Indonesian Journal of Electrical Engineering and Computer Science
Vol. 41, No. 2, February 2026, pp. 515~531
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v41.i2.pp515-531

A new approach for distance vector-Hop localization algorithm improvement in wireless sensor networks

Omar Arroub[1], Anouar Darif[2], Rachid Saadane[3], My Driss Rahmani[1], Zineb Aarab[1]
[1]LRIT-GSCM Associated Unit to CNRST (URAC 29), FSR Mohammed V-Agdal University, Rabat, Morocco
[2]LIMATI, Faculte Polydisciplinaire, University of Sultan Moulay Slimane, Beni Mellal, Morocco
[3]SIR2C2S/LASI-EHTP, Hassania School of Public Labor, Oasis, Morocco

Article Info

Article history:

Received Mar 14, 2025
Revised Nov 14, 2025
Accepted Dec 30, 2025

Keywords:

Distance vector-Hop
Grey wolf optimization
Localization
Opposition-based learning
Wireless sensor network

ABSTRACT

This article shows a new range-free localization technique based on a meta-heuristic algorithm (MA) dedicated to wireless sensor network (WSN), named sequential online-grey wolf optimization-distance vector-Hop (SO-GWO-DVHOP). Indeed, we use the improved GWO based on selective opposite learning to improve GWO in order to enhance the traditional DVHOP localization algorithm. In reality, we choose GWO due to its better outcomes compared to other meta-heuristics, which leads us to improve this algorithm further. In the literature, the improvement works of GWO try to reconstruct the hierarchy of GWO or improve specifically the role of omega individuals. In our contribution, we opt for opposition-based learning (OBL) to ameliorate GWO, aiming to further enhance the quality of localization made by DVHOP. On the other hand, we make an empirical comparison of DVHOP and its improved versions in terms of accuracy. The results of the simulation demonstrate that SO-GWO-DVHOP gives the best performance when we vary the anchor ratio and the density of nodes.

Corresponding Author:

Omar Arroub
LRIT-GSCM Associated Unit to CNRST (URAC 29), FSR, Mohammed V-Agdal University
BP 1014, Rabat, Morocco
Email: omar_arroub@um5.ac.ma

1. INTRODUCTION

Nowadays, researchers have devoted a special interest to wireless sensor network (WSN) [1] due to its importance in several fields. Indeed, a sensor contains a battery for energy storage, antennas for receiving and sending information, memory, and GPS. However, using GPS doesn't guarantee a long sensor lifetime because the latter consumes more energy than other components. At this point, it was seen to use range-free techniques instead of ordinal techniques such as time of arrival (TOA) [2], time different of arrival (TDOA) [3], Angle of arrival [4]. Indeed, range-fee techniques don't depend entirely on using GPS. That's to say, we need just a few nodes to be equipped with GPS which are named beacons or anchors, and those latters help the target nodes to have their proper locations on the basis of the connectivity between nodes. It's worth mentioning that distance vector-Hop (DVHOP) localization algorithm [5], [6] is still the best technique among other range-free techniques because, experimentally, with a few anchors, DVHOP can locate the majority of nodes that constitute the network. We assume that several improvements have been made to ameliorate the precision of DVHOP. The problems that arise in those works are either they don't take the time calculation taken by the enhanced method into account or those improvements only take one way of distribution of nodes (random, uniform). In our approach, we have used two variants of GWO metaheuristic for the purpose of replacing the last step of DVHOP and reducing more localization errors of DVHOP. It's

worth mentioning that we have proven our improvements work in the uniform and random distribution of nodes for WSN.

GWO [7] is a robust metaheuristic that was proposed in 2014 by Mirjalili. It's worth mentioning that GWO has attracted researchers and scientists due to its simplicity and precise results, among other metaheuristics [8]. Also, we assume that GWO has the ability to offer a precise result for complex optimization problems, especially those that have high dimensions or have more than one local solution. In addition to that, GWO may serve resolution purposes for a discrete or continuous problem. In detail, this technique performs on the basis of the behavior adopted by grey wolves in the operations of searching and encircling the prey. In detail, the pack contains four individual classes. The individuals displace by respecting the hierarchy; the alpha is the leader who has the role of making the decision and guiding the whole pack, the beta assists the alpha and replaces the alpha in the case of death, the delta is responsible for hunting and attacking, the omega just follows the behavior of their leaders. The efficiency of the wolf to hunt well has given the algorithm GWO the reputation of good convergence and precise results. Also, this algorithm is easy to deploy because it has a minimum number of control parameters; it needs just two parameters, a and C. In addition to that, GWO guarantees an equilibrium between the exploration and exploitation phases. Hence, we find GWO is applied to many engineering problems that require optimization processes, especially the battery energy storage problem and the image-processing field. Its drawbacks reside in the lack of diversity in the population and its weakness in the exploitation phase.

In reality, each metaheuristic has its advantages and drawbacks. For example, PSO [9] can give a solution with a few control parameters, but its drawbacks reside in slow convergence and sticking to the local optimum for problems with high dimensions. Also, our GWO can provide a better solution than other MAs. Its disadvantages reside in the weakness in the exploitation phase, especially for the multi-modal problems that trap the algorithm in premature convergence, which leads to the apparition of several improvement works to overcome the issues that reside in GWO. For example, the apparition of opposition-based learning (OBL) [10], [11] consists of establishing the opposite positions of individuals not only at the beginning of the algorithm but also at each iteration. Also, we can apply dimension learning hunting (DLH) [12], [13] in order to increase the precision offered by GWO. In detail, DLH uses DLH learning, which consists of two phases: the initializing step: creating the neighbors of individuals based on certain rules and equations, and the updating phase: selecting or rejecting the new position created according to the fitness function. In the literature, OBL is still the most commonly used method to enhance GWO and there are many kinds of OBL that appear for that purpose, such as ROBL [14] based on luminous refraction, OBL based on the jump [15], which consist of creating an avoiding or a jump when we find a minimum local. In our contribution, we have used selective opposition-based learning because we found experimentally that SOBL suits our localization problem and effectively has the role of enhancing the precision of DVHOP localization algorithm.

The rest of the article is organized as follows: Firstly, we present different improved versions of DVHOP presented in literature. In section 3, we present the research method used to accomplish our work. In section 4, we define DVHOP, highlighting the source of its imprecision. Then, we present our ameliorated versions of the traditional DVHOP. Simulation and discussions are presented in section 5. We conclude the paper by presenting the conclusion and the perspective in section 6.

2. RELATED WORKS

Researchers and scientists have invented many meta-heuristics over the past few years. In our case of localization, we find that intelligent swarm-based algorithms are mostly used to improve DVHOP localization algorithm. That reflects that swarm-based algorithms may optimize several kinds of problems better than other cited meta-heuristics. Long *et al.* [16], it's shown that Amorphous gives better accuracy than DVHOP by varying the density of sensor nodes and anchor ratio. The experimental results prove that Amorphous localization algorithm offers a high level of accuracy with just a few anchors. That means Amorphous is a prominent solution for localization in WSN with minimal energy consumption. Ali *et al.* [17], the authors use a genetic algorithm (GA) for the purpose of ameliorating the precision of DVHOP. However, they also use PSO to ameliorate the crossover step of GA. The results obtained prove that DVHOP based on GA-PSO gives better accuracy than traditional DVHOP. Han *et al.* [18], the authors adopt PSO in order to optimize DVHOP. Also, they use LDIW [19] to create a balance between the diversification and test steps of PSO. However, the solution obtained by the modified PSO denotes the optimal location of nodes in WSN. Kessentini and Barchiesi [20] the authors try to improve the quality of convergence made by PSO-DVHOP. Indeed, they use QPSO for the purpose of reaching a global solution to the localizing problem. In general, the goal of QPSO-DVHOP is to ameliorate the accuracy of positioning. Additionally, QPSO-DVHOP performs with a good time of calculation in comparison with the traditional PSO-DVHOP. Zhang *et al.* [21], the authors present Centroid, APIT, DVHOP, and explain in detail the principle of functioning of

each cited localization algorithm. In addition to that, they present DVHOPmax which presents an enhanced version of DVHOP owned by maxhop parameter that's adjustable according to the topology of the network in order to reach a high level of accuracy. A performance comparison has been done in terms of the precision for the purpose of proving the efficiency of DVHOPmax.

3. RESEARCH METHODOLOGY

The research method is organized as follows: Firstly, we present the algorithm-based hop DVHOP highlighting the source of its huge errors in the localization process. Secondly, we interpret the last step of DVHOP as an optimization problem that is able to be resolved under GWO and two other improved GWO algorithms. Finally, our simulation is split into two steps. In the first phase, we set all the parameters of GWO and improved GWO algorithms properly for the regular deployment of nodes. Indeed, we set the number of nodes to 36, and we vary the percentage of nodes. Then, we keep the percentage of nodes at 30% and vary the number of nodes. We make such manipulations in order to compare the performance of our algorithms in uniform WSN. In the second phase, we re-set the parameters of our algorithms correctly by reducing the search space of our improved GWO algorithms for the random deployment. Then, repeat to handle the anchor ratio and the density of nodes as we do for uniform deployment. It's worth mentioning that in random deployment of nodes we reduced the surface area from 100×100 m^2 to 20×20 m^2 in order to narrow the search space of our metaheuristic.

4. PROPOSED NEW VERSIONS OF DVHOP

4.1. Traditional DVHOP localization algorithm

DVHOP is a range-free positioning technique; it belongs to a multi-hop localization algorithm, it makes positioning on the basis of the distance calculation. Also, this algorithm adopts an online method to calculate hop size. Contrary to other range-free techniques, this hop-based algorithm is known for its high coverage of localization. That means the cited localization algorithm is able to locate the majority of unknown sensors by using only a few anchor nodes. DVHOP takes the following steps to accomplish the localization process:

Step1: Flooding

A broadcast is done by anchors in order to assign each sensor by its number of jumps to their anchor.

Step2: Distance calculation

After the step of flooding, we calculate the size of the jump presented between anchors by using the following equation:

$$\text{hopsizei} = \frac{\sum_{j=1\ j\#i}^{n} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j=1}^{n} hopcount_{ij}} \quad (1)$$

where (x_i,y_i), (x_j,y_j) denote the positions of beacons i, j.

Then, we use (2) in order to calculate the distance that separates anchor i and the target sensor node.

$$d_{u,i} = hopsize_i \times hopcount_{u,I} \quad (2)$$

Step3: Determination of target node position

In this part, we calculate the positions of all target nodes. For that purpose, we use the least square method to specify the position of each unknown node.

(x,y) represents the position of the target nodes, (a_i,b_i) denotes the coordinates of the anchor node, where i = 1,2,..n and n is the number of beacons. Therefore, we can calculate the distance that separates unknown nodes and n anchors by using the non-linear equations.

$$\begin{cases} (x-a_1)^2 + (x-b_1)^2 + (z-c_1)^2 = d_1^2 \\ \vdots \\ (x-a_n)^2 + (x-b_n)^2 + (z-c_n)^2 = d_n^2 \end{cases} \quad (3)$$

Then we find:

$$\begin{cases} x^2 - 2a_1x + a_1^2 + y^2 - 2b_1y + b_1^2 = d_1^2 \\ \cdot \\ \cdot \\ x^2 - 2a_nx + a_n^2 + y^2 - 2b_ny + b_n^2 = d_1^2 \end{cases} \quad (4)$$

In (4) can be reformulated to,

$$\begin{cases} -2x(a_1 - a_n) + a_1^2 - a_n^2 - 2y(b_1 - b_n) + b_1^2 - b_n^2 = d_1^2 \\ \vdots \\ -2x(a_{n-1} - a_n) + a_{n-1}^2 - a_n^2 - 2y(b_{n-1} - b_n) + b_{n-1}^2 - b_n^2 = d_{n-1}^2 \end{cases} \quad (5)$$

The solution of the system may be interpreted as the resolution of the equation Ax=b,

$$A = \begin{bmatrix} 2(a_1 - a_n) & 2(b_1 - b_n) \\ \vdots & \vdots \\ 2(a_{n-1} - a_n) & 2(b_{n-1} - b_n) \end{bmatrix} \quad (6)$$

$$b = \begin{bmatrix} a_1^2 - a_n^2 + b_1^2 - b_n^2 + c_1^2 - c_n^2 - d_1^2 \\ \vdots \\ a_{n-1}^2 - a_n^2 + b_{n-1}^2 - b_n^2 + c_{n-1}^2 - c_n^2 - d_{n-1}^2 \end{bmatrix} \quad (7)$$

The main advantage of DVHOP is its simplicity of implementation. Also, it makes localization faster. In addition to that, it doesn't require many anchor nodes for offering a high coverage of localization. However, in DVHOP algorithm, it's assumed that the resolution method adopted by the algorithm causes an imprecision in estimating each position of the unknown node, which negatively impacts the accuracy of DVHOP. Additionally, DVHOP needs more anchors to offer the best accuracy. That's to say. DVHOP performs with high energy consumption. Hence, DVHOP requires amelioration in purpose to mitigate its issues.

In our approach, we replace the least squares method with an optimization problem. It's noted that we keep the two first steps of DVHOP as they are without making any changes to them because those steps allow DVHOP to make localization with good coverage of. In detail, our improved method is described as follows: Firstly, we calculate the number of hops between beacons and target nodes. In the second step, we calculate the distance between beacons and target nodes on the basis of the size-hop. Finally, a specified metaheuristic is applied in order to minimize the function described below:

$$f(x,y) = \frac{1}{n}\sum_{i=1}^{n} \left|\sqrt{(x - a_i)^2 + (y - b_i)^2} - d_i\right| \quad (8)$$

Where n denotes the number of beacons, (a_i,b_i) are the positions of beacons, d_i represents the distance that separates the unknown node and anchor *i*.

4.2. DVHOP algorithm-based GWO

The GWO technique is an optimization tool based on a swarm that can imitate the hierarchy and hunting model of grey wolves. The GWO algorithms consider only three individuals named α, β, and δ. In addition to that, the best solution is considered as α wolves. The technique adopted by the wolves in chasing uses three steps: encircling, hunting, and attacking. Indeed, the pack of wolves follows the described organization:

The alpha ones are the leaders of the pack. Their role is to make decisions about hunting, location of sleep, wake-up time, and so on. The rest of the pack should follow the alpha's judgment. The alpha category is the controlling wolf because their orders must be respected by the individuals of the pack. The alpha ones are not allowed in mating operations. It's worth mentioning that it's not conditioned that the alpha be the strongest member of the pack. However, the alpha ones are the best individuals to manage and guide the whole pack. That reflects that the hierarchy of wolves is based upon discipline and good management, not strength.

The beta wolves present the second class of the hierarchy; they assist the alpha to make decisions and to do other tasks. Also, they may be male or female, and they replace the alpha when alpha wolves are dead or would be weak (very old). The beta wolves should obey the orders of alphas, but the beta ones command the delta ones and omegas (the lowest level of the hierarchy). Delta wolves must respect alphas and betas. However, they make orders to the omegas.

Omega wolves represent the lowest ranking of the pack. On the other hand, omega should respect the orders of other controlling wolves. It could appear that omega wolves aren't important members of the pack, but it seems that the pack has trouble, such as fighting between members, when we lose the omega

wolves. That reflects the role of the omegas to maintain peace between the other members. Also, the omega ones may be the baby sisters of the pack.

In the following, we explain mathematically all the steps taken by wolves before attacking their prey. Briefly, we summarize those steps to encircling, hunting and attacking the prey that are described as follows:

4.2.1. Encircling the prey

After the initialization step, the wolves encircle the prey. We illustrate the mathematical model of the encircling operation by the following equations:

$$D = |C*Xp(t) - X(t)| \quad (9)$$

$$X(t+1) = Xp(t) - A*D \quad (10)$$

where: A and C are controlling parameters, X_p indicates the prey location, X is the grey wolf's coordinates. We calculate A and C on the basis of the following equations:

$$A = 2*a*r1 - a \quad (11)$$

$$C = 2*r2 \quad (12)$$

where: A decrease in a linear manner from 2 to 0 , We assume that r1 , r2 are values between 0 and 1.

4.2.2. Hunting the prey

We suppose that alpha, beta, delta individuals know the prey coordinates. However, the omega wolves update the positions according to the positions of alpha, beta, delta. We illustrate the mathematical model of hunting operations by the following equations:

$$D\alpha = | C1*X\alpha - X| \quad (13)$$

$$D\beta = | C2*X\beta - X| \quad (14)$$

$$D\delta = | C3*X\delta - X| \quad (15)$$

$$X1 = X\alpha - A1*D\alpha \quad (16)$$

$$X2 = X\beta - A2*D\beta \quad (17)$$

$$X3 = X\delta - A3*D\delta \quad (18)$$

$$X(t+1) = (X1 + X2 + X3)/3 \quad (19)$$

4.2.3. Exploitation/attacking

When we execute encircling and hunting prey, we pass to attacking the prey. This behavior is mathematically formulated by the linear decrease of the value of a from 2 to 0. Also, we note that the position of the wolf in this step is taken randomly throughout the position of the prey. It's worth mentioning that during this stage, the value of |A| is kept less than 1.

4.2.4. Summarize

The search operation begins by creating a population of individuals (chosen randomly). During the iterations of the algorithm, the different wolves try to know the position of prey. That's to say, each of the categorized individuals mentioned makes a change to its distance from the prey in order to create an equilibrium between the diversification and test phases. It's noted that the termination of the method is conditioned by the specified number of iterations. Algorithm 1 describes the different steps of GWO.

Algorithm 1: Algorithm of GWO

```
Initialize the grey wolf population Xi=(i=1,……,n)
Initialize a, A and C
Calculate the fitness of different agents
Xα, Xβ, Xδ are the solutions
while ( t < Max_nbre_iterations)
 for each agent
```

```
 make a change to the current position by using (19)
 end for
 modification of a, A and C
 find fitness of the agents
 update Xα, Xβ, Xδ
 t=t+1
end while
return Xα
```

In the beginning, we generate hop-size and the distance targets and anchors using the first two steps of DVHOP. Then we select a population of wolves generated randomly and presented in the search space. Also, we use the cost function to evaluate the wolves's position. Then we repeat the following steps until we reach the specified number of iterations (max_iter). Indeed, in each step, we consider just leader wolves because their positions minimize the cost function. Then, each individual makes a change to its location on the basis of the attacking steps followed by wolves. Finally, we repeat those steps until we reach the α position that presents the best position and denotes the best coordinates of the target node. Figure 1 shows in detail the different steps of the GWO-DVHOP algorithm,

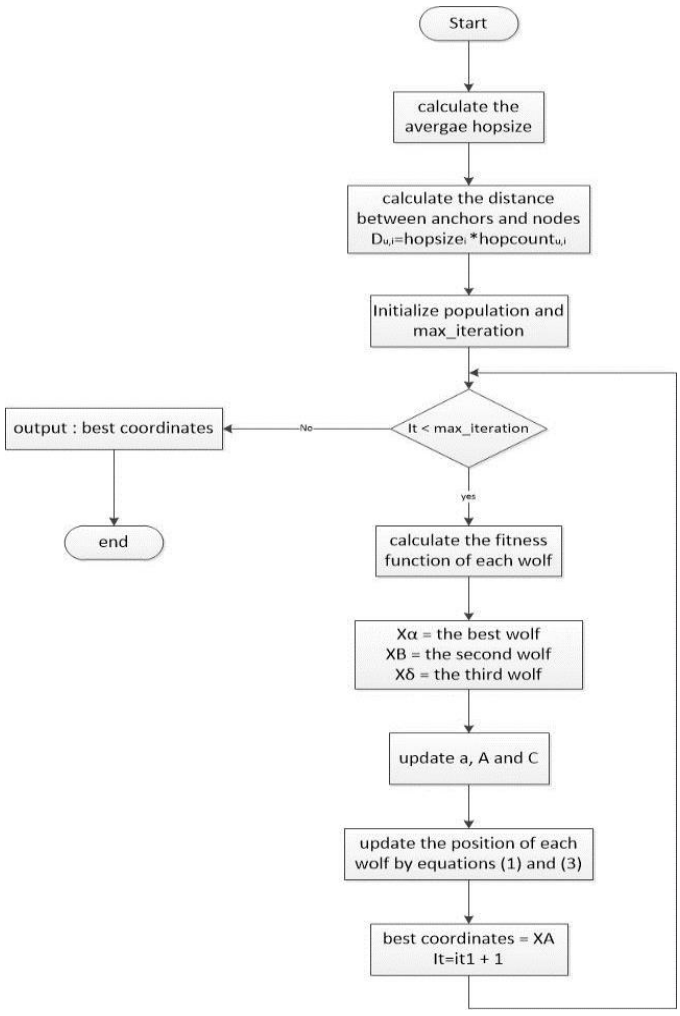


Figure 1. Flowchart of GWO-DVHOP algorithm

4.3. DVHOP algorithm based DLH-GWO

In fact, the traditional GWO doesn't have diversity in the population and has an imbalance between diversification and test steps. As a consequence, GWO makes optimization with trapping in the local minimum. In order to overcome those issues, a new DLH-GWO is presented. The improved algorithm enhances the hunting strategy of search agents by using DLH. This method serves to enlarge the search space of individuals by using DLH learning. Furthermore, DLH-GWO creates two wolves through DLH learning

and the strategy adopted by GWO for updating the position. Additionally, DLH-GWO adds an updating step for selecting the best wolf. This process is repeated at each iteration. Briefly, we summarize the DLH-GWO into two steps: the initializing phase and the updating phase.

Initializing phase: In this step, we consider N individuals that are chosen randomly throughout the space of searching. Also, we consider the search space equal to [l_p u_j]. The initial position of the wolf is described by the (20).

$$Xi,j = lj + randj[0,1] * (uj – lj) , i \in [1,N], j \in [1,D] \quad (20)$$

where D denotes the dimension of the problem.

DLH-GWO add DLH learning that consists of the following idea: each wolf calculates its position according to its learning by its neighbors, then we calculate the position taken by the wolf using a canonical GWO search strategy and adopting the (21).

$$Xi_{GWO(t+1)} = \frac{Xi1(t)+Xi2(t)+Xi3(t)}{3} \quad (21)$$

DLH search strategy makes the calculation of another wolf's new position named $X_{i\text{-}DLH}$. To accomplish that, another radius calculation is made. In detail, the radius $R_i(t)$ is the distance presented between the positions $X_i(t)$ and $X_{i\text{-}GWO}(t+1)$. Mathematically, $R_i(t)$ is calculated by using equation (22),

$$R_i(t) = \| X_i(t) – X_{i\text{-}GWO}(t+1)\| \quad (22)$$

then, we calculate mathematically the neighbors of $X_i(t)$ by using the (23),

$$N_i(t) = \{X_j(t) \mid D_i(X_i(t) , X_j(t)) \leq R_i(t) , X_j(t) \in Pop\} \quad (23)$$

where D_i is the distance that separates the position $X_i(t)$ and the position $X_j(t)$.

After calculating all the possible neighbors of $X_i(t)$, we execute multi-neighbor learning using the (24):

$$X_{i\text{-}DLH,d}(t+1) = X_{i,d}(t) + rand * (X_{n,d}(t) – X_{r,d}(t)) \quad (24)$$

where X_{i_DLH} is calculated according to the neighbor $X_{n,d}(t)$ chosen from $N_i(t)$ and a random individual $X_{r,d}(t)$.

Updating phase: In this step, we make a comparison between the fitness of $X_{i\text{-}GWO}(t+1)$ and $X_{i\text{-}DLH}(t+1)$ aiming to select the good solution. The (25) describes the updating step.

$$X_i(t+1) = \begin{cases} X_{i-GWO}(t+1), if\ f(X_{i-DLH}) > f(X_{i-GWO}) \\ X_{i-DLH}(t+1), if\ f(X_{i-DLH}) \leq f(X_{i-GWO}) \end{cases} \quad (25)$$

After executing this process for all individuals, we increase the number of iterations by one. That's to say, we stop the search operation when the condition of ternination is satisfied. Figure 2 illustrates the functioning of DLH-GWO.

We prepare a surface of capting, where we properly set the number of nodes, and we also designate the nodes that denote the anchors. Then, we serve by DLH-DVHOP algorithm to locate precisely the unknown nodes. Algorithm 2 describes the process of DLH-GWO-DVHOP.

Algorithm 2: pseudo-code of DLH-GWO-DVHOP

```
Initialization:
number of nodes =NB,
number of anchors=NA,
area surface =1000×1000m2 ,
radio radius=500m
1.calculation of hopcounti,j by using the shortest path
2.calculate the size of hop by using (1)
3.target positions calculation
for i=NA to NB
4. calculate required distance
unknown_to_anchrs_dist=hopsize(i) × shortest_path(i,1 to NA);
5.calculate fitness function f by using (8)
6.execute DLH-GWO algorithm
initializing parameters: A, C, a , t=0 , max_iter =500 and population size N = 30 ,
for t=1 to max_iter
select Xα, Xβ, Xδ
for i=1 to N
Xi,1 = Xα(t) – Ai,1* Dα(t)
Xi,2 = Xβ (t) – Ai,2* Dβ (t)
Xi,3 = Xδ (t) – Ai,3* Dδ (t)
```

```
X (t+1) = (Xi,1(t)+ Xi,2(t)+ Xi,3(t))/3
Ri(t) = || Xi(t) – Xi-GWO(t+1)||
calculate the neighbors of Xi(t)
NLi (t) = {Xi(t)| Di(Xi(t), Xj(t)) ≤ Ri(t), Xj(t) ∈ pop}
for d =1 to D
Xi_DLH (t) = X(i,d)(t) + rand( X(n,d)(t) – X(r,d)(t))
end for
if(fitness (Xi_GWO(t+1) < fitness(Xi_DLH(t+1))
Xi(t+1) = Xi_GWO(t+1)
else
Xi(t+1) = Xi_DLH(t+1)
update population
end for
end for
return global solution
6.assign the result of DLH-GWO to target node
node.estimated(i,1to 2)= global solution;
```

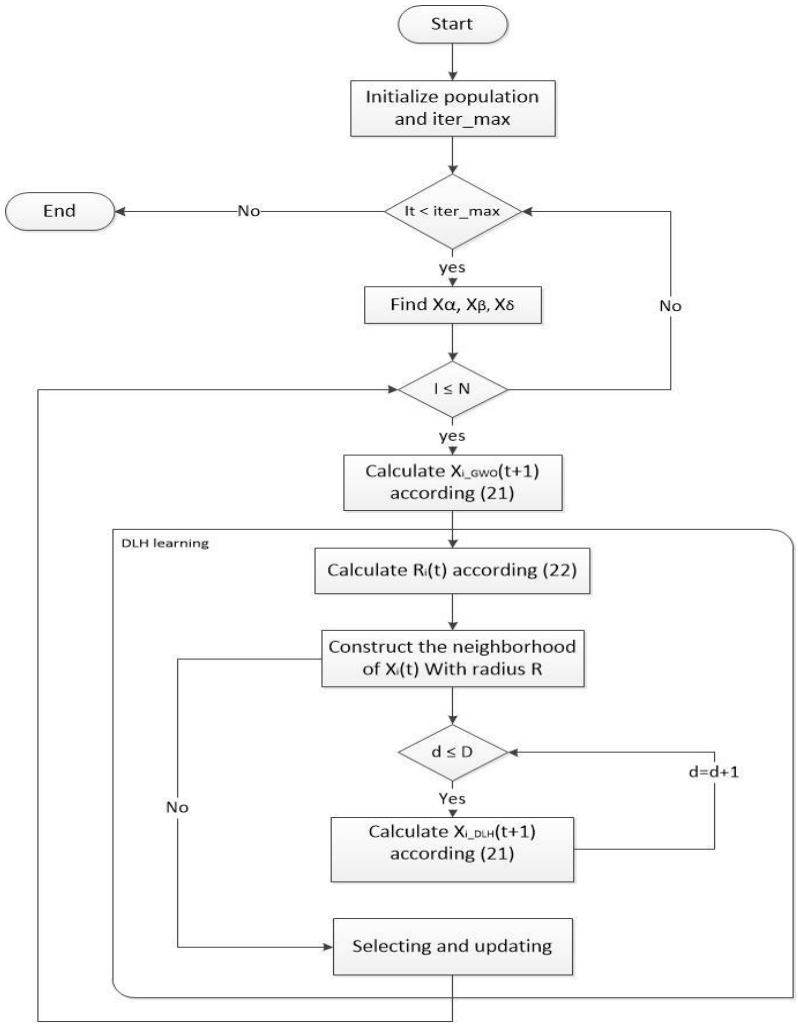


Figure 2. Flowchart of DLH-GWO algorithm

4.4. SO-GWO- DVHOP algorithm

In our improved version of GWO, the new integrated parameters help us to realize an equilibrium between exploration and exploitation. In detail, we develop a decent method of calculation in order to enhance the exploration phase adopted by the algorithm, and we reach that by using opposition-based learning (OBL). However, OBL helps GWO converge faster to the global optimum. The process is as follows: In each iteration, we select candidate solutions using Spearmn's rank Correlation Coefficient in order to avoid unnecessary exploration, which will also serve to ensure fast convergence. On the other hand,

if ω wolves in the inverse direction, then d has a negative correlation. This relation will serve us to determine the dimensions and the bounds of search of α, where d denotes the threshold, and it will vary linearly as we advance in iterations. The disparity between positions X and Xα is calculated by (26):

$$diff(j) = | X(j) - X\alpha(j) | \quad (26)$$

where: j denotes the dimension, g is determined on the basis of the value of diff(j). Indeed, if diff(j) > threshold, then $g_{n0} = g_{n0} - 1$. Also, it's noted that if $(n - g_{n0}) < g_{n0}$ and diff(j) > threshold, then X'(i) is calculated by (27),

$$X'(i) = ub(i) + lb(i) - X(i) \quad (27)$$

where: i∈ {j: diff(j) ≤ threshold} and X'(i) denotes the inversed vector. ub(i) and lb(i) denote the upper and lower bounds.

In the traditional GWO, we aim to calculate the positions of individuals according to the locations of α, β, and δ individuals. Indeed, if the leaders get trapped at the local minimum, the rest of the population also has the vulnerability to be trapped in local extremes. To address this issue. The new configuration adopted in GWO tackles this problem by combining OBL with GWO. Indeed, spearman's rank correlation coefficient is exploited to decide the omega wolves on which to apply OBL. This helps us avoid useless exploration and expands the search space. Hence, by adopting SO-GWO we realize an equilibrium between exploration and exploitation. Additionally, SO-GWO reduces the time of calculation without causing any mistakes in finding the global solution. Figure 3 shows in detail the steps of SO-GWO.

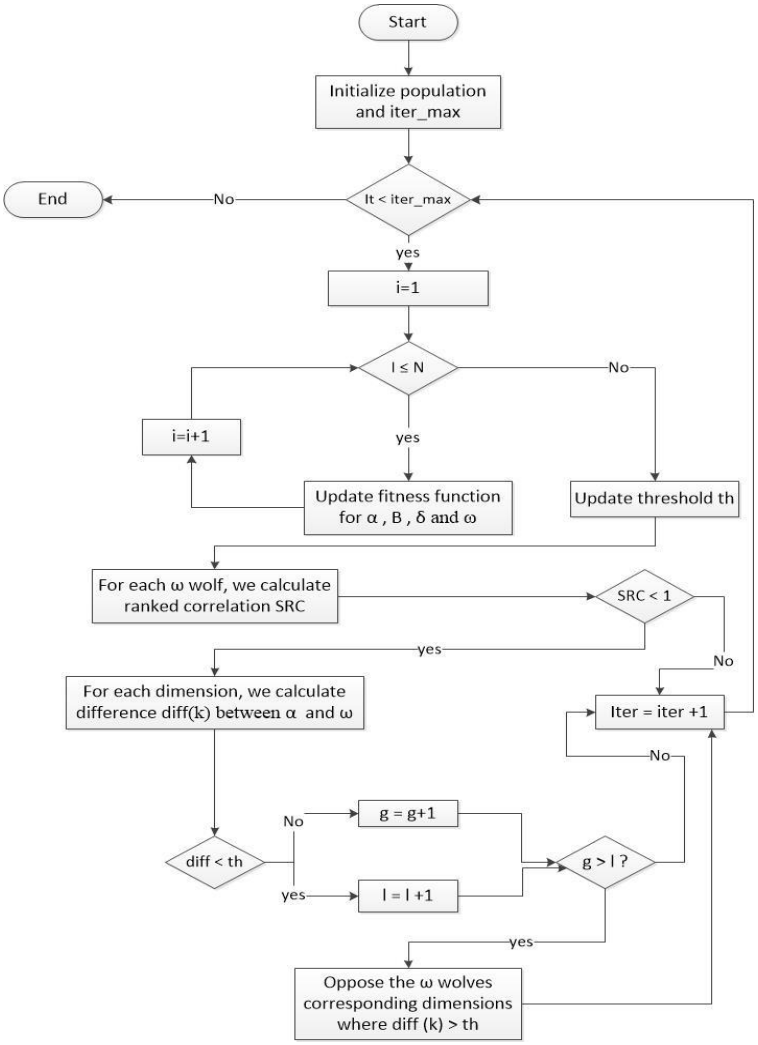


Figure 3. Flowchart of SO-GWO algorithm

SO-GWO-DVHOP consists of the following descriptions: We make selective opposition to the pack of wolves inside the browsing of unknown nodes. That's to say, in each iteration, we assign to an unknown node the value returned by SO-GWO according to the number of search agents, the maximal number of iterations, the value of control parameter a, the upper and lower bounds, and the distance that separates anchors and the unknown node generated by the second step of DVHOP. It's noted that the population vector is initialized randomly. Algorithm 3 describes in detail the steps followed by SO-GWO-DVHOP.

Algorithm 3: pseudo-code of SO-GWO-DVHOP

```
1.Assign for each unknown node the value returnd by SO-GWO
for i=1 to total number of unknown nodes
2.Generate the distance between unknown node and anchors according the two first steps of
DVHOP
3.Execute SO-GWO algorithm
Initialize n search agent positions, α, β and δ positions, t
while (t < max_iterations)
for i=1 to n
reinitialize the wolf dimensions
for each search agent, we calculate fitness function
end for
sort the search agents and update α, β and δ positions
a=2- [iter* (2/max_iterations)]
threshold=a
for i=1 to number_ω_search_agents
for j=1 to nbre_dimensions
diff(j)=|X(j) - X_α(j)|
if diff(j)> thresh
g_no=g_no+1
end if
end for
src = 1 - ∑j(diff(j))²/[dim*(dim²-1)]
if src < = 0
if(dim-g_n0<g_n0)
for k ∈ {j : diff(j) > thresh}
X(k) = lb(k) + ub(k) - X(k)
end for
end if
end if
end for
make change to the position of each agent by using equation (19)
t=t+1
end while
return position of α
4.Assign the result of SO-GWO to a target node
node.estimated(i,1to 2)=position of α ;
end for
```

5. SIMULATIONS AND RESULTS

In this part, we make a performance comparison of SO-GWO-DVHOP, DLH-GWO-DVHOP, GWO-DVHOP and DVHOP in terms of precision. It's noted that we can judge the quality of the localization on the basis of several metrics, such as energy consumption and coverage of localization. In reality, DVHOP has a significant success in locating the whole nodes of the network with a minimum of anchors, and we confirmed that in several scenarios of simulation by varying other metrics such as radio radius, percentage of anchors. Consequently, we don't consider assessment in terms of coverage, and we evaluate our methods just on the basis of their precision of localization in a network with regular and random distribution of sensor nodes. The metric of comparison is average localization error (ALE). Obviously, we make such simulations in order to choose the most performant algorithm with a specified configuration. The work that has already been done in the field of localization presented in WSN did not give a high accuracy of localization [21]–[25]. It's worth mentioning that the works that have already been done in the field of positioning did not give a high accuracy of localization [22]-[25]. The parameter settings of SO-GWO-DVHOP, DLH-GWO-DVHOP, GWO-DVHOP are listed in Table 1, summarizing the pertinent variables used in our assessment.

We use ALE in order to assess the quality of each cited algorithm in terms of precision. ALE denotes the ratio of localization error to the total number of nodes. However, ALE is used to determine the accuracy of each positioning technique according to the total number of nodes, the anchor ratio, and the topology of the nodes's distribution. Indeed, we say that a technique is less precise if it has high ALE. We calculate ALE using (28):

$$ALE = \frac{\sqrt{(x_t - x_e)^2 + (y_t - y_e)^2}}{(n_t - n_h)r} \quad (28)$$

Where (x_t, y_t) represents the true location of nodes
Where (x_e, y_e) represents the estimated location of nodes
n_t is the number of nodes
n_h is the non-positioned nodes
r denotes the radio radius of a sensor node

Table 1. Parameter's settings of GWO-DVHOP, DLH-GWO-DVHAP and SO-GWO-DVHOP

Parameter	Value
Dimension	2
Lower bound	0
Upper bound	100 in uniform deployment of nodes 20 in random deployment of nodes
Number of iterations	500
a	Linearly decrease from 2 to 0
Number of population	30
Cost function	Cost function of DVHOP

5.1. Simulation results

To compare our algorithms in terms of accuracy, we split the simulation scenario into two phases. We first compare GWO-DVHOP, DLH-GWO-DVHAP, and SO-GWO-DVHOP in a WSN with a uniform distribution of nodes, and we vary the anchor ratio and the total number of nodes. Secondly, another comparison is made by keeping the same set of algorithms and the same metrics. The only difference is that we change the distribution of nodes to a non-regular distribution. The parameter settings used in our simulation are summarized in Table 2. We assume that we use an area whose surface equals $100 \times 100 m^2$.

The strategy followed in our simulation consists of two phases. Firstly, we keep the total number of nodes at a value of 36 and the radio radius at 34 m. Then, we change the anchor ratio between the values of 5% and 30%. In the second phase, we vary the number of nodes by keeping the beacon ratio at 20% and the radio radius at 34m. Figure 4 illustrates the initial field of sensing. Indeed, the number of nodes is 16 nodes (3 anchors and 13 unknown nodes). In Figure 4(a), we use a regular topology. In Figure 4(b), we change the network topology to a random distribution. Also, we change the surface area from the value of 100×100 m^2 to 20×20 m^2 on purpose to decrease the search space of GWO algorithm and its variants.

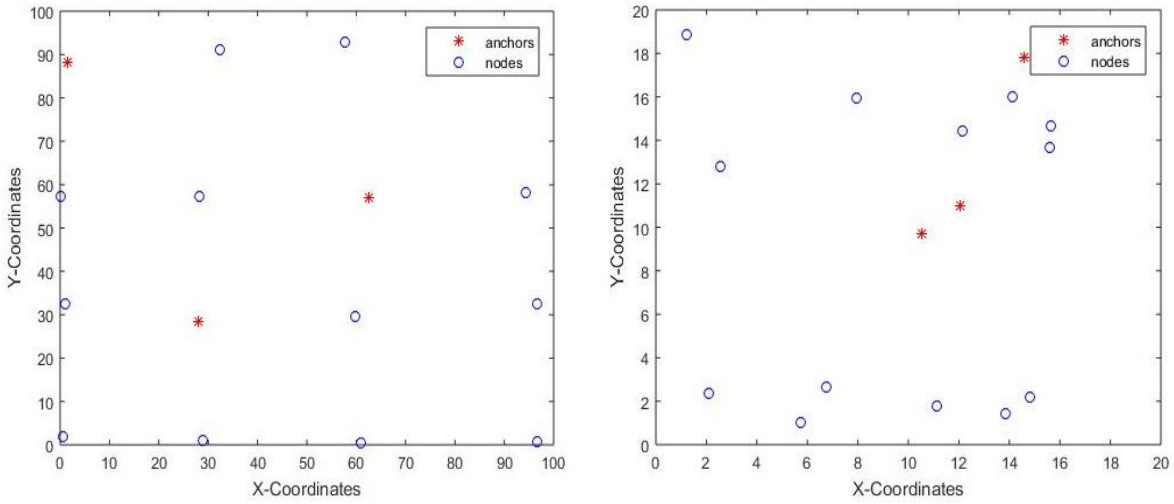
Table 2. Parameter settings of simulations

Parameter	Value
Area	100×100 m^2 in grid topology 20×20 in random topology
Total number of nodes	16, 25, 36, 49, 64, 81
Distribution of nodes	Uniform Random
Anchor rachio (%)	5 - 10 - 15 - 20 - 25 - 30
radio radius	34 m, [100/(number_of_nodes_by_side-1)]*2 (29)
Model of communication	Regular

5.2. Discussions

5.2.1. The comparison under a regular distribution

In this part, we compare the algorithms SO-GWO-DVHOP, DLH-GWO-DVHOP, GWO-DVHOP and DVHOP in terms of precision by changing the anchor ratio. It's worth mentioning that the total number of nodes is 36 and the radio radius is set to 34 m. Figure 5 shows the variation of ALE according to the anchor ratio. According to Figure 5, it's clear that DVHOP gives the worst results because there's an accumulation of errors during its last step, which leads to huge errors in estimating the position of each node by that algorithm. Also, it's noted that the precision of each method augments when we augment the anchor ratio. However, SO-GWO-DVHOP, DLH-GWO-DVHOP show better performance due to the efficiency of both SO-GWO and DLH-GWO for optimization purposes. Indeed, the two meta-heuristics aim to diversify the population of wolves. Hence, they could improve the precision of DVHOP better than the traditional GWO. Also, it can be seen that GWO has successfully enhanced the precision of DVHOP because GWO is based on the hunting hierarchy of individuals. Also, it's shown its ability to minimize our cost function despite its complexity and multi-modality. However, the results offered by GWO-DVHOP are still less precise than those offered by DLH-GWO-DVHOP and SO-DLH-DVHOP.



(a) (b)

Figure 4. Initial distribution (a) uniform and (b) random

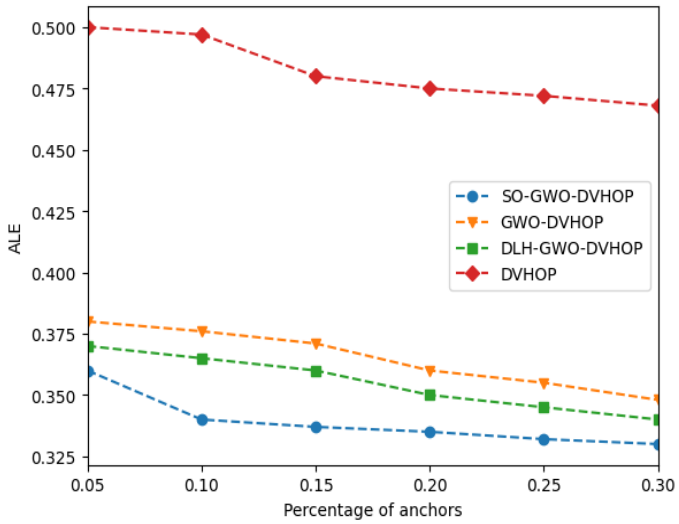


Figure 5. Variation of average location error of algorithms with communication range =34m

In Figure 6, we vary the total number of nodes between 16 nodes and 81 nodes. Also, we vary the communication range according to (29), and we keep the percentage of anchors at the value of 20%. According to the results, we can observe that the performance of SO-GWO-DVHOP is outstanding compared to other algorithms, and the accuracy offered by that algorithm is less than the others. We first explain the high precision of SO-GWO-DVHOP, DLH-GWO-DVHOP by the efficiency of the resolution methods adopted and their suitability to our cost function. Secondly, when we increase the total number of nodes, we also increase the connectivity between nodes because we keep the same area size. That contributes positively to generating input parameters for our meta-heuristics. Hence, SO-GWO-DVHOP and DLH-GWO-DVHOP could locate more precisely the unknown nodes and give a high level of precision compared to GWO-DVHOP and DVHOP. We also observe that GWO-DVHOP has shown good outcomes, and its results are close to those of the improved version of GWO-DVHOP. Finally, DVHOP takes the worst rank, and its results are far from the results of the mentioned algorithms. We also note the non-stability of its results when we augment the total number of nodes.

In Tables 3 and 4, we show maximum and minimum ALE of our algorithms. It's shown that SO-GWO-DVHOP has the best performance in comparison to other localization methods in a network with regular deployment of nodes.

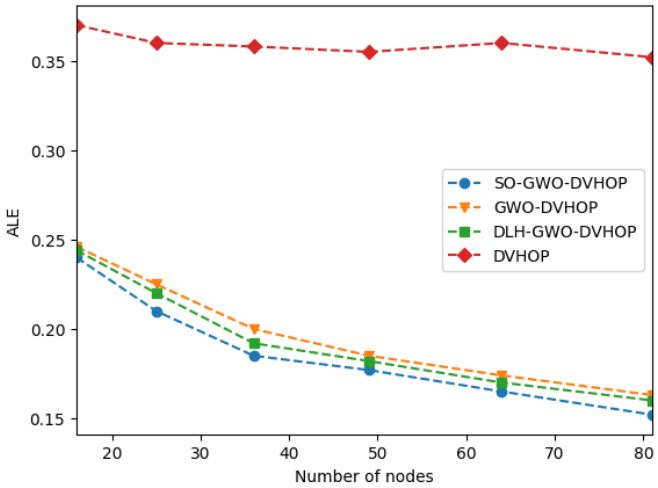


Figure 6. Variation of average location error of algorithms

Table 3. Localization error comparison in uniform distribution of nodes (variation of the ratio of anchors)

Localization algorithm	Max. average localization error (M)	Min. average localization error (M)	Mean. average localization error (M)
DVHOP	0.500	0.470	0.485
GWO-DVHOP	0.382	0.350	0.366
DLH-GWO-DVHOP	0.370	0.340	0.355
SO-GWO-DVHOP	0.360	0.335	0.347

Table 4. Localization error comparison in uniform distribution of nodes (variation of the number of nodes)

Localization algorithm	Max. average localization error (M)	Min. average localization error (M)	Avg. average localization error (M)
DVHOP	0.365	0.350	0.358
GWO-DVHOP	0.248	0.175	0.211
DLH-GWO-DVHOP	0.246	0.170	0.208
SO-GWO-DVHOP	0.240	0.152	0.196

5.2.2. Performance comparison with random distribution

In this part, we compare SO-GWO-DVHOP, DLH-GWO-DVHOP, GWO-DVHOP, DVHOP by changing the anchor ratio. Also, we note that the value of the radio radius is fixed at 34m and the number of nodes is set at 36. Figure 7 shows the variation of ALE of our algorithms according to the anchor ratio.

According to the results, we observe that SO-GWO-DVHOP gives the best result that reflects the efficiency of SOBL to create a diversity of population and expands positively the search space aiming to avoid local. Also, it's observed that when we augment the anchor ratio, the ALE of the improved method still decreases because, when we increase the total number of anchors, we also augment the precision of the distance between anchors and target nodes. Hence, that serves the algorithm to estimate precisely the target. Additionally, it's clear that DLH-GWO-DVHOP and GWO-DVHOP give approximately the same result, but their performances are still less than those offered by SO-GWO-DVHOP. However, OBL consists of selecting some individuals in the opposite direction in order to avoid that omega wolves follow their leaders mistakenly, so SO-GWO has succeeded in diversifying the population more efficiently than GWO and its variant DLH-GWO. Consequently, the performance of SO-GWO-DVHOP is greater than that done by DLH-GWO-DVHOP and GWO-DVHOP. Finally, we remark that DVHOP shows the worst outcomes, and it's sensitive enough to add more anchors to offer a slight increase in its accuracy.

In the configuration shown below, we change the total number of nodes between 16 and 81. Also, we change the value of communication range according to (29) and keep the percentage of anchors at 20%. The results of the experiments are shown in Figure 8.

According to Figure 8, a non-stability in DVHOP calculation is observed. That reflects the non-stability of the resolution method adopted by DVHOP. Indeed, in non-uniform deployment of nodes, we are facing the huge imprecision of averaging hop-size made by the algorithm that leads DVHOP to locate the whole nodes of the network imprecisely. Also, we observe that the error ratio of DVHOP is superior to the ratio of other improved algorithms. On the other hand, it's noted that SO-GWO-DVHOP is the most precise

algorithm, indicating that Selective-OBL has significantly enhanced the quality of searching made by GWO. Hence, the resulting algorithm has successfully optimized the cost function of localization. Although DLH learning is devoted to enhancing GWO and the performance of DLH-GWO-DVHOP is greater than that of GWO-DVHOP as shown in Figure 8. Nevertheless, this offered performance is still less than that of SO-GWO-DVHOP. We can conclude that in our case of localization and the form of our cost function, SO-OBL is the most appropriate solution to optimize GWO. Consequently, SO-GWO still the most convenient choice to optimize DVHOP.

In Table 5 and Table 6, we show maximum, minimum average localization error of our algorithms. It's shown that SO-GWO-DVHOP has the best performance in comparison to other localization algorithms when we vary the total number of nodes and the percentage of anchors in network with non-regular distribution of nodes. In Table 7 we present the relevant work that has already been done to optimize DVHOP in terms of precision.

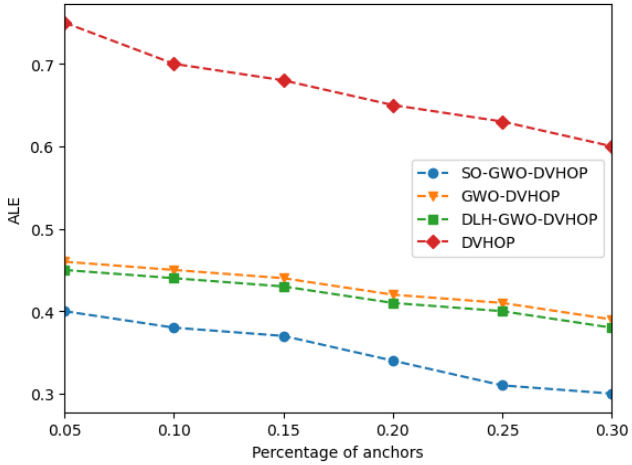


Figure 7. Variation of average location error of algorithms with communication range =34m

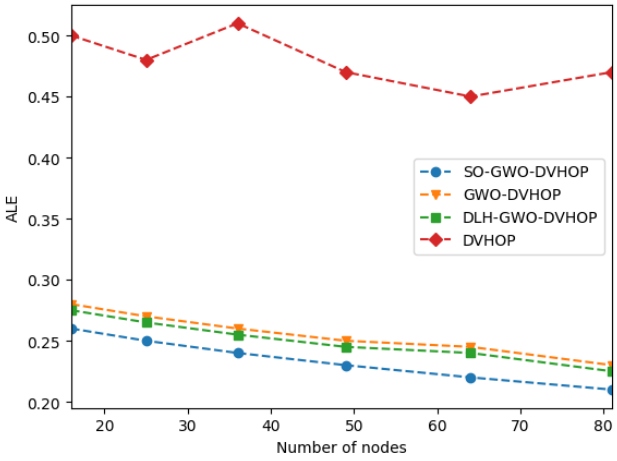


Figure 8. Variation of average location error of algorithms

Table 5. Localization error comparison in random distribution of nodes (variation of the ratio of anchors)

Localization algorithm	Max. average localization error (M)	Min. average localization error (M)	Avg. average localization error (M)
DVHOP	0.640	0.600	0.620
GWO-DVHOP	0.460	0.400	0.430
DLH-GWO-DVHOP	0.450	0.390	0.420
SO-GWO-DVHOP	0.400	0.300	0.350

Table 6. Localization error comparison in random distribution of nodes (variation of the number of nodes)

Localization algorithm	Max. average localization error (M)	Max. average localization error (M)	Max. average localization error (M)
DVHOP	0.520	0.450	0.485
GWO-DVHOP	0.280	0.240	0.260
DLH-GWO-DVHOP	0.270	0.230	0.250
SO-GWO-DVHOP	0.260	0.210	0.235

Table 7. Summarizes the relevant works

Researchers	Algorithm	Network settings	Metric	Value
Sharma and Kumar [23]	DVHOP with Genetic Algorithm	Random deployment of nodes with variation of percentage of anchors	Average Localization error (m)	2.48
		Random deployment of nodes by varying the number of nodes	Average Localization error (m)	2.86
Messous *et al*. [24]	DVHOP with polynomial approximation	Random deployment of nodes by varying the percentage of anchors	Average Localization error (m)	2.58
		Random deployment of nodes by varying the communication range	Average Localization error (m)	2.83
Cheng *et al*. [25]	DVHOP with Archimedes algorithm	Random deployment of nodes by varying the number of nodes	Average Localization error (m)	0.30
		Random deployment by varying the percentage of anchors	Average Localization error (m)	0.37
Xue 2019 [19]	DVHOP with PSO	Random deployment by varying the percentage of anchors	Average Localization error (m)	0.39
		Random deployment of nodes by varying the communication range	Average Localization error (m)	0.39
Zhang *et al*. [21]	DVHOP with Quantum-Behaved Particle Swarm Optimization	Uniform distribution of nodes by varying the communication range	Average Localization error (m)	0.21
		Uniform distribution of nodes by varying the number of anchors	Average Localization error (m)	3.68
		Uniform distribution of nodes by varying the node density	Average Localization error (m)	0.18

6. CONCLUSION

DVHOP is our target for improvement, to reach that we present a combined SO-GWO-DVHOP. The main idea of SO is to enhance the opposite learning process by using Spearmn's rank Correlation Coefficient. This strategy serves to expand the search space of wolves. Also, it creates a variety of populations and can create an equilibrium between exploration and exploitation. Additionally, we use the first two steps of DVHOP for generating the distance that separate the target node from anchors which will denote input parameters for our improved GWO algorithm. On the other hand, the coordinates of nodes in WSN will be calculated iteratively by adopting the resultant algorithm SO-GWO-DVHOP. Our simulation consists of comparing DVHOP with its enhanced versions. The results obtained confirm that SO-GWO-DVHOP gives the best precision of localization in a network with both regular and non-random distribution of nodes compared to DVHOP and their other variants GWO-DVHOP and DLH-GWO-DVHOP.

Although SO-GWO-DVHOP has shown superior efficiency against other algorithms. However, there are some obstacles, such as noise and multipath, that occur in communication between nodes, and those phenomena reduce the localization accuracy of algorithms. As a consequence, we suggest making another deep study aiming to enhance SO-GWO-DVHOP in a network with regular and non-regular deployment of nodes by taking the presence of noise effect into account.

FUNDING INFORMATION

The authors state no funding is involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Omar Arroub	✓	✓	✓			✓	✓	✓	✓	✓			✓	✓
Aouar Darif		✓		✓	✓	✓			✓					
Rachid Saadane				✓										
My Driss Rahmani				✓							✓	✓		
Zineb Aarab									✓	✓				

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

To Authors state no conflict of interest.

DATA AVAILABILITY

Simulation data about this study are available from the corresponding author.

REFERENCES

[1] D. Kandris, C. Nakas, D. Vomvas, and G. Koulouras, "Applications of wireless sensor networks: An up-to-date survey," *Applied System Innovation*, vol. 3, no. 1, pp. 1–24, 2020, doi: 10.3390/asi3010014.

[2] T. E. Oliveira, J. R. Reis, and R. F. S. Caldeirinha, "Implementation of a WSN for environmental monitoring: From the base station to the small sensor node," *Sensors*, vol. 22, no. 20, p. 7976, 2022.

[3] K. Yu, Y. J. Guo, and M. Hedley, "TOA-based distributed localisation with unknown internal delays and clock frequency offsets in wireless sensor networks," *IET Signal Processing*, vol. 3, no. 2, pp. 106–118, 2009, doi: 10.1049/iet-spr:20080029.

[4] W. Li and B. Zhao, "Analysis of TDOA location algorithm based on ultra-wideband," *Proc. of International Conference in Communications, Signal Processing, and Systems*, pp. 1257–1261, 2020.

[5] H. Li and Z. Cheng, "Angle-of-arrival estimation using difference beams in localized hybrid arrays," *Sensors*, vol. 21, no. 5, pp. 1–11, 2021, doi: 10.3390/s21051901.

[6] W. Liu, J. Li, A. Zheng, Z. Zheng, X. Jiang, and S. Zhang, "DV-Hop algorithm based on multi-objective salp swarm algorithm optimization," *Sensors*, vol. 23, no. 7, 2023, doi: 10.3390/s23073698.

[7] S. Messous and H. Liouane, "Online sequential DV-hop localization algorithm for wireless sensor networks," *Mobile Information Systems*, vol. 2020, 2020, doi: 10.1155/2020/8195309.

[8] W. Gai, C. Qu, J. Liu, and J. Zhang, "A novel hybrid meta-heuristic algorithm for optimization problems," *Systems Science and Control Engineering*, vol. 6, no. 3, pp. 64–73, 2018, doi: 10.1080/21642583.2018.1531359.

[9] S. Rajendran, N. Ganesh, R. Čep, R. C. Narayanan, S. Pal, and K. Kalita, "A conceptual comparison of six nature-inspired metaheuristic algorithms in process optimization," *Processes*, vol. 10, no. 2, 2022, doi: 10.3390/pr10020197.

[10] A. M. Nassef, M. A. Abdelkareem, H. M. Maghrabie, and A. Baroutaji, "Review of metaheuristic optimization algorithms for power systems problems," *Sustainability (Switzerland)*, vol. 15, no. 12, 2023, doi: 10.3390/su15129434.

[11] Y. Wang *et al.*, "A dynamic opposite learning-assisted grey wolf optimizer," *Symmetry*, vol. 14, no. 9, 2022, doi: 10.3390/sym14091871.

[12] V. Chandran and P. Mohapatra, "Enhanced opposition-based grey wolf optimizer for global optimization and engineering design problems," *Alexandria Engineering Journal*, vol. 76, pp. 429–467, 2023, doi: 10.1016/j.aej.2023.06.048.

[13] A. A. Z. Diab, H. I. Abdul-Ghaffar, A. A. Ahmed, and H. A. Ramadan, "An effective model parameter estimation of PEMFCs using GWO algorithm and its variants," *IET Renewable Power Generation*, vol. 16, no. 7, pp. 1380–1400, 2022, doi: 10.1049/rpg2.12359.

[14] M. H. Nadimi-Shahraki, S. Taghian, and S. Mirjalili, "An improved grey wolf optimizer for solving engineering problems," *Expert Systems with Applications*, vol. 166, 2021, doi: 10.1016/j.eswa.2020.113917.

[15] L. Sun, B. Feng, T. Chen, D. Zhao, and Y. Xin, "Equalized grey wolf optimizer with refraction opposite learning," *Computational Intelligence and Neuroscience*, vol. 2022, 2022, doi: 10.1155/2022/2721490.

[16] W. Long, J. Jiao, X. Liang, S. Cai, and M. Xu, "A random opposition-based learning grey wolf optimizer," *IEEE Access*, vol. 7, pp. 113810–113825, 2019, doi: 10.1109/ACCESS.2019.2934994.

[17] A. Ali, Y. Ming, S. Chakraborty, and S. Iram, "A comprehensive survey on real-time applications of WSN," *Future Internet*, vol. 9, no. 4, 2017, doi: 10.3390/fi9040077.

[18] D. Han, Y. Yu, K. C. Li, and R. F. de Mello, "Enhancing the sensor node localization algorithm based on improved DV-Hop and DE algorithms in wireless sensor networks," *Sensors (Switzerland)*, vol. 20, no. 2, 2020, doi: 10.3390/s20020343.

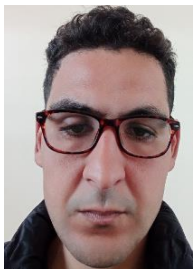
[19] D. Xue, "Research on range-free location algorithm for wireless sensor network based on particle swarm optimization," *Eurasip Journal on Wireless Communications and Networking*, vol. 2019, no. 1, 2019, doi: 10.1186/s13638-019-1540-z.

[20] S. Kessentini and D. Barchiesi, "Particle swarm optimization with adaptive inertia weight," *International Journal of Machine Learning and Computing*, vol. 5, no. 5, pp. 368–373, 2015, doi: 10.7763/ijmlc.2015.v5.535.

[21] D. Zhang, X. Zhang, and H. Qi, "A new location sensing algorithm based on DV-Hop and quantum-behaved particle swarm optimization in WSN," *ASP Transactions on Pattern Recognition and Intelligent Systems*, vol. 1, no. 2, pp. 1–17, 2021, doi: 10.52810/tpris.2021.100034.

[22] E. Shakshuki, A. Abu Elkhail, I. Nemer, M. Adam, and T. Sheltami, "Comparative study on range free localization algorithms," *Procedia Computer Science*, vol. 151, pp. 501–510, 2019, doi: 10.1016/j.procs.2019.04.068.
[23] G. Sharma and A. Kumar, "Improved range-free localization for three-dimensional wireless sensor networks using genetic algorithm," *Computers and Electrical Engineering*, vol. 72, pp. 808–827, 2018, doi: 10.1016/j.compeleceng.2017.12.036.
[24] S. Messous, H. Liouane, O. Cheikhrouhou, and H. Hamam, "Improved recursive dv-hop localization algorithm with rssi measurement for wireless sensor networks," *Sensors*, vol. 21, no. 12, 2021, doi: 10.3390/s21124152.
[25] 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, 2022, doi: 10.1155/2022/7026728.

BIOGRAPHIES OF AUTHORS



Omar Arroub received the bachelor in computer sciences and mathematics from Faculty of Sciences Rabat in 2011. He received the Master's in Computer Sciences and networks from Faculty of Sciences and Techniques of Tanger in 2014. He is currently a Ph.D. student at University Mohamed 5, Morocco. His research interests include wireless sensor network (WSN), internet of things (IoT), metaheuristics, and machine learning. He can be contacted at email: omar_arroub@um5.ac.ma.

Aouar Darif received the diplôme d'etudes supérieurs approfondies in computer sciences and telecommunications from Faculty of Sciences Rabat in 2007. He received the Ph.D. degree in computer sciences and telecommunications from Faculty of Sciences of Rabat in 2015. He is currently a research and teaching associate in the multidisciplinary Faculty at University of Sultan Moulay Slimane Beni Mellal, Morocco. His research interests include wireless sensor network (WSN), mobile edge computing (MEC), internet of things (IoT), cloud cumputing and neural networks. Anouar Darif is an active reviewer of various international conferences and journals. He can be contacted at email: anouar.darif@usms.ac.ma.

Rachid Saadane received the B.S. degree in physic electronic from the Faculty of Science of Rabat, Rabat, Morocco, in 2001. He received the Diplôme d 'Etudes Supérieurs Approfondies in computer sciences and telecommunications from the Faculty of Sciences of Rabat, in 2003. He received the Ph.D. degree in computer sciences and telecommunications from Faculty of Sciences of Rabat jointly with Eurecom Institute in 2007. He currently works as a research and a teaching Associate at the EHTP. He can be contacted at email: saadane@ehtp.ac.ma.

My Driss Rahmani is full professor of computer science in Mohammed V University in Rabat (Morocco) where he was Head of Department of Computer Science until 2015. He holds a Ph.D. in Sciences from University of Montpellier 2 (France) in 1989. He has over 25 years of teaching experience (concurrent programming, graphic user interface, compilation, and XML technology). His main area of interest includes wireless sensor network, urban modelling, Car-following modelling, business process modelling, and smart cities. He can be contacted at email: d.rahmani@um5r.ac.ma.

Zineb Aarab is an assistant professor of computer science and applied computer science at Mohammadia School of Engineers (EMI). She is a collaborator researcher at Computer Science and Telecommunications Research Laboratory LRIT. Her doctoral thesis has been prepared in the Computer Science and Telecommunications Research Laboratory LRIT. Her research interests lie in model-driven engineering (MDE) and information systems (IS) with a focus on context awareness. She can be contacted at email: aarab@emi.ac.ma.