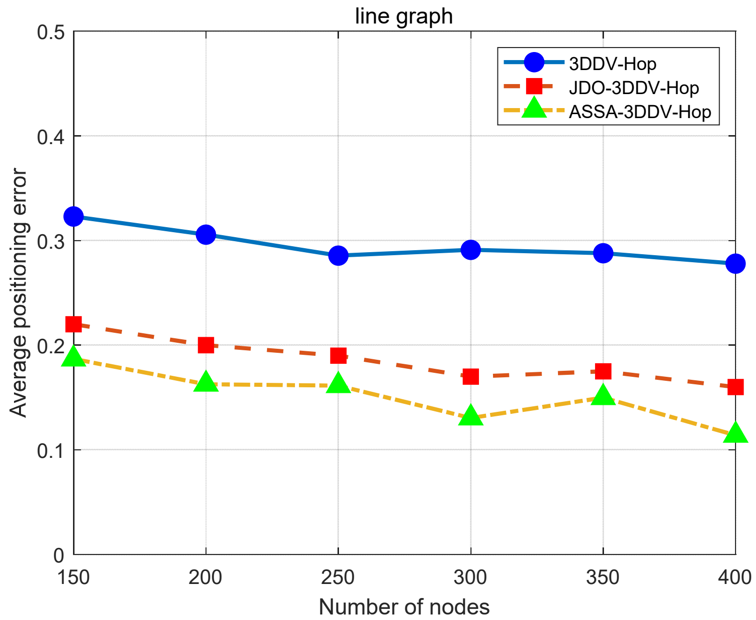


**Figure 5.** Effect of the number of nodes on the average localization error

When the number of nodes takes a certain fixed value, the average localization error of the ASSA-3DDV-Hop algorithm is smaller than that of the traditional 3DDV-Hop algorithm and the JDO-3DDV-Hop algorithm. Therefore, the ASSA-3DDV-Hop algorithm is able to achieve higher localization accuracy with the same number of nodes. With different number of nodes, the ASSA-3DDV-Hop algorithm improves the localization accuracy by 38.89% and 10.82% relative to the traditional 3DDV-Hop algorithm and the JDO-3DDV-Hop algorithm, respectively.

## 5. CONCLUDE

The underwater DV-Hop algorithm suffers from low localization accuracy and poor stability due to the problems of underwater DV-Hop algorithm. In order to solve these problems, an underwater DV-Hop localization algorithm optimized by squirrel algorithm is proposed. The algorithm firstly uses the hop adjustment factor to optimize the number of hops between nodes, which can adjust the distance between nodes more finely, secondly, it uses the covariance degree to remove beacon nodes that will lead to larger errors, and excludes the nodes that are not reliable enough or are in a special position, and then it uses the weighted processing to optimize the average hopping distance of the beacon nodes, and the average value of the improved average hopping distance is taken as the average hopping distance of each unknown node, to The average of the improved average hopping distance is taken as the average hopping distance of each unknown node, which reduces the localization error due to the average hopping distance problem; secondly, the squirrel algorithm is introduced, which further improves the algorithm's optimization efficiency and localization accuracy. The experimental results show that the ASSA-3DDV-Hop algorithm outperforms other algorithms with better accuracy and higher stability when the number of deployed sensor nodes, the communication range and the number of beacon nodes are changed.

# REFERENCES


[1] XU Chen, WANG Biao. A key extraction method for underwater wireless sensor networks[J]. Computer Simulation, 2023, 40(2):415-419.

[2] ZHANG Chi, ZHANG Baoheng, ZHANG Jing,et al. A review of localization techniques for underwater wireless sensor networks[J]. Computer Applications,2022,42(z1):226-235.

[3] Qiang Zhang, Qing Zhang, Lei Zhang, et al. Path planning for mobile beacons in node localization for wireless sensor networks[J]. Journal of Weapons and Equipment Engineering,2016,37(2):144-149.

[4] WAQAS KHAN M, KHAN M, Hafeez A. Localization of wireless nodes with partial connectivity in wireless sensor systems[J]. IET Wireless Sensor Systems, 2020, 10(6): 299-307.

[5] Wang Guowu. Research and improvement of DV-Hop localization algorithm for wireless sensor networks[D]. Guangxi: Guangxi Normal University,2021.

[6] WANG Lei, LIU Jingjing, QI Junyan, et al. DV-HOP positioning algorithm based on range correction and improved whale optimization[J]. Instrumentation Technology and Sensors,2022(2):116-121,126.

[7] SONG Xiaodong, SUN Lijun, CHEN Tianfei. Performance analysis and comparison of DV-Hop optimization algorithm[J]. Journal of Electronic Measurement and Instrumentation,2019,33(5):125-133.

[8] Song Yang. An iterative method for handling node flip ambiguity in node localization for wireless sensor networks [D]. Shandong: Shandong University,2014.

[9] Huei-Chao Yan, Zeng Zi-Wei, Gang Wang. An improved DV-Hop algorithm integrating hop distance correction and sparrow search[J]. Electronic Measurement Technology,2021,44(21):133-138.

[10] PAN Zhiyuan, BU Fanliang. DV-Hop positioning algorithm based on dung beetle algorithm optimization[J]. Journal of Electronic Measurement and Instrumentation,2023,37(7):33-41.

[11] Wang Guowu, Chen Yuanyan. Optimization of DV-Hop localization algorithm based on hop count correction and genetic simulated annealing[J]. Computer Science,2021,48(z1):313-316.

[12] Shi Yunong, Liu Hailong. Research on three-dimensional DV-Hop positioning algorithm based on hop count and hop distance optimization[J]. Journal of Sensing Technology,2022,35(8):1080-1085.

[13] ZHANG Jing, GUO Yi-Han. Levy flight strategy combined with 3D gray wolf optimization for DV-hop positioning algorithm[J]. Small Microcomputer Systems,2022,43(7):1518-1522.

[14] Liu Chuanzhou. Research on wireless sensor network localization algorithm based on non-ranging [D]. Jiangsu: Nanjing University of Posts and Telecommunications,2022.

[15] Nie Jinghua. Research on DV-Hop localization algorithm based on squirrel algorithm[D]. Guangxi: Guilin University of Technology,2023.