Tabriz Journal of Electrical Engineering (TJEE), vol. 55, no. 1, 2025 Serial no. 111
DOI: 10.22034/tjee.2025.63880.4902

Smart Objects Localization by Improved IDVHOP Algorithm and the WOA Optimization Algorithm

Enciye Farmanbar, Nazbanoo Farzaneh*

Engineering Department, Imam Reza International University, Mashhad, Iran.
nc.fadafan@gmail.com,Nazbanou.farzaneh@imamreza.ac.ir
*Corresponding author

Received: 08/10/2024, Revised:09/12/2024, Accepted:15/01/2025.

Abstract
Sensors of the Internet of Things and wireless sensor networks need accurate localization to provide information, such as road information. Installing a locator on all the nodes and sensors is very expensive, and, for this reason, indirect localization is done. One of the low-cost localization methods is DVHop algorithm. Due to the simplicity of DVHop algorithm, its execution time is not long; for this reason, it does not impose much energy consumption, and it is considered a low-cost algorithm. One of the challenges of the DVHop method is its significant error in localization. To solve this problem, in the present paper, a smart locator system is presented using the DVHop algorithm and improved Whale optimization algorithm to estimate the location of objects and sensors. The proposed method has three main steps for smart localization. Experiments showed that the proposed method has less localization error than PSO, WOA, GWO, and HHO algorithms and has high stability due to less standard deviation in localizing the error. Compared to the DVHop, PSODVHop, GSODVHop, and DEIDVHop, the proposed method reduces errors by 1.73, 1.60, 1.28, and 1.13 times, respectively.

Keywords
Internet of Things, DVHop algorithm, Localization, Whale optimization algorithm.

1. Introduction

Wireless sensor networks (WSNs) can be considered as network or a part of IoT because WSN is in the IoT perception or sensor layer [1, 2].

One of the prominent tasks of each sensor node is collecting environmental information with a specified location. Without localization, the collected information is not much worthy, therefore, each node must be capable of localization by GPS information [3]. Without localization, many WSN protocols are not sufficiently efficient; moreover, Routing Protocol faces challenges and errors [4-8].

Various methods have been proposed for localization in WSN based on using GPS. The first challenge is the high cost of installing a locator on all nodes. The second challenge is that, in this situation, the energy consumption increases in the network and the network lifetime decreases [9].

To overcome these challenges, indirect localization has attracted attention from various studies. In these methods, only some of the nodes are anchor nodes; others are not. The non-anchor nodes can perform localization by network information and the anchor nodes [10]. The indirect localization methods enjoy numerous advantages, such as low-cost implementation, less energy consumption, etc.

For indirect localization in the sensor network, different methods have been proposed, such as neural network-based methods [11], deep learning [12], the methods based on hop count algorithm (such as DVHop) [13], the methods based on time-difference algorithm [14], and the methods based on trigonometry [15]. To decrease the localization errors in these methods, typically, the optimization-based approaches and meta-heuristic algorithms are utilized [16]. For instance, algorithms such as genetic algorithm [17] and particle-optimization algorithm [4] are used in these networks to decrease the localization error.

One of the well-known algorithms for localization is DV-Hop algorithm. DV-Hop algorithm is a range free localization algorithm in WSN. This algorithm performs localization based on distance. But localization error is one of the main problems in DV-Hop. Many methods have been proposed to solve this problem, and one of them is the use of metaheuristic methods to reduce the localization error [18]. The most important problem with this group of methods is their long running time.

In this paper, an indirect localization method is presented which is based on the DVHop method for two-dimensional environments of agricultural, rescue, and commercial applications. In the proposed method, DVHOP algorithm and Improved Whale optimization algorithm (IWOA) [19] are used to minimize the localization error.

The DVHop algorithm has many advantages due to its simplicity, but it cannot perform localization with high accuracy. For this reason, the proposed method uses the Whale Optimization algorithm to increase the localization speed and accuracy and reduce the localization error of the DVHop algorithm. An important advantage of WOA is the use of simple but powerful search mechanisms to find the optimal solution at high speed compared to many metaheuristic algorithms such as PSO,SOA,BOA, [20, 21] The following presents the different sections of the present paper: Section 2 has investigated the solutions available in localization methods. In Section 3, the proposed method is described in detail. Section 4 has depicted the simulation results; finally, Section 5 has offered the conclusions.

2. Related Works

In this section, a number of articles that are based on indirect location methods and use optimization methods for location are reviewed.

In [22], the authors investigated the application of group-optimization methods for localization in new networks while focusing on energy consumption decrease. Totally, ten different meta-heuristic algorithms for localization have been analyzed. The obtained results show the high potential of some crowd-based optimization algorithms; based on the results, these methods can localize the objects in the network with a short delay.

In [23], an underwater localization method has utilized the measurement of TDOA and FDOA based on IWO optimization algorithm. For many applications in sonar systems and underwater sensor networks, localizing the underwater source is an essential challenge receiving massive attention in recent decades.

This method has higher localization accuracy than CRLB (Cramer-Rao lower bound) methods. However, underwater noise has a negative effect on localization accuracy. In [24], an improved localization method based on a Parallel and Compact WOA (PCWOA) for localization in WSNs is proposed . In this study, an algorithm with Parallelization Method and Compact Technique has been proposed based on Whale Optimization Algorithm (PCWOA) for improving the DV-Hop performance.

In [25], the authors have proposed an optimal DVHop localization version based on the SLO algorithm for WSN. In this paper, to decrease the localization error, an efficient version of DVHop in the Social Learning Theory Class is implemented for WSNs. The advantage of this method is higher localization accuracy compared to the DVHOP method, but the complexity of the algorithm is high, and this can be a big challenge for sensor nodes.

In [26], a 3D DVHop localization algorithm based on Hop size correction and Sparrow-improved algorithm is presented. For improving the localization precision and decreasing the energy consumption, a new localization algorithm has been proposed for improving the DV-Hop algorithm based on hop size correction and error decrease with the help of Sparrow research optimization. The advantage of this method is that it is more accurate than the DDV-hop algorithm, the D-GAIDV-hop algorithm, and the HCLSO algorithm, but the algorithm is sensitive to noise and the localization accuracy decreases due to noise.

In [27], a node localization method in WSNs utilizing Gorilla Troops Optimizer is proposed. The advantage of this method is that it has more accuracy than the DDV-hop algorithm, the D-GAIDV-hop algorithm, and HCLSO, but it has a slower localization speed than the WOA optimization and the particle optimization algorithm.

[28] introduced an internal localization system based on low-energy Bluetooth technology and optimization algorithm inspired by nature. The advantage of this method is its higher accuracy than the DDV-hop algorithm, but its energy consumption is higher than the DVHop algorithm.

In [29], a collaborative localization algorithm based on Firefly optimization for smart IoT is presented. This method has been used in smart agriculture and its location accuracy is moderate. In [30], a localization method for WSN utilizing an Arithmetic optimization algorithm is proposed. In this method, the localization error is reduced, but the computational time of the proposed localization algorithm is increased. In [31], a localization algorithm based on machine learning has been presented for WSNs. This method has high accuracy in localization and reduced energy consumption, but has a high training time. A method (which is aware of the obstacles) for estimating the improved distance for localizing by the particles optimization algorithm for the WSNs is proposed in [32]. This method is more accurate than the particle algorithm, but has high error in some simulated scenarios.

In [33], an improved DV-Hop localization algorithm called MAOADV-Hop is proposed which is based on the modified Archimedes optimization algorithm (MAOA). In this method, a balance is achieved between localization speed and localization accuracy. MAOA is used to improve the localization accuracy of the algorithm. In addition, it uses the PSO algorithm to improve the localization speed of the algorithm.

In [34], an improved DV-Hop algorithm (named DEIDV-Hop) is proposed which is based on differential evolution (DE) and can improve the problem of possible error in calculating the average distance. The DE algorithm is used to locate the non-anchor nodes, and it reduces the localization error.

In [35], to solve the disadvantages of the DV-Hop algorithm, an algorithm based on differential evolution is proposed. In this method, some modifications of the DVHOP method are proposed. In the second step, the DV-hop algorithm performs the weighting process by leading the average hop distance error correction value in order to reduce the hop distance error. In the third step of the DV-hop algorithm, the differential evolution algorithm is used to optimize the positioning result of the unknown node.

In [36], a multi-stage algorithm for location is presented based on the DVHOP method. In the first stage, different weights are assigned to beacons based on the error of each hop, and the weighted minimum mean square error (MMSE) is calculated continuously. Finally, the nearest beacon nodes are selected to calculate the node position.

3. Proposed Method

In order for a sensor node to be able to do localization, it needs to, firstly, calculate or estimate its distance to three anchor nodes. In the case that three anchor nodes are adjacent to the sensor node, the distance can be calculated by sending a round trip signal and measuring the round-trip time. To measure the distance in this case, the relationship $d=v/t$ can be used, where v is the speed of radio signal transmission and t is the transmission time from the anchor node to the non-anchor sensor node. However, this method has challenges, one of which is the need to synchronize the clock among the nodes, and on the other hand, it is to be equipped with additional equipment for the sensor nodes, which costs the implementation. Another method is to use the RSSI algorithm, which compares the signal intensity on the transmitter and receiver and then estimates the distance. If the anchor nodes are in the neighbourhood of the sensor node, then the RSSI method calculates the distance with high accuracy. If the anchor nodes are not adjacent to the sensor node, then the RSSI method is prone to noise due to the larger distance dimension between the sensor node and the anchor node, and the distance measurement error will increase. On the other hand, one of the important challenges of the RSSI method in distance measurement is that it is sensitive to environmental noise. Due to high dynamicity of wireless environment, the RSSI measurements are extremely unpredictable. To solve this challenge, an improved version of DVHop is used to measure the distance of a sensor node from non-adjacent anchor nodes, which is less sensitive to environmental noise than the RSSI method.

For this reason, in the proposed method, DVHOP algorithm has been used to estimate the distance between anchor nodes and non-anchor nodes. The proposed method is presented based on an improved version of IDVHop algorithm, the three-reference, and WOA by chaotic maps and quadratic opposition-based learning.

Despite its simplicity, the DVHOP algorithm has a high localization error. To reduce the localization error, metaheuristic methods can be used to reduce the error. But the main problem of these methods is the high execution time. The WOA algorithm, like other metaheuristic methods, can help reduce the localization error of the algorithm, but it is faster than many other methods, which is why this method has been used in this article.

Therefore, the objectives of the article are as follows:

1- Reducing the location error

2- Increasing the location speed

The proposed method is explained in detail below.

3.1. The proposed method framework

More precisely, the proposed algorithm is described in Fig. 1 and works as follows. The following stages are considered for localization:

- Firstly, anchor and non-anchor nodes are distributed in the environment based on Probability Distribution.
- The anchor nodes can perform localization by GPS and transmit their location to other nodes by flooding algorithm.
- The anchor nodes determine their hop count and, then, can accordingly calculate the distance among themselves.
- The main challenge is locating non-anchor nodes. These nodes can calculate their location in two ways:
    - In the first method, there are at least three anchor nodes around it, and it can perform localization by them without requiring an error correction mechanism (because the exact location is obtained.).
    - In the second, the non-anchor node does not find three anchor nodes around itself and, to calculate its distance from the three anchor noes, it is necessary to calculate the distance from some of the non-neighbouring anchor nodes. In this situation, the distance between the sensor node and the anchor node is calculated by the DVHop algorithm (section 3.2).
- If the non-anchor node or smart object can calculate its distance from the three anchor nodes (which are all or some non-neighbouring), it would be possible to calculate the location of this sensor node by the three-reference localization method. In this mode, the anchor node and the smart object know the location of the three neighbouring or non-neighbouring anchor nodes; then, the smart object utilizing the DVHop can measure its distance from the three anchor nodes and, finally, can perform localization.
- In the proposed method, after localization by the three-reference method, it is possible to decrease the localization errors by the WOA-improved algorithm (section 3.3).
- In each iteration, the location of non-anchor node is calculated. This process continues until all non-anchor nodes are located.

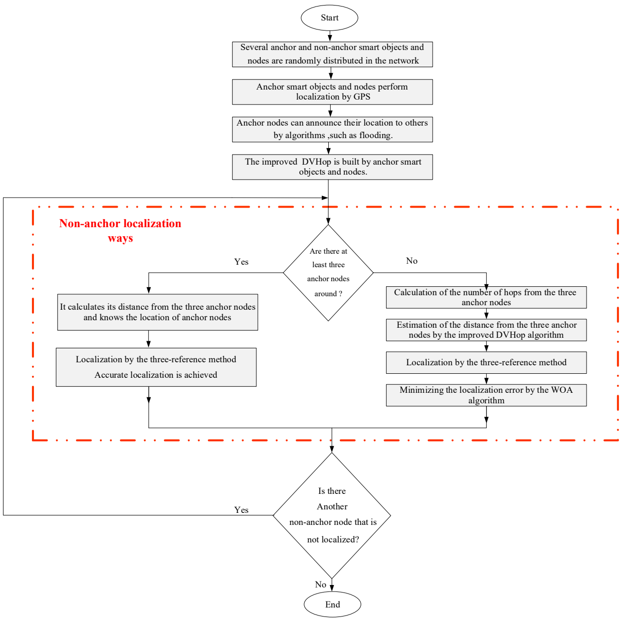

Fig 1. Localization framework in the proposed method

3.2. Evaluating and estimating the distance of the sensor node from the anchor nodes by the DVHop

In this mode, it is assumed that the non-anchor nodes have no neighbouring anchor nodes and utilize the DVHop method to estimate the distance from them. When a sensor node estimates its distance from the three anchor nodes, the estimation may not be exact. Therefore, the node location is obtained approximately. After determining the distance among nodes, the location of non-anchor nodes can be determined by the Multilateral Method or Maximum Likelihood Estimation Method.

The specific implementation process of the standard DV-Hop method basically includes the following three steps:

Calculation performed to find the minimum number of hops among sensor nodes: All anchor nodes first send a packet to their neighbour nodes through flooding method. This packet contains its own location and hop count values, whose initial value is set to zero. Neighbouring nodes only store the packet from the same node with the minimum number of hops and increment its value by one. Then, they send packets to their other neighbouring nodes. With this broadcast strategy, all sensor nodes calculate the location of each anchor node and the minimum number of hops for all anchor nodes.

Calculation of AHS[1] index of sensor nodes: AHS index of anchor nodes can be determined using (1) based on unbiased estimation method [37].

Calculation of the estimated location of non-anchor nodes: The position of non-anchor nodes can be obtained by the multi-dimensional method or the maximum likelihood estimation method when the distance between the nodes is determined. Then, the three-reference method is executed.

$$AHS_i = \frac{\sum_{j=1}^{Na} \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}}{\sum_{j=1}^{Na} h_{ij}} \qquad (1)$$

In (1), Na shows the number of anchor nodes and (x_i,y_i) and (x_j,y_j) show the location of anchor node *i* and anchor node *j*, respectively. h_{ij} is the minimum number of hops between anchor node *i* and anchor node *j*. AHS_i indicates the AHS of anchor node *i*. By using the AHS index, a sensor node can calculate its distance from each anchor node in the network. After that, each anchor node broadcasts its AHS to the network. Non-anchor nodes store only the AHS received from the nearest anchor node and consider it as their AHS. This anchor node can ensure that the AHS of the non- anchor nodes will be the same as the AHS of the nearest anchor node. After that, the distance between the non- anchor nodes and the anchor nodes is calculated by (2) [37].

$$d_{ui} = AHS_u \times h_{ui} \qquad (2)$$

where, AHS_u and h_{ui} represent, respectively, the AHS of the non-anchor node *u* and the minimum number of hops to the anchor node *i*. d_{ui} indicates the estimated distance between the non-anchor node *u* and the anchor node *i*. The following steps are performed:

Step 1: The initial AHS and its error are calculated. Assuming that all anchor nodes initially have the same influence on each other, the initial AHS of each anchor node can be calculated by (3). At the same time, the actual distance among anchor nodes is known in advance, and the estimated distance can be replaced by the multiplying AHS and the number of hops. Therefore, the error in the initial AHS can be obtained by each anchor by (4) [37].

$$AHS_i^{\text{init}} = \frac{\sum_{j=i+1}^{N_n} D_{ij} + \hbar_j}{\sum_{ii_i}^{K_0} h_j^2} \qquad (3)$$

$$\eta_i^{\text{init}} = \frac{\sum_{j \neq i}^{N_a} \left| Dis_{ij} - AHS_i^{\text{init}} \cdot h_{ij} \right|}{N_a - 1} \qquad (4)$$

where AHS_i^{init} represents the initial AHS of anchor node i. Dis_{ij} represents the actual distance between anchor node i and anchor node j. η_i^{init} is the initial AHS error value of anchor node i.

Step 2: The calculation error of the hops among anchors is calculated by (5) [37].

Where AHS_i^* represents the initial AHS or the AHS obtained with anchor node *i*.

Step 3: The weighted AHS is calculated, and the interaction among anchor nodes can be reflected by the error of each hop count. If the error of each hop count between anchor node *i* and anchor node *j* is large, anchor node *j* is assigned a smaller weight to weaken the AHS effect of anchor node *i*. Otherwise, more weight is given to node *j*. After that, the AHS can be updated by the weighted criterion. The equation for calculating weight and weighted AHS are given in (6) (7) [37].

$$a_{(i)j} = \left(\frac{1}{EH_{(i)j}}\right)^2 \qquad (6)$$

$$AHS_i^{(m)} = \frac{\sum_{j \neq i}^{N_a} \alpha_{(i)j} \cdot Dis_{ij} \cdot h_{ij}}{\sum_{j \neq i}^{N_a} \alpha_{(i)j} \cdot h_{ij}^{\ 2}} \qquad (7)$$

Step 4: Next, the weighted AHS error is calculated based on (8) [37].

$$\eta_i^{(m)} = \frac{\sum_{j \neq i}^{N_a} \left| Dis_{ij} - AHS_i^{(m)} \cdot h_{ij} \right|}{N_a - 1} \qquad (8)$$

Where, $\eta_i^{(m)}$) represents the AHS error obtained in m_{th} iterations of anchor node *i*.

Step 5: The optimal AHS is determined and if the value of $\eta_i^{(m)}$ is less than $\eta_i^{(m-1)}$, it returns to step 2 and recalculates the error of each hop count using $AHS_i^{(m)}$. Therefore, a new AHS of anchor node i can be recalculated by (7). According to the above process, iterative weight calculation is repeatedly performed on the AHS of the anchor node.

The standard DVHop or other available DVHops typically utilize all the anchor nodes for calculating the position of the non-anchor nodes. However, the number of hops among nodes is considered as a hop in the communication area; therefore, a specific distance deviation among nodes and the anchor nodes which are far from the non-anchor nodes has typically more significant hop values. It means that if an anchor node or a significant estimation error concerning the distance from a non-anchor node exists in the Multilateral localization process, a considerable deviation in estimated and real distances may remarkably decrease the precision of localizing the non-anchor nodes. Hence, a new idea must be proposed for determining the optimal anchor nodes for localizing the non-anchor nodes [37].

[1] **Estimated average hop size**

To find the optimal anchor nodes for localization, a grouping strategy has been introduced based on the estimated distance between the anchor and non-anchor nodes. Generally, the farther the anchor node from the non-anchor node is, the more significant the probability of error in distance estimation is. Moreover, the minimum number of anchor nodes for localizing a non-anchor node is three. The proper localization target function could be considered as (9)[37]:

$$\gamma = \frac{1}{N_a}\sum_{i=1}^{N_a}\left(\sqrt{(x_u^* - x_i)^2 + (y_u^* - y_i)^2} - d_{ui}\right)^2 \quad (9)$$

Where, $(x_u^*,)$ is the position of the non-anchor node, u; the goals are minimizing this function and finding the position for minimizing it. In the proposed method, around every estimated position created by the three-reference method, it is possible to create some position, and every position could be considered a member of the WOA algorithm. The goal is updating and optimizing the position by the WOA algorithm to minimize the target function and localization error.

3.3. Decreasing the three-reference localization error by the WOA algorithm

Localization by the three-reference Method contains errors because evaluating the distance by the DVHop algorithm itself contains errors. To decrease and remove this error, the WOA optimization could be used. In the proposed method, every solution is a 2D situation or position, considered as a WOA. In the WOA algorithm, by spiral and circular movements around the random movements or optimal answers, the problem space is searched.

In these movements, the WOAs or problem solutions can search the victim's surroundings more to find the optimal answer and better solutions. The WOA algorithm does the search in two forms:

- circular movements
- spiral movements

In the proposed method, each WOA is a position which could be formulated as (10):

$$\mathrm{W_i} = (\mathrm{X_i}, \mathrm{Y_i}) \quad (10)$$

Where, W_i is a solution (WOA) and X_i, Y_i are, the position of the $\mathrm{i_{th}}$ WOA. Firstly, a primary population of random WOA algorithms is created like (11):

$$\mathrm{W_{Pop}} = \{(\mathrm{X_1}, \mathrm{Y_1}), (\mathrm{X_2}, \mathrm{Y_2}), \ldots, (\mathrm{X_n}, \mathrm{Y_n})\} \quad (11)$$

Where, *n* is the number of solutions (WOA) and X_i, Y_i is the position of $\mathrm{i_{th}}$ WOA. The WOA algorithm, in addition to these two movements, has a random search which will later be formulated. For the circular movements, the WOAs do the circulation around the optimal answer; then, based on (12) and (13), firstly, the distance of a solution from an optimal answer is calculated in line with the *x* and *y* position [36].

$$\mathrm{D_x} = |\mathrm{C}\vec{\mathrm{X}}^*(\mathrm{t}) - \mathrm{X_i}(\mathrm{t})| \quad (12)$$

$$\mathrm{D_y} = |\mathrm{C}\vec{\mathrm{Y}}^*(\mathrm{t}) - \mathrm{Y_i}(\mathrm{t})| \quad (13)$$

Where, $X_i(t)$ and $Y_i(t)$ are, respectively, the position of *x* and *y* of the $\mathrm{i_{th}}$ WOA. $\vec{X}^*(t)$ and $\vec{Y}^*(t)$ are the optimal position of the gained *x* and *y* until the *t* iteration. Here, C is a convergence coefficient. Based on the above equation, for the circular movements, it is possible to use the (14) and (15) in line with the *x* and *y* axes [36].

$$\vec{X}(t+1) = \vec{X}^*(t) - A \cdot D_x \quad (14)$$

$$\vec{Y}(t+1) = \vec{Y}^*(t) - A \cdot D_y \quad (15)$$

Where, $\vec{X}(t+1)$ and $\vec{Y}(t+1)$ are the updated position of the WOA in the new iteration and in line with the *x* and *y* axes. The C and A parameters are, respectively, calculated as (16) and (17) [36].

$$\vec{\mathrm{A}} = 2\vec{\mathrm{a}} \cdot \vec{\mathrm{r_1}} - \vec{\mathrm{a}} \quad (16)$$

$$\mathrm{C} = 2\mathrm{r_2} \quad (17)$$

Where, $\vec{r_1}$ and $\vec{r_2}$ are two random numbers between 0 and 1, and *a* is calculated based on (18) and is a decreasing value, from 2 to 0 [36].

$$\mathrm{a} = 2 - 2\mathrm{t}/\mathrm{T}_{max} \quad (18)$$

Where, T_{max} is the maximum the WOA algorithm iteration. For doing the spiral movements, the following equations are utilized: (19), (20), (21), and (22) [36].

$$\vec{\mathrm{X}}(\mathrm{t}+1) = \vec{\mathrm{X}}^*(\mathrm{t}) + \mathrm{D_x}\mathrm{e}^{\mathrm{bl}} cos\,(2\pi\mathrm{l}) \quad (19)$$

$$\mathrm{D_x} = \left|\vec{\mathrm{X}}^*(\mathrm{t}) - \vec{\mathrm{X}}(\mathrm{t})\right| \quad (20)$$

$$\vec{\mathrm{Y}}(\mathrm{t}+1) = \vec{\mathrm{Y}}^*(\mathrm{t}) + \mathrm{D_y}\mathrm{e}^{\mathrm{bl}} cos\,(2\pi\mathrm{l}) \quad (21)$$

$$\mathrm{D_y} = \left|\vec{\mathrm{Y}}^*(\mathrm{t}) - \vec{\mathrm{Y}}(\mathrm{t})\right| \quad (22)$$

Where, D_x and D_y are the distances (in line with the horizontal and vertical axes) from the optimal solution such as$\vec{W}^*(t)$. Here, l is a random number between +1 and -1, and b is a numerical constant between 0 and 1. Each WOA can do the spiral or circular search with the probability of p, as (23) [36].

$$\vec{W}(t+1) = \begin{cases} \vec{W}^*(t) - A \cdot D & P < 0.5 \\ \vec{W}^*(t) + D_{x,y}e^{ht} cos\,(2\pi l) & P \geq 0.5 \end{cases} \quad (23)$$

To do random movements, the following equations are used: (24), (25), (26), and (27) [36].

$$D_x = \left|C * \vec{X}_{rand} - \vec{X}(t)\right| \quad (24)$$

$$\vec{X}(t+1) = \vec{X}_{rand} - A \cdot D \quad (25)$$

$$D_x = \left|C * \vec{Y}_{rand} - \vec{Y}(t)\right| \quad (26)$$

$$\vec{Y}(t+1) = \vec{Y}_{rand} - A \cdot D \quad (27)$$

In the above equations, $\vec{X}_{rand}$ and $\vec{Y}_{rand}$ are random solutions in the population in line with x and y axes. A proper approach for increasing the precision of the WOA algorithm is utilizing chaotic functions, such as Tent, which administrates the random variables in the WOA algorithm by a more random behaviour and decreases the probability of being trapped in the local optimum. In (28), the chaotic Tent function is formulated [36].

$$z_{k+1} = \begin{cases} z_k/\beta, & 0 < z_k \leq \beta \\ (1 - z_k)/(1 - \beta), & \beta < z_k \leq 1 \end{cases} \quad (28)$$

Where, $z_0 = 0.125$ and $\beta = 2.59$.

The chaotic behaviour causes the WOA algorithm to do more problem space search for finding more optimal

answers. For quantifying the primary population, a proper primary population means a population in which the members' situation is closer to the situation of the optimal solution, which can effectively speed up the algorithm convergence. Opposition-based Learning Strategy is a simple and effective tool to achieve this goal.
It is possible to create another population with new members in a situation opposing the situation of the present situation in the solution space. The advantage is that the probability of two contradictory members being close to the optimal solution is 50%. The mathematical wording of the current population and the reverse population could be expressed as (29) and (30)[36].

$$X_{i,j} = Lb_j + z_{i,j}(Ub_j - Lb_j) \quad (29)$$

$$OX_{i,j} = b_j + Lb_j - X_{i,j} \quad (30)$$

Where, Ub_j and Lb_j are, respectively, the bottom area and top area after j. $X_{i,j}$ is the current solution of the problem. $OX_{i,j}$ is the reciprocal situation. Here, b_j is the center between the $X_{i,j}$ solution and its reciprocal solution or $OX_{i,j}$ which is calculated as (31) [36].

$$b_j = \left| X_{i,j} - \frac{Ub_j + Lb_j}{2} \right| \quad (31)$$

To decrease the localization error, the improved WOA algorithm, contradictory learning, and chaos theory are used as follows:

- Creating some random situations as the WOA population around each sensor situation
- Regulating the parameters of the WOA algorithm
- Creating a random sequence of chaotic function
- Running the random, spiral, and circular search phase
- Reciprocal learning among solutions
- Choosing the optimal situation and location for each sensor node
- Increasing the algorithm iteration counter
- Updating the parameters of the WOA algorithm
- Evaluating the WOA algorithm in the last iteration and localizing the sensor based on the location of the optimal WOA

3.4. Why is the whale optimization algorithm used?

To solve the problem of DVHOP algorithm (localization error), metaheuristic algorithms can be used. For example, A PSO based method has presented in [38], and methods based on GA have proposed in [39, 40].
Evolutionary-based algorithms are widely investigated meta-heuristic algorithms in engineering domains to address optimization problems. In the proposed method, the whale optimization algorithm was used for the following reasons [41] :

(1) Simple structure,
(2) Adaptability in dynamic condition,
(3) Solving low-dimensional and uni-model problems,
(4) Solving continues and convex problems,
(5) Local optima avoidance capability,
(6) Fast convergence speed due to its exploration and exploitation ability

The efficiency and performance of Whale Optimization have been evaluated in several papers, and in a comprehensive comparison with other metaheuristic methods such as 1 and 2, the speed of Whale Optimization algorithm and its better performance have been shown. [41-43]
The simulations also showed that the proposed method has been able to reduce the localization error faster than other metaheuristic methods.

4. Simulation Results

In this section, the proposed method has been implemented for localizing in smart objects and sensors. For implementing the proposed method, MATLAB software (version 2021) is used.
In order to evaluate the efficiency of the proposed method, several algorithms have been evaluated. Since the proposed method is based on the DVHOP method, some versions of the DVHOP method have been investigated. In addition, several methods based on metaheuristic techniques have also been compared.
The main goal of the proposed method is to reduce localization error and reduce localization time, so the following parameters have been used to evaluate the proposed method:

4.1. Simulation parameters

- *Localization Error*

Suppose that the situation of a sensor node, such as i equals $(x_i . y_i)$ and the situation estimated for that sensor node equals$(x_i' . y_i')$; then, the localization error for that sensor node is calculated in (32):

$$E_i = \sqrt{(x_i - x_i')^2 + (y_i - y_i')^2} \quad (32)$$

In (24), E_i equals the localization error for the ith node of the sensor or smart object.

- *Average Localization Error*

If the network has n number of smart objects or sensor nodes, then, the average localization error of all sensor nodes is calculated by (33):

$$E = \frac{\sum_{i=1}^{N} E_i}{N} = \frac{E_1 + E_2 + E_3 + .. + E_N}{N} \quad (33)$$

If the number of experiments equals M, then, the average localization error of all experiments is calculated by (34):

$$\bar{E} = \frac{E_1 + E_2 + \cdots + E_M}{M} \quad (34)$$

- *Standard deviation index*

The standard deviation index is very crucial for localization and evaluating the stability of localization algorithm behaviour. It is formulated based on (35):

$$std = \sqrt{\frac{1}{R} \sum_{i=1}^{R} (E_i - \bar{E})^2} \quad (35)$$

The more minimum the standard deviation average of localization error in an algorithm is, the more stable is the behaviour of that algorithm while localizing, and it is more trustworthy.

- *Localization Time*

Time spent performing localization.

4.2. Simulation and analysis of results

Two scenarios were used in the simulation, which differ in the number of nodes and the degree of network density. In Fig. 2 and Fig. 3, two network scenarios are illustrated. In the first scenario, there are 10 anchor nodes and 30 sensor nodes. In the second scenario, there are 30 anchor nodes and 70 sensor nodes. In simpler words, in the second scenario, there is more significant density of anchor and sensor nodes.

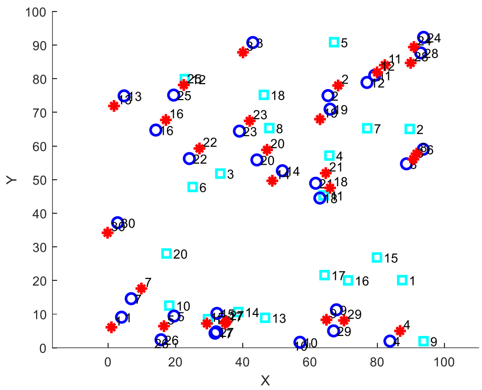

Fig. 2. The first scenario with a tiny density of smart objects and nodes

In the first scenario, the network density is less than the second scenario; the less the network density is, the weaker the DVHop efficiency is. Therefore, the precision of the localization algorithm will decrease.

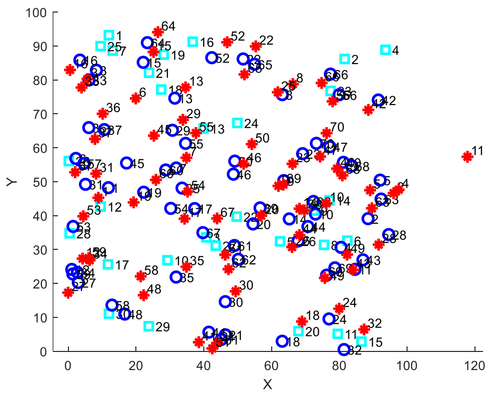

Fig. 3. The second scenario with a significant density of smart objects and nodes

In this section, the proposed method is evaluated by various parameters in order for its performance to be analyzed and compared regarding localization. In Fig. 4 and Fig. 5, respectively, the localization error will be shown based on the iteration of the improved WOA algorithm, in the first and second scenarios. Considering the simulation performed, the localization error process based on the iteration of WOA is decreasing because the WOA algorithm is optimizing and decreasing the localization error. However, the process of error decrease is faster in the second scenario. In better words, the denser the network is, the more precise is the distance model based on hop created by the DVHop algorithm, so the localization error decreases. In the first experiment, based on observations, error has reached 5.32, but it has decreased to 0.429 in the second scenario. Based on the experiments, the WOA algorithm has decreased the objects localization error; this decrease depends on the primary population, the number of iterations, and the network density. The reason of the decrease in localization error is the effect of the WOA optimization algorithm, which causes situations with fewer errors.

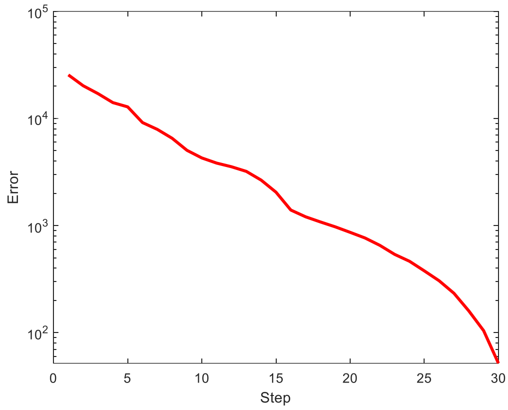

Fig. 4. Localization error in the first scenario

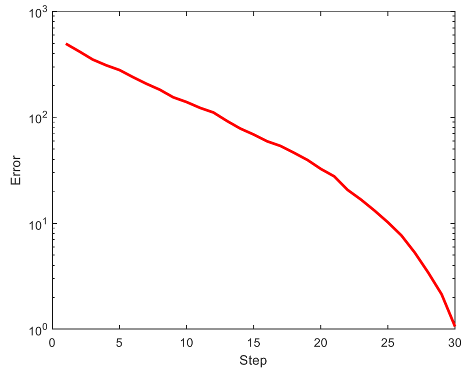

Fig. 5. Localization error in the second scenario

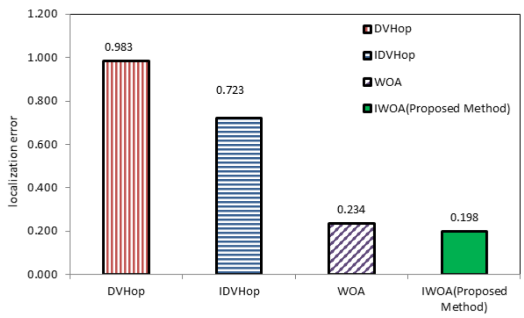

Fig. 6. Comparison of the localization error of the proposed method in the density of 10 anchor nodes and 30 sensor nodes

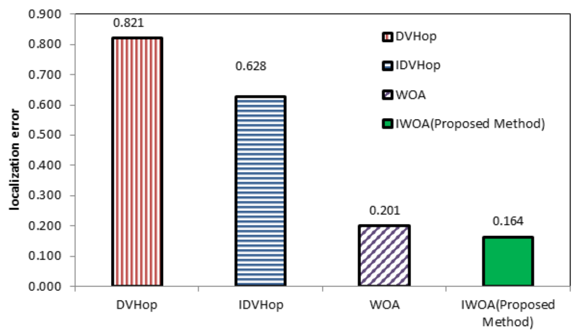

Fig. 7. Comparison of the localization error of the proposed method in the density of 20 anchor nodes and 40 sensor nodes

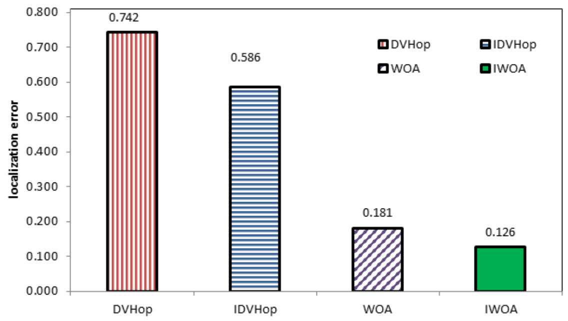

Fig. 8. Comparison of the localization error of the proposed method in the density of 30 anchor nodes and 70 sensor nodes

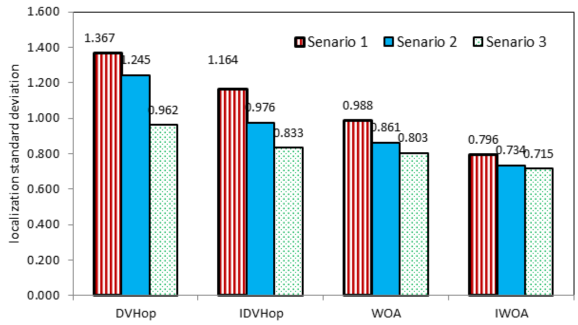

Fig. 9. Comparing the standard deviation of the proposed method localization error in three different scenarios

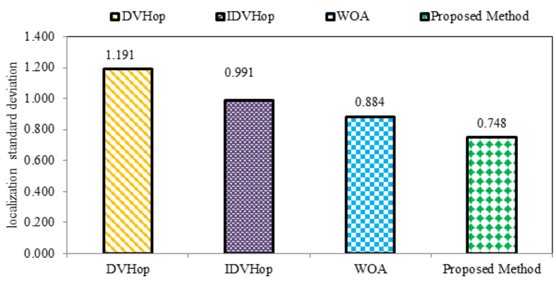

Fig. 10. Comparing the standard deviation of the

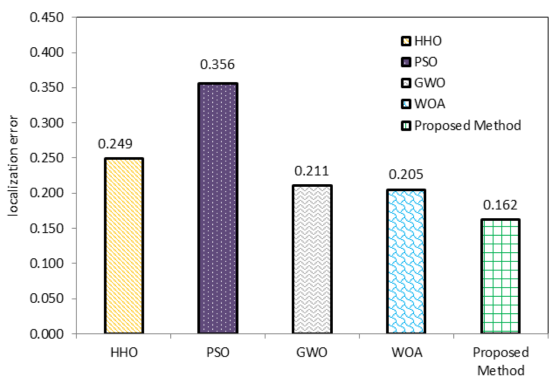

Fig. 11. Comparing the average localization error of the proposed method with the meta-heuristic methods

In Fig. 6, the localization error of a network with 10 anchor nodes and 30 sensor nodes is displayed. In Fig. 7 and Fig. 8, the ratio of anchor nodes to sensor nodes are, respectively, 0.5 and 0.42 to show the effect of increasing the density on decreasing the localization error (scenarios 2 and 3). Here, the following methods are implemented by MATLAB, so no references are written: the DVHop, IDVHop, WOA, and the proposed method (IWOA).

The experiments analysis shows that, on average, if the number of anchor nodes is 10 and the number of sensor nodes is 30, then, the localization errors of the DVHop algorithm, IDVHop algorithm, the WOA algorithm, and the proposed algorithm (IWOA) are, respectively, as follows: 0.983, 0.723, 0.234, and 0.198. If the number of anchor nodes is 20 and the number of sensor nodes is 40, then, the localization errors of the DVHop algorithm, IDVHop algorithm, the WOA algorithm, and the proposed algorithm (IWOA) are, respectively, as follows: 0.821, 0.628, 0.201, and 0.164. If the density of anchor nodes and sensor nodes is increased, respectively, to 30 and 70, then, the localization errors are, respectively, 0.742, 0.586, 0.181, and 0.126. The simulation performed in this section brings about the following results for localizing smart sensors and objects.

Increasing the density of anchor and sensor nodes causes the DVHop and the IDVHop perform better and also causes a decrease in their localization error.

The localization error of IDVHop is less than DVHop in various densities.

When the number of anchor nodes increases compared to the number of sensor nodes, more decrease is observed in localization error. In better words, increasing the density of anchor nodes has a more significant effect on decreasing the error compared to increasing the sensor nodes.

In Fig. 9, the average standard deviation of the proposed method and other methods for localization are compared. Also, in Fig. 10, the average standard deviation of all the three scenarios are compared to compare the stability of the proposed method with the three rival methods (WOA, DVHop, and IDVHop).

Proposed method localization error in various densities

The experiments show that, as the network density increases, the standard deviation of localization error of all methods decreases. It means that localization algorithms with more density have a more stable behaviour due to an increase in the DVHop precision. Based on evaluations, the standard deviation of localization errors of the DVHop algorithm, IDVHop algorithm, WOA algorithm, and the proposed algorithm (IWOA) are, respectively, 1.191, 0.991, 0.884, and 0.748. Based on the experiments performed, the average standard deviation of the proposed method is less than the DVHop algorithm, the IDVHop algorithm, and the WOA algorithm. It means that the proposed algorithm is more stable while localizing. The proposed algorithm, compared to the WOA algorithm, has succeeded in decreasing the localization standard deviation by about 11.37%. In better words, the proposed algorithm, compared to the standard WOA algorithm, has increased the stability in localization by about 11.37%. Based on the experiments, the localization error of the proposed method is, on average, 0.162 in three scenarios, and the localization error of the standard WOA algorithm equals 0.205. On average, the proposed algorithm has a localization error of less than about 20.97%. In Fig. 11, the average localization error and, in Fig. 12, the average standard deviation of the proposed method is compared to the meta-heuristic methods.

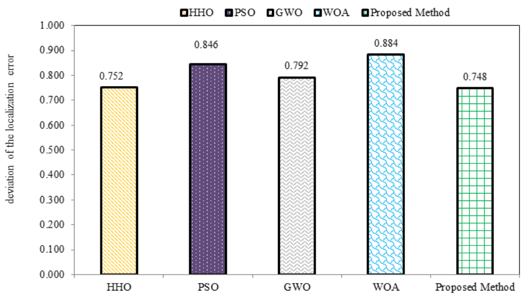

Fig. 12. Comparing the average standard deviation of the localization error of the proposed method with the meta-heuristic methods

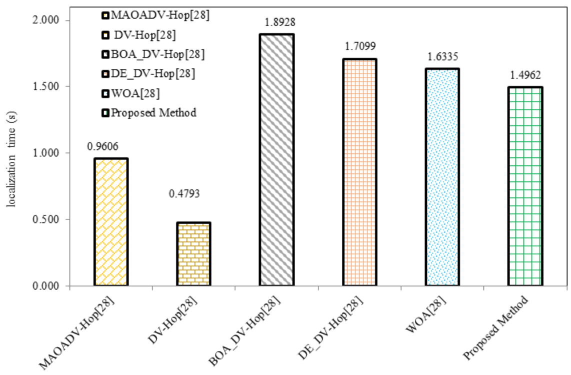

Fig. 13. Comparison of the proposed method localization time with similar localization methods

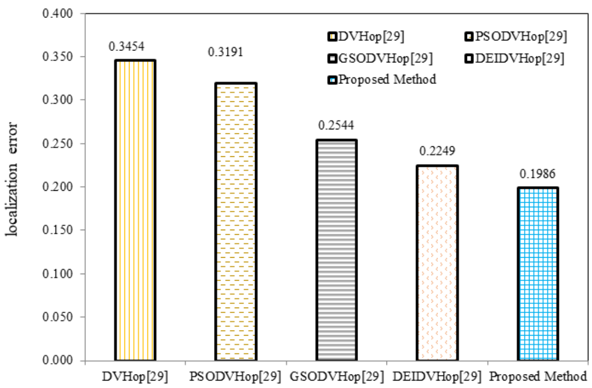

Fig. 14. Comparison of the proposed method localization error with the DVHop-based localization methods

Simulation showed that the average localization errors in the proposed algorithm, the WOA optimization algorithm, the GWO optimization algorithm, PSO, and HHO are, respectively, 0.162, 0.205, 0.211, 0.356, and 0.249. Simulation has shown that the localization error in the proposed method is less than WOA algorithm, GWO algorithm, PSO algorithm, and HHO algorithm. The best performance in localization error index, after the proposed method, belongs to the WOA algorithm and, then, GWO. The best performance in the standard deviation index of the localization error, after the proposed method, belongs, respectively, to HHO and GWO. In better words, regarding the localization error index, the best performance belongs to the proposed method and, then, GWO; the worst performance belongs to PSO. The best and worst performances regarding the standard deviation index, after the proposed method, belong to HHO and GWO.

In [28], methods, such as the DVHop MAOADVHop, BOA-DVHop, and DE-DVHop are used for localization. The running time of the localization algorithm of each method is, per second, compared with the running time of the WOA algorithm in Fig. 13.

Based on the experiments performed, the localization time of the proposed method is 1.496 seconds; this time is shorter than the WOA algorithm, BOA-DVHop algorithm, and DE-DVHop algorithm. The proposed method has a more extended time based on DVHop compared to non-optimized localization methods, but the proposed method error is, instead, much less than those methods. However, as shown in Fig. 13, the proposed method is faster and achieves results in less time compared to other methods that use metaheuristic methods.

In [29], for localization, methods, such as the DVHop, PSODVHop, GSODVHop, and DEIDVHop have been used; here, the proposed method is compared with this article's results. All experiment conditions are considered equal.

In Fig. 14, the proposed method is compared to the following regarding the localization error: the DVHop, PSODVHop, GSODVHop, and DEIDVHop.

The experiments show that the localization errors of the proposed method, the DVHop, PSODVHop, GSODVHop, and DEIDVHop are, respectively, 0.1986, 0.3454, 0.3191, 0.2544, and 0.2249, and the proposed method has fewer errors among these because the improved WOA algorithm is smart in finding a solution and a better situation.

Simulation results showed that the improved Wall algorithm has less localization error than the Wall algorithm, and both IDVHop and DVHop algorithms have less localization error. By increasing the density of anchor and sensor nodes in the network, the accuracy of the DVHop model increases, and since the proposed method is based on DVHop, its accuracy will also increase with the increase in the density of network nodes.

4.3. Time complexity of the proposed algorithm

The execution time of the proposed method depends on the execution time of the following two algorithms: DVHOP and WOA.

The time complexity of the DVHOP algorithm is O(n2), where n is the number of non-anchor nodes.

The parameters involved in the computational time complexity of WOA algorithm are the number of iterations of the algorithm (M), the number of whale populations (N), and the dimensionality (D). Therefore, the total space complexity of the WOA for each non-anchor nodes is O(M*N*D).

The important point here is that for all non-anchor nodes, the DVHOP and WOA algorithms are not implemented.

5. Conclusion

WSNs are a form of smart networks that form through a free organization and combination of tens of thousands of sensor nodes by wireless communication technology used in smart cities and IoT. Although GPS does a precise localization in wireless sensor and smart networks, equipping all sensor nodes with GPS in WSNs is unrealistic due to high costs and limited practicality.

Indirect localization algorithms can have an acceptable localization precision; as a result, they have attracted much attention in recent years. The DVHop algorithm is a distributed localization method in the network.

The main advantage of the DVHOP method is its simplicity and its major problem is the high localization error. To reduce the localization error, the proposed method uses the whale optimization algorithm, which has a higher speed for finding the solution than many other metaheuristic methods. Therefore, localization can be done faster and with fewer errors.

In the proposed method, after evaluating the distance of a sensor node from the three anchor nodes by the improved DVHop algorithm, localization can be performed by the three-reference method. The three-reference method contains localization error due to the DVHop algorithm errors; therefore, in the present study, the improved WOA algorithm which is based on chaos theory and opposition-based learning has been used to decrease the localization error. The results showed that the proposed algorithm has a lower localization error than the DVHOP method and is faster than methods that use metaheuristic techniques along with DVHOP.

As a future work, better solutions can be achieved by combining the whale optimization method with another method such as genetics.

6. References

[1] M. Faris, M. N. Mahmud, M. F. M. Salleh, and A. Alnoor, "Wireless sensor network security: A recent review based on state-of-the-art works," *International Journal of Engineering Business Management,* vol. 15, p. 18479790231157220, 2023.

[2] K. K. Gola, M. Dhingra, B. Gupta, and R. Rathore, "An empirical study on underwater acoustic sensor networks based on localization and routing approaches," *Advances in Engineering Software,* vol. 175, p. 103319, 2023.

[3] Y. Wu, R. Chen, W. Fu, W. Li, and H. Zhou, "CWIWD-IPS: A crowdsensing/walk-surveying inertial/Wi-Fi data-driven indoor positioning system," *IEEE Internet of Things Journal,* vol. 10, no. 10, pp. 8786-8798, 2023.

[4] H. Sun, D. Wang, H. Li, and Z. Meng, "An improved DV-Hop algorithm based on PSO and Modified DE algorithm," *Telecommunication Systems,* vol. 82, no. 3, pp. 403-418, 2023.

[5] M. K. Mohanty, P. K. G. Thakurta, and S. Kar, "Efficient sensor node localization in precision agriculture: an ANN based framework," *OPSEARCH,* 2023/02/08 2023, doi: 10.1007/s12597-023-00625-4.

[6] K. Sathish, R. C. Venkata, R. Anbazhagan, and G. Pau, "Review of localization and clustering in USV and AUV for underwater wireless sensor networks," in *Telecom*, 2023, vol. 4, no. 1: MDPI, pp. 43-64.

[7] A. Mahfozi and Y. Darmani, "A Trust and Energy-based routing framework for the IoT network," *TABRIZ JOURNAL OF ELECTRICAL ENGINEERING,* 2023.

[8] M. Parandeh and S. Aghdasi, "Position-based Energy-Efficient Data Forwarding Protocol for Visual Sensor Networks," *TABRIZ JOURNAL OF ELECTRICAL ENGINEERING,* vol. 47, no. 1, pp. 29-38, 2017.

[9] P. Yadav and S. C. Sharma, "A systematic review of localization in WSN: Machine learning and optimization‐based approaches," *International journal of communication systems,* vol. 36, no. 4, p. e5397, 2023.

[10] P. Singh, P. Singh, N. Mittal, U. Singh, and S. Singh, "An optimum localization approach using hybrid TSNMRA in 2D WSNs," *Computer Networks,* vol. 226, p. 109682, 2023.

[11] A. H. M. Hashim *et al.*, "Application of ANFIS and ANN for Partial Discharge Localization in Oil Through Acoustic Emission," *IEEE Transactions on Dielectrics and Electrical Insulation,* vol. 30, no. 3, pp. 1247-1254, 2023.

[12] I. Bizon, A. Nimr, P. Schulz, M. Chafii, and G. P. Fettweis, "Blind transmitter localization using deep learning: a scalability study," in *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, 2023: IEEE, pp. 1-6.

[13] Y. Cao and J. Xu, "DV-Hop-based localization algorithm using optimum anchor nodes subsets for wireless sensor network," *Ad Hoc Networks,* vol. 139, p. 103035, 2023.

[14] X. Wu, , L. Zhao, , X. Zhu, and "Efficient semidefinite solutions for TDOA-based source localization under unknown PS," vol. 91, p. 101783, 2023.

[15] Y. Li, , others, and "Research on compression sensing positioning algorithm of indoor complex environment visible light indoor based on hybrid APIT," vol. 2022, 2022.

[16] S. Nematzadeh *et al.*, "Maximizing coverage and maintaining connectivity in WSN and decentralized IoT: an efficient metaheuristic-based method for environment-aware node deployment," vol. 35, pp. 611-641, 2023.

[17] J. Chen *et al.*, "A Neighborhood Grid Clustering Algorithm for Solving Localization Problem in WSN Using Genetic Algorithm," vol. 2022, 2022.

[18] G. Farjamnia, Y. Gasimov, C. Kazimov, and M. hashemi, "A Survey of DV-Hop Localization Methods in Wireless Sensor Networks," 1399.

[19] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances in Engineering Software,* vol. 95, pp. 51-67, 2016/05/01/ 2016, doi: https://doi.org/10.1016/j.advengsoft.2016.01.008.

[20] K. V. Lopes, H. R. Santos, L. N. Mendonça, J. P. Pereira, and D. S. Pires, "COMPARATIVE STUDY OF METAHEURISTICS BASED ON SWARM INTELLIGENCE: WHALE OPTIMIZATION ALGORITHM AND PARTICLE SWARM OPTIMIZATION," in *XL Ibero-Latin American Congress on Computational Methods in Engineering*, 2019, vol. 1, no. 01.

[21] S. Gao and Y. Ma, "A Multi-Objective Optimization Framework That Incorporates Interpretable CatBoost and Modified Slime Mould Algorithm to Resolve Boiler Combustion Optimization Problem," *Biomimetics,* vol. 9, no. 11, p. 717, 2024. [Online]. Available: https://www.mdpi.com/2313-7673/9/11/717.

[22] J. Fé, S. D. Correia, S. Tomic, and M. Beko, "Swarm optimization for energy-based acoustic source localization: A comprehensive study," *Sensors,* vol. 22, no. 5, p. 1894, 2022.

[23] Y. Lin, Z. Zhang, and H. E. Najafabadi, "Underwater source localization using time difference of arrival and

frequency difference of arrival measurements based on an improved invasive weed optimization algorithm," *IET Signal Processing,* vol. 16, no. 3, pp. 299-309, 2022.
[24] R.-B. Wang, W.-F. Wang, L. Xu, J.-S. Pan, and S.-C. Chu, "Improved DV-Hop based on parallel and compact whale optimization algorithm for localization in wireless sensor networks," *Wireless Networks,* vol. 28, no. 8, pp. 3411-3428, 2022.
[25] T. K. Mohanta and D. K. Das, "Improved DV-Hop localization algorithm based on social learning class topper optimization for wireless sensor network," *Telecommunication Systems,* vol. 80, no. 4, pp. 529-543, 2022.
[26] P. Gou, Z. Yu, X. Hu, and K. Miao, "[Retracted] Three‐Dimensional DV‐Hop Localization Algorithm Based on Hop Size Correction and Improved Sparrow Search," *Wireless Communications and Mobile Computing,* vol. 2022, no. 1, p. 1540110, 2022.
[27] Q. Liang, S.-C. Chu, Q. Yang, A. Liang, and J.-S. Pan, "Multi-group gorilla troops optimizer with multi-strategies for 3D node localization of wireless sensor networks," *Sensors,* vol. 22, no. 11, p. 4275, 2022.
[28] A. Corbacho Salas, "Indoor positioning system based on bluetooth low energy," Universitat Politècnica de Catalunya, 2014.
[29] C. p. Liu, B. Xia, and L. Zhang, "Firefly Optimization‐Based Cooperative Localization Algorithm for Intelligent IoT," *Discrete Dynamics in Nature and Society,* vol. 2022, no. 1, p. 3398071, 2022.
[30] S. J. Bhat and S. KV, "A localization and deployment model for wireless sensor networks using arithmetic optimization algorithm," *Peer-to-Peer Networking and Applications,* vol. 15, no. 3, pp. 1473-1485, 2022.
[31] Y. H. Robinson, S. Vimal, E. G. Julie, K. Lakshmi Narayanan, and S. Rho, "3-dimensional manifold and machine learning based localization algorithm for wireless sensor networks," *Wireless Personal Communications,* pp. 1-19, 2022.
[32] H. Sun *et al.*, "An improved DV-Hop algorithm based on PSO and Modified DE algorithm," vol. 82, pp. 403-418, 2023.
[33] M. Cheng, T. Qin, and J. Yang, "Node localization algorithm based on modified Archimedes optimization algorithm in wireless sensor networks," *Journal of Sensors,* vol. 2022, no. 1, p. 7026728, 2022.
[34] D. Han *et al.*, "Enhancing the sensor node localization algorithm based on improved DV-hop and DE algorithms in wireless sensor networks," vol. 20, p. 343, 2020.
[35] W. Zhang and X. Yang, "DV-Hop Location Algorithm Based on RSSI Correction," *Electronics,* vol. 12, no. 5, p. 1141, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/5/1141.
[36] M. Li *et al.*, "A chaotic strategy-based quadratic opposition-based learning adaptive variable-speed whale optimization algorithm," vol. 193, pp. 71-99, 2022.
[37] T. Chen *et al.*, "An Enhanced DV-Hop Localization Scheme Based on Weighted Iteration and Optimal Beacon Set," vol. 11, 2022, doi: 10.3390/electronics11111774.
[38] Q. Shi, Q. Xu, and J. Zhang, "An improved DV-Hop scheme based on path matching and particle swarm optimization algorithm," *Wireless Personal Communications,* vol. 104, pp. 1301-1320, 2019.
[39] G. Sharma and A. Kumar, "Improved range-free localization for three-dimensional wireless sensor networks using genetic algorithm," *Computers & Electrical Engineering,* vol. 72, pp. 808-827, 2018.
[40] V. Kanwar and A. Kumar, "Range free localization for three dimensional wireless sensor networks using multi objective particle swarm optimization," *Wireless Personal Communications,* vol. 117, no. 2, pp. 901-921, 2021.
[41] N. Rana, M. S. A. Latiff, S. i. M. Abdulhamid, and H. Chiroma, "Whale optimization algorithm: a systematic review of contemporary applications, modifications and developments," *Neural Computing and Applications,* vol. 32, pp. 16245-16277, 2020.
[42] I. O. H. Ahmed S. Menesy , Hamdy M. Sultan, "Comparison of Particle Swarm and Whale Optimization Algorithms for Optimal Power Flow Solution," *International Journal of Engineering Research & Technology (IJERT),* vol. 11, no. 12, 2022.
[43] A. S. Ahmed, M. A. Attia, N. M. Hamed, and A. Y. Abdelaziz, "Comparison between genetic algorithm and whale optimization algorithm in fault location estimation in power systems," in *2017 Nineteenth International Middle East Power Systems Conference (MEPCON)*, 2017: IEEE, pp. 631-637.