



## AN EXPLORATION OF HIPPOPOTAMUS OPTIMISATION ALGORITHM FOR NODE LOCALISATION IN WIRELESS SENSOR NETWORKS

Dagadu, J. C.<sup>1</sup>, Antwi, E. B.<sup>2</sup>, Aboagye-Dapaah, E. O.<sup>3</sup>, and Kpikpi O. M.<sup>4</sup>

<sup>1,2,4</sup> *Department of Information Technology Education, University of Skills Training and Entrepreneurial Development, Ghana.*

<sup>3</sup> *Department of Computer Science, Kumasi Technical University, Kumasi, Ghana.*

<sup>1</sup>[joscaldag@yahoo.com](mailto:joscaldag@yahoo.com)

### ABSTRACT

**Purpose:** The study explores the application of the newly proposed Hippopotamus Optimisation Algorithm (HOA) for solving the node localisation problem in Wireless Sensor Networks.

**Design/Methodology/Approach:** In the proposed approach, anchor nodes with predetermined coordinates act as reference points, while HOA calculates the positions of unknown nodes by minimising the error between estimated and actual nodes. Performance is evaluated through simulations and compared with the standard Particle Swarm Optimisation (PSO) and Whale Optimisation Algorithm (WOA) algorithms using metrics such as localisation accuracy, the number of correctly localised nodes, and computational time.

**Research Limitation:** Despite its strong global search capability, for large-scale Wireless Sensor Networks (WSN), HOA will have a high computational cost because, as a population-based metaheuristic algorithm, it has to evaluate the localisation fitness function for a lot of WSN nodes (which are energy and computation constrained) across multiple iterations.

**Findings:** Results reveal that HOA has an advantage in strong early random exploration (i.e., peer-based movements and large random defence jumps to escape predators), which makes it very aggressive, thereby placing estimated solutions closer to true node positions very early. However, HOA has a high computational cost due to its heavy structure. Besides, its performance also degrades with more iterations because it does not preserve early-best solutions.

**Practical Implication:** Real-time or fully distributed localisation is challenging with HOA due to its high computational cost. Thus, it is more suitable for centralised or offline node localisation.

**Social Implication:** Accurate positioning of sensor nodes is essential for data gathering and efficient network operation because WSN play a key role in application domains such as smart cities, precision farming, environmental monitoring, and defence operations.

**Originality/Value:** HOA is a recent nature-inspired metaheuristic algorithm that has not been explored sufficiently to solve the node localisation problem in WSN. Thus, this is the first direct adoption and application of HOA to the node localisation problem in WSN.

**Keywords:** HOA. nature-inspired metaheuristic. node localisation. optimisation



## **INTRODUCTION**

Wireless Sensor Networks (WSNs) have emerged as a vital technology in modern applications such as environmental monitoring, disaster management, smart agriculture, intelligent transportation and military reconnaissance (Trigka & Dritsas, 2025; Kumari & Tyagi, 2024). A key necessity in WSNs is node localisation, which involves identifying the positions of unknown sensor nodes. Precise localisation not only enhances the reliability of sensed data but also facilitates efficient routing, coverage management and energy optimisation (Fadel et al., 2015). Traditional localisation techniques, such as GPS-based positioning, are unsuitable for large-scale WSNs due to their high cost, energy demands, and limited effectiveness in obstructed or indoor environments (Lavanya & Udgata, 2011). As a result, research has increasingly focused on algorithmic localisation techniques that rely on anchor nodes with known coordinates and a mathematical model to determine the positions of unknown nodes.

Localisation techniques are generally categorised into range-based and range-free approaches (Wang et al., 2010). Range-based techniques, including Received Signal Strength Indicator (RSSI), Time of Arrival (ToA) and Angle of Arrival (AoA) rely on measuring physical parameters to achieve high accuracy, but require additional hardware (Win et al., 2018). Range free methods, such as DV-Hop and centroid, leverage connectivity or hop counts between anchor and unknown nodes, providing scalability at the expense of accuracy (Aspnes et al., 2006).

To overcome the shortcomings of conventional approaches, nature-inspired metaheuristic algorithms have been extensively investigated for WSN localisation. These include Ant-Lion Optimization (ALO) (Dao et al., 2021), Salp Swarm Algorithm (SSA) (Kanoosh et al., 2019), Grey Wolf Optimization (GWO) (Rajakumar et al., 2017), Cuckoo Search (Goyal & Patterh, 2014), Firefly Optimization Algorithm (Harikrishnan et al., 2016), Cat Swarm Optimization (Zahia et al., 2023), and Butterfly Optimization Algorithm (BOA) (Arora & Singh, 2017). These algorithms iteratively reduce localisation error and demonstrate better adaptability in uncertain or dynamic network environments. Thus, the use of these nature-inspired algorithms to solve the localisation problem in WSNs is well grounded in theories such as optimisation and swarm intelligence. The node localisation problem in WSNs is formulated as an optimisation problem with the objective of minimising the error between measured distances (from known to unknown nodes) and estimated distances (from candidate node coordinates), as mathematically expressed in Section 3.3. In Swarm Intelligence theory, where agents interact to solve complex problems (via the principles of collective behaviour, self-organisation, and non-centralised control), candidate node positions are formulated as search agents. The swarm moves towards the best coordinate estimates collaboratively.



Despite the success of the earlier-mentioned algorithms, issues such as premature convergence, scalability, and maintaining a balance between exploration and exploitation remain unresolved research problems. The Hippopotamus Optimisation Algorithm (HOA) (Amiri et al., 2024), a novel swarm intelligence algorithm inspired by the foraging and defensive behaviours of hippopotamuses, was proposed recently. In HOA, each hippopotamus represents a potential solution, with its position adjusted through exploration (seeking food and evading predators) and exploitation (moving towards the optimal safe zone). This balance makes HOA a promising candidate for solving the WSN localisation problem, which is inherently non-linear and NP hard.

This study explores the potential of the HOA for solving the WSN localisation problem and evaluates its performance compared with the well-known Particle Swarm Optimisation (PSO) and the standard Whale Optimisation Algorithm (WOA) (Mirjalili & Lewis, 2016). The comparison is based on metrics such as localisation accuracy, computational time, and the number of nodes localised correctly. The rest of the paper is organised as follows: We briefly review the literature in Section 2, provide an overview of the HOA, and introduce the proposed HOA-based WSN node localisation in Section 3. Experimentation and results are presented in Section 4, and in Section 5, we give our concluding remarks.

## **LITERATURE REVIEW**

### **Traditional Node Localisation Approaches**

Early studies on node localisation in WSNs focused primarily on range-based and range-free approaches. Range-based methods exploit physical measurements such as RSSI, ToA, or AoA, offering high accuracy but requiring specialised hardware. Almuzaini & Gulliver (2010) extended this approach by integrating density-based outlier detection (DBOD), which improved robustness against anchor geometry challenges and transmission range limitations. However, the gain in accuracy came at the expense of increased computational complexity and energy consumption, making large-scale deployment challenging.

To reduce hardware dependency, range-free techniques were introduced. Zhang et al. (2008) introduced a distributed scheme that integrates sampling with a Genetic Algorithm (GA) to refine node positions when anchor density is sparse. While this method improved accuracy compared to pure trilateration, its performance degraded significantly in noisy environments, highlighting the need for more robust solutions. Singh & Sharma (2015) offered a comprehensive review of range-free algorithms such as APIT, DV-Hop, Centroid and Gradient techniques. These schemes were shown to be cost-effective and energy-efficient but suffered from reduced accuracy compared to range-based approaches.



Zaidi et al. (2016) categorised localisation algorithms into centralised vs distributed and deterministic vs probabilistic schemes. Their survey emphasised the scalability and hardware efficiency of range-free methods but also stressed their vulnerability to environmental noise and radio irregularities. The authors recommended hybrid approaches, combining range-free techniques with optimisation methods to overcome these shortcomings. This observation provides a direct motivation for exploring bio-inspired metaheuristic algorithms such as the HOA.

### **Optimisation-Based Node Localisation Approaches**

With the limitations of traditional techniques, modern research now relies on metaheuristic algorithms, which treat node localisation as a nonlinear optimisation problem. In this formulation, the objective function is to minimise localisation error between estimated and actual node coordinates. This model enables algorithms to balance exploration and exploitation.

Kanoosh et al. (2019) introduced the Salp Swarm Algorithm (SSA), which models salp chain behaviour to monitor the search process. Simulation results showed that SSA achieved lower mean localisation error, shorter computational time and higher localisation success rate compared to PSO, BOA, FA and GWO. However, the study was limited to simulation scenarios, ignoring real-world factors like energy consumption, noise, and scalability in large networks. Building on the need to prevent premature convergence in optimisation.

Cheng et al. (2022) proposed a Modified Archimedes Optimisation Algorithm (MAOA) that incorporates adaptive inertia and dynamic parameter control to overcome premature convergence. MAOA demonstrated higher accuracy and efficiency compared to PSO, GA and Archimedes Optimisation Algorithm (AOA), but experiments were confined to simulations without hardware variations. Seeking further improvement, Hao et al. (2021) introduced a hybrid method combining Voronoi diagrams and Support Vector Machines (SVM) to enhance positioning accuracy in WSNs, which first narrows the candidate region for each unknown node using geometric constraints, then refines positions with an SVM classifier. While this technique achieves a higher localisation accuracy and stability compared to the traditional range-free approaches, SVM introduces higher computational complexity.

Collectively, these studies highlight the potential of optimisation-based approaches but also point out recurring challenges: most remain confined to simulations, high computational overhead and energy efficiency issues. These shortcomings drive the need to investigate novel bio-inspired algorithms like the HOA, which seeks to balance exploration and exploitation while reducing complexity and improving robustness in WSN localisation.



To the best of our knowledge, HOA has not been used for the localisation problem so far. Hence, in this paper, we explore the use of the raw/original HOA to optimise the localisation problem and compare its performance with two other established algorithms.

## MATERIALS AND METHODS

### Study Design

A simulation-based experimental design was used in assessing the performance of the proposed HOA-based Node Localisation Algorithm for WSNs. Randomly deployed anchors and unknown nodes were simulated in a two-dimensional non-convex network environment. WSN node localisation is formulated as an optimisation problem aimed at minimising error between measured and estimated anchor distances. Mean localisation error, time of execution, and number of correctly localised nodes are metrics used to measure performance.

### Overview of Hippopotamus Optimisation Algorithm (HOA)

The Hippopotamus Optimisation Algorithm (HOA) is a novel swarm-based metaheuristic algorithm inspired by the social behaviours of hippopotamuses (Amiri et al., 2024). The algorithm mimics how hippos explore their environments, defend against danger, and evade predators to balance exploration and exploitation in a search field. Each hippopotamus symbolises a candidate solution, and its position denotes a potential solution in the optimisation space. The stages of the HOA are summarised as follows (Amiri et al., 2024):

#### *Stage 1: Initialisation phase*

The initialisation phase of the HO algorithm involves generating random candidate solutions, similar to other metaheuristic algorithms. The formula is as follows:

$$X_{i,j} = Y_{l,j} + r(Y_{u,j} - Y_{l,j}) \quad i = 1,2 \dots N; j = 1,2 \dots M \quad (1)$$

Where  $N$  is the population size,  $M$  is the decision variable,  $X_{i,j}$  denotes the candidate location of the  $i$ - the solution,  $j$  shows the current decision variable,  $r$  is the random number within the range of  $[0, 1]$ ,  $Y_{u,j}$  is the upper bound and  $Y_{l,j}$  is the lower bound.

#### *Stage 2: Position update of Hippos in the lake or pond (Exploration)*

The herd of the hippo is made up of males, females, young, and a dominant leader. Males drive movement within the group, while displaced males either seek mates or compete for dominance. Equations (2) and (3) represent these interactions and the resulting position updates within the group.

$$X_{ij}^{Mhippo} = Y_{i,j} + r(X^{Dhippo} - l_1 X_{i,j}) \quad i = 1,2 \dots \frac{N}{2}; j = 1,2 \dots M \quad (2)$$



$$S = \left\{ \begin{array}{c} l_2 * r_1 + (\sim p_1) \\ 2 * r_2 - 1 \\ r_3 \\ l_1 * r_4 + (\sim p_2) \\ r_5 \end{array} \right\} \quad (3)$$

The position of a male hippo is denoted by  $X_{ij}^{Mhippo}$ , while the best hippo's location is represented by  $X^{Dhippo}$ .  $l_1$  and  $l_2$  are integers within the range of [1,2].  $r_1 - r_4$  are random vectors in the range [0,1]. Young hippos leave the herd due to curiosity, if  $T > 0.6$ . It shows that a young hippo has left the group. If random  $r_6 > 0.5$ , it means the young hippo is not far from the herd; otherwise, the young hippo is far from the group.

$$T = esp\left(-\frac{t}{T_{max}}\right) \quad (4)$$

$$X_{i,j}^{FBhippo} = \left\{ \begin{array}{l} X_{i,j} + S_1 \\ Z \quad else \end{array} (X^{Dhippo} - l_2 h_i) T > 0.6 \right\} \quad (5)$$

$$Z = \left\{ \begin{array}{l} X_{i,j} + S_2 (h_i - X_{i,j}^{Dhippo}) r_6 > 0.5 \\ Y_{l,j} + r_7 (Y_{u,j} - Y_{l,j}) else \end{array} \right\} \quad (6)$$

$T$  signifies the distance between the current iteration and the maximum number of iterations. Here,  $t$  shows the current iteration number, and  $T_{max}$  is the maximum number of iterations.  $X_{i,j}^{FBhippo}$  shows the updated position of young hippopotamuses.  $h_i$  denotes the average position of a group of random hippos, while  $Z$  indicates the status of young hippos that are positioned far from the main herd. The position update for both male and female is defined as follows:

$$X_i = \left\{ \begin{array}{ll} X_i^{Mhippo} & F_i^{Mhippo} < F_i \\ X_i & else \end{array} \right\} \quad (7)$$

$$X_i = \left\{ \begin{array}{ll} X_i^{FBhippo} & F_i^{FBhippo} < F_i \\ X_i & else \end{array} \right\} \quad (8)$$

Where  $F_i$  shows the objective function,  $F_i^{Mhippo}$  is the fitness of male hippopotamus and  $F_i^{FBhippo}$  corresponds to the fitness of the female or young hippopotamus.

### **Stage 3: Defensive phase (Exploration):**

Hippopotamuses always defend their region when a predator tries to attack them. Young hippos wander away from the herd out of curiosity, making them vulnerable to predators



such as lions, leopards, hyenas, etc. To shield themselves, hippos exhibit defensive behaviours such as producing a vocalisation to frighten predators and discourage them from approaching. Predator's position is shown by Equation (9).

$$X_j^{Predator} = Y_{l,j} + r_8(Y_{u,j} - Y_{l,j}) \quad j = 1,2 \dots M \quad (9)$$

Where  $r_8$  is a random vector within the range of [0,1]

$$D = |X_j^{Predator} - X_{i,j}| \quad (10)$$

Equation (10) denotes the distance between the hippopotamus and the predator. At this stage, the hippo uses a defensive strategy based on the factor  $F_j^{predator}$ . The hippopotamus moves in the direction of the predator and forces it to withdraw if  $F_j^{predator}$  is less than  $F_i$ , which shows that the predator is closer to the hippopotamus. However, if  $F_j^{predator}$  is higher, it indicates that the predator is far away; the hippo then chases the predator with a lower level of aggression. This is described by Equation (11).

$$X_{i,j}^{HippoR} = \begin{cases} RL \oplus X_j^{predator} + \frac{\alpha}{(c-d*\cos(2\pi\beta))} * \frac{1}{D} & F_j^{predator} < F_i \\ RL \oplus X_j^{predator} + \frac{\alpha}{(c-d*\cos(2\pi\beta))} * \frac{1}{2*D+r_9} & F_j^{predator} \geq F_i \end{cases} \quad (11)$$

The defensive position of a hippo, when encountering a predator, is denoted as  $X_{i,j}^{HippoR}$  which reflects a restricted movement change. The term  $RL$  refers to a levy-based random vector used to model unpredictable shifts in the predator's location. Equation (12) is applied to generate the levy-based random movement used in the model, where variables  $\omega$  and  $\mu$  are random numbers in range [0, 1],  $\vartheta$  is a constant equal to 1.5,  $\Gamma$  denotes gamma function and  $\sigma_\omega$  is shown in equation (13).

$$Levy(\vartheta) = 0.05 * \frac{\omega * \sigma_\omega}{|\mu|^{\frac{1}{\vartheta}}} \quad (12)$$

$$\sigma_\omega = \left| \frac{\Gamma(1+\vartheta) \sin(\frac{\pi\vartheta}{2})}{\Gamma(\frac{(1+\vartheta)}{2}) \vartheta * 2^{\frac{(\vartheta-1)}{2}}} \right|^{\frac{1}{\vartheta}} \quad (13)$$

According to equation (14), when  $F_i^{HippoR} > F_i$  it shows that the hippo has been replaced. If the condition is reversed, the predator escapes, and the hippo rejoins the herd.



$$X_i = \begin{cases} H_i^{HippoR} F_i^{HippoR} < F_i \\ X_i & F_i^{HippoR} \geq F_i \end{cases} \quad (14)$$

#### **Stage 4: Escape Phase (Exploitation)**

When hippos are surrounded by numerous predators or fail to protect themselves, they retreat towards a nearby water source for protection. This is the third phase of the HOA where hippos search for secure areas to hide. New positions with a higher fitness function indicate that the hippo has located a safer environment.

$$\begin{cases} Y_{i,j}^{local} = \frac{Y_{i,j}}{t} \\ Y_{u,j}^{local} = \frac{Y_{u,j}}{t} \end{cases} \quad t = 1, 2 \dots T_{max} \quad (15)$$

$$X_{i,j}^{Hippo\epsilon} = X_{i,j} + r_{10}[Y_{l,j}^{local} + K(Y_{u,j}^{local} - Y_{l,j}^{local})] \quad (16)$$

The local upper and lower bounds are denoted as  $Y_{u,j}^{local}$  and  $Y_{l,j}^{local}$  respectively. The term  $X_{i,j}^{Hippo\epsilon}$  represents the location of the hippo during its search for a safe hiding location, while  $r_{10}$  is a random number in the interval [0, 1].

$$K = \begin{cases} 2 * r_{11} - 1 \\ r_{12} \\ r_{13} \end{cases} \quad (17)$$

$K$  Is randomly selected from the three scenarios; each scenario guides the local search process, helping it to effectively refine and exploit better solutions. From Equation 17,  $r_{11}$  and  $r_{13}$  are random numbers in the interval [0, 1],  $r_{12}$  is a random number with a normal distribution

$$X_i = \begin{cases} X_i^{Hippo\epsilon} & F_i^{Hippo\epsilon} < F_i \\ X_i & F_i^{Hippo\epsilon} \geq F_i \end{cases} \quad (18)$$

The algorithm continuously tracks and updates the best-performing solution. After all iterations are completed, the best candidate solution is identified as the final solution to the problem.



### **Formulation of the WSN node localisation problem**

Node localisation in WSNs is formulated as an optimisation problem aimed at estimating the coordinates of unknown nodes by minimising localisation error. The process is described as follows (Arora & Singh, 2017; Rajakumar et al., 2017; Kanoosh et al., 2019):

1. Randomly deploy  $M$  anchor nodes and  $N$  unknown nodes within a communication range  $R$ . Anchor nodes broadcast their coordinates to the neighbouring nodes.
2. Euclidean distance between a target node  $(x, y)$  and anchor node  $(x_i, y_i)$  is calculated as

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (19)$$

3. The optimisation problem is formulated to minimise the error of localisation. The objective function for the localization problem is formulated as

$$f(x, y) = \min \sum_{i=1}^M (\sqrt{(x - x_i)^2 + (y - y_i)^2} - d_i)^2 \quad (20)$$

Where  $M$  is the anchor node within the transmission range  $R$  of the target node.

4. Once a localisable node  $N_L$  are identified, the whole localisation error is calculated as the mean of the squares of distances of the estimated coordinates of the node  $x_i, y_i$  and the actual node coordinates  $X_i, Y_i$ , for  $i = 1, 2, \dots, N_L$

$$E_L = \frac{1}{N_L} \sum_{i=1}^{N_L} (\sqrt{(x_i - X_i)^2 + (y_i - Y_i)^2}) \quad (21)$$

The performance of HO algorithm is evaluated using  $E_L$  and the number of non-localised nodes  $N_{NL}$ , where  $N_{NL} = [N - N_L]$ .

5. Repeat steps 2-4 until all unknown nodes get localised or no more nodes can be localised.

### **Mapping of HOA on node localisation**

To directly apply the HOA algorithm to the Node Localisation problem, we first do a mapping as shown in Table 1.



*Table 1: Mapping of HOA and Node Localisation*

| <b>Stage</b>                                     | <b>Description in HOA</b>  | <b>Node Localisation Application</b>   |
|--|--|--|
| Initialisation of population                     | A herd of hippopotamuses are positioned randomly within the search space.  | Sensor nodes are randomly positioned in the wireless sensor deployment area. Each hippopotamus in HOA represents the estimated coordinates for all unknown sensor nodes (candidate solutions).   |
| Fitness Evaluation                               | The strength of each hippopotamus is assessed based on its position in the environment. The strongest hippopotamus is identified as the dominant male  | The localisation error of each candidate solution is calculated based on estimated distances between anchor and unknown nodes. The lower the error (the objective function), the better (strength) the solution. The candidate solution with the lowest error is identified as the best solution |
| Phase 1: Position update (exploration)           | Each hippopotamus explores different areas of the search space for new sources of food and makes that area its territory.  | Candidate nodes are perturbed randomly to explore new possible positions across the deployment area, to avoid early convergence  |
| Phase 2: Defence against predators (exploration) | Hippopotamuses aggressively protect their territories while searching locally within the territory   | Candidate nodes exploit a promising region around the best node position estimates to reduce estimation errors by moving closer, with smaller steps to the best known solution coordinates.  |
| Phase 3: Escape from predators (Exploitation)    | If there are stronger predators or the territory is not favourable, hippopotamuses escape or relocate.   | If there is the risk of early convergence, candidate solutions make random moves to new positions away from the current best known solution coordinates. This escapes local minima to ensure global convergence  |
| Position Updates, re-evaluation and selection    | After exploration and exploitation, a hippopotamus moves to a new position. Its strength is re-evaluated to reflect its new state. If the new position is better, it is adopted by the hippo; else, the previous position is maintained. | Coordinates of candidate nodes are updated with their new positions in the deployment area. The localisation error is recalculated. If the error associated with the new position is lower, the current coordinates are adopted; else, the previous coordinates are maintained.                  |
| Termination                                      | The process is iterated until the maximum is reached.  | When the maximum iteration is reached or the localisation error becomes negligible, the search is terminated. The best estimated positions of all unknown nodes are returned.  |



---

*Table 2: Algorithm 1: HOA-Based Node Localisation*

---

Start

```
1: Initialize the hippopotamus population. (Equation 1)
2: Evaluate the fitness of each search agent
3: Choose the best solution Xbest
4: // main algorithmic loop
5: For t = 1 to Tmax do
6:   Let  $T = \exp(-t / T_{max})$  using Equation 4
7:   //Phase 1: Updating node positions (Exploration)
8:     For i = 1 to N/2 do
9:       Find new position (Xnew) of unknown nodes using Equation 2
10:      Apply the boundary constraints to Xnew
11:       $F_{new} = \text{calculate\_Fitness}(\mathbf{X}_{new}[i])$ 
12:      If  $F_{new} < F[i]$ 
13:        Update  $X_i$  with Equation 7
14:      End if
15:    End For
16:    For i = N/2 + 1 to N do
17:      If  $T > 0.6$ 
18:        Generate Xnew using Equation 5
19:      Else
20:        Generate Xnew using Equation 6
21:      End if
22:      Apply the boundary constraints to Xnew
23:      If  $F_{new} < F[i]$ 
24:        Update  $X_i$  using Equation 8
25:      End if
26:    End For
27:   //Phase 2: Defensive (Exploration)
28:   For i = N/2 + 1 to N Do
29:     Generate predator position using Equation 9
30:     Compute distance to predator using Equation 10
31:     Apply defensive movement to assume Xnew position using Equation 11
32:     Apply the boundary constraints to Xnew
33:     If  $F_{new} < F[i]$ 
34:       Update  $X_i$  using Equation 14
35:     End if
36:   End For
37:   //Phase 3: Escaping from predators (Exploitation)
38:   For i = 1 to N do
39:     Reduce bounds:  $lb_{local} = lb/t$ ,  $ub_{local} = ub/t$ , using Equation 15
40:     Compute escape movement to position Xnew using Equations 16 and 17
```

---



---

```
41:      Apply the boundary constraints to X_new
42:      If F_new < F[i]
43:          Update Xi using Equation 18
44:      End if
45:  End For
46: Update X_best if a better solution is found
47: End For
48: Return X_best
```

End

---

## **EXPERIMENTATION AND RESULTS**

### **Experimental setup**

The proposed HOA-based node localisation algorithm is evaluated under different simulation scenarios. Further, its performance is compared with those of PSO and WOA, using three performance metrics: localisation error, computational time, and number of nodes localised successfully. All computations are carried out using MATLAB R2021a on a machine equipped with an Intel Core i7, 16 GB RAM and Windows 11 Operating System. The simulation environment is modelled as a two-dimensional area of 300 \* 300 units, within which nodes are deployed randomly. The number of anchor nodes are varied incrementally as  $\sum_{i=1}^4 i * 10$ . It is assumed that all unknown nodes are within range and can hear all anchor nodes; thus, are localizable. For comparison purposes, the number of iterations is varied (as 1, 50, and 100) to investigate its effect on performance. The WOA used for the experiment is based on the original algorithm, adapted by minimising distance estimation error via the inclusion of a greedy selection mechanism to enhance convergence speed.

### **Performance of Node Localisation using HOA**

The HOA's performance on node localisation is assessed using different scenarios. Different ratios of anchor and unknown nodes were randomly deployed (i.e., 10 anchor nodes to 25 unknown nodes, 20 anchor nodes to 50 unknown nodes, 30 anchor nodes to 75 unknown nodes, 40 anchor nodes to 100 unknown nodes) in the network area. and the algorithm run a number of times. The best performance for each setup is shown in Figure 1. For each run, the hippopotamus population is 30, the maximum number of iterations is 100, the noise is 0.5, and the correct localisation threshold is set at 10 meters. From the figure, it can be observed that the HOA performs very well by localising the majority of the unknown nodes correctly with minimal localisation errors. This is further shown by the values in Table 3.

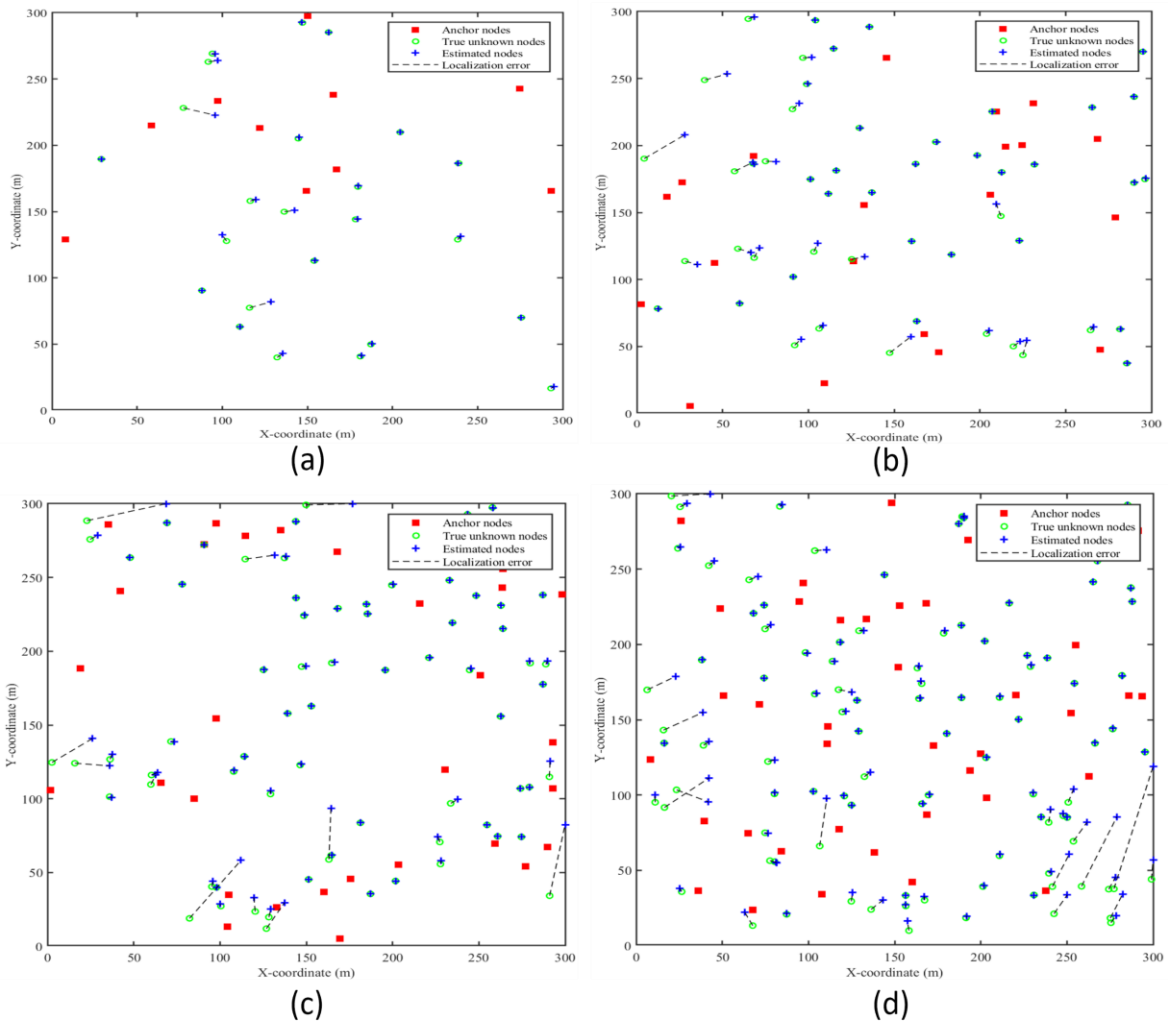


Figure 1: Node localisation performance using HOA. (a) Anchor nodes = 10, Target nodes=25. (b) Anchor nodes = 20, Target nodes=50. (c) Anchor nodes = 30, Target nodes=75. (d) Anchor nodes = 40, Target nodes=100.



*Table 3: Results of node localisation using HOA*

| Anchor | Unknown | Execution Time (s) | Mean localisation Error (m) | RMSE (m) | No. of Correctly Localised Nodes |
|--------|---------|--------------------|-----------------------------|----------|----------------------------------|
| 10     | 25      | 0.4164             | 2.8713                      | 5.3064   | 23 / 25 nodes                    |
| 20     | 50      | 0.8455             | 3.6682                      | 6.7023   | 45 / 50 nodes                    |
| 30     | 75      | 1.4174             | 5.1367                      | 12.3171  | 65 / 75 nodes                    |
| 40     | 100     | 1.6947             | 5.3822                      | 12.6653  | 87 / 100 nodes                   |

### Comparison with PSO and WOA

The performance of the HOA-node localisation is compared with other well-known algorithms, using the same experimental settings (with hippopotamus population set at 30). It is worth noting that at each execution instance, node locations are re-randomised, so performance values for the algorithms change. The number of iterations is varied (i.e., 1, 50, 100) to evaluate performance. For each iteration value, the algorithms are run using the same randomised node positions; the best HOA performance after 10 runs is recorded (i.e., MLE=Mean Localisation Error, NLC=Nodes Localised Correctly). The aim of varying the number of iterations is to find out whether this greatly impacts performance. The results are shown in Table 4.

*Table 4: Performance comparison of HOA with PSO and WOA*

| Anchor | Target | Iteration | HOA  |        |     | PSO  |        |     | WOA  |        |     |
|--------|--------|-----------|------|--------|-----|------|--------|-----|------|--------|-----|
|        |        |           | Time | MLE    | NLC | Time | MLE    | NLC | Time | MLE    | NLC |
| 10     | 25     | 1         | 0.03 | 15.485 | 9   | 0.02 | 10.357 | 14  | 0.01 | 21.64  | 6   |
|        |        | 50        | 0.39 | 3.708  | 24  | 0.1  | 0.52   | 25  | 0.07 | 0.278  | 25  |
|        |        | 100       | 0.42 | 2.871  | 23  | 0.19 | 0.255  | 25  | 0.14 | 0.548  | 25  |
| 20     | 50     | 1         | 0.05 | 17.898 | 21  | 0.03 | 7.981  | 37  | 0.01 | 19.108 | 14  |
|        |        | 50        | 0.55 | 8.151  | 42  | 0.29 | 0.471  | 49  | 0.15 | 0.215  | 50  |
|        |        | 100       | 0.85 | 3.668  | 45  | 0.38 | 0.5    | 49  | 0.31 | 0.244  | 50  |
| 30     | 75     | 1         | 0.09 | 18.391 | 28  | 0.05 | 9.741  | 46  | 0.04 | 21.321 | 21  |
|        |        | 50        | 0.79 | 5.765  | 65  | 0.3  | 0.298  | 75  | 0.22 | 0.361  | 74  |
|        |        | 100       | 1.42 | 5.137  | 65  | 0.59 | 0.289  | 75  | 0.42 | 0.172  | 75  |
| 40     | 100    | 1         | 0.11 | 16.463 | 37  | 0.08 | 10.453 | 62  | 0.05 | 18.508 | 26  |
|        |        | 50        | 0.88 | 6.436  | 83  | 0.41 | 0.356  | 99  | 0.32 | 0.156  | 100 |
|        |        | 100       | 1.55 | 4.974  | 85  | 0.84 | 0.227  | 100 | 0.72 | 0.15   | 100 |

In Table 4, it is clear that at just one iteration, HOA's performance is quite comparable with the other two algorithms, and is in fact better than WOA in some cases. However, when the iterations increase, the other two algorithms far outperform HOA, reaching perfectly correct



localisation values. This is further shown in Figures 2, 3, and 4, using the same experimental setting with 40 anchor nodes and 100 target nodes.

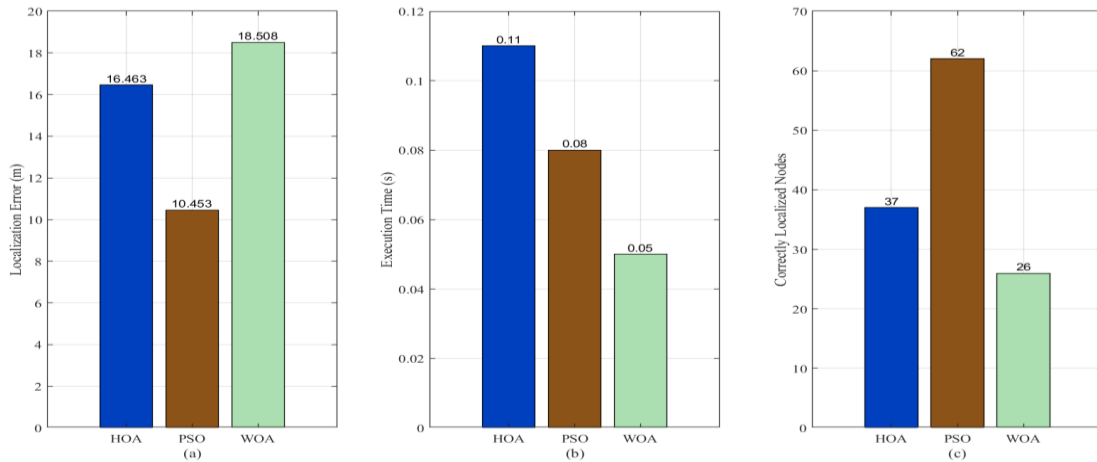


Figure 2: Performance comparison at iteration 1. (a) Localisation error (b) Execution time. (c) Number of correctly localised nodes.

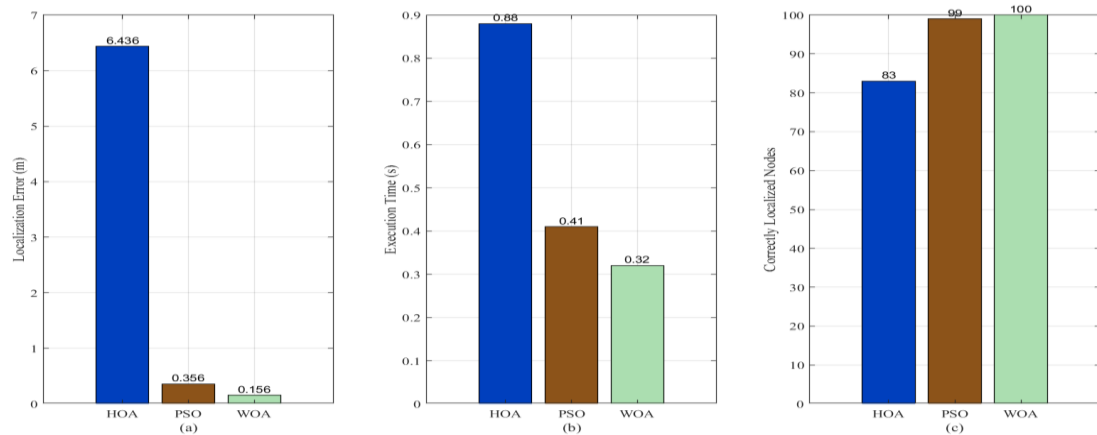


Figure 3: Performance comparison at iteration 50. (a) Localisation error (b) Execution time. (c) Number of correctly localised nodes.

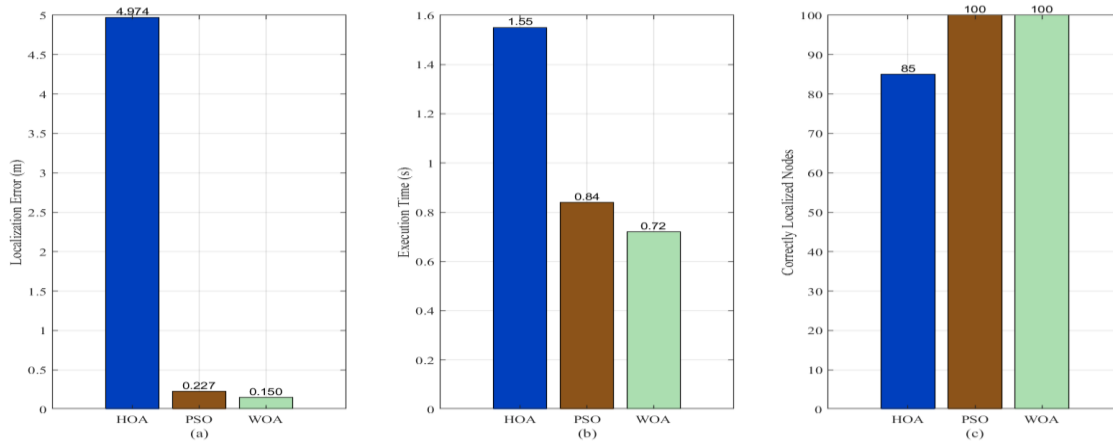
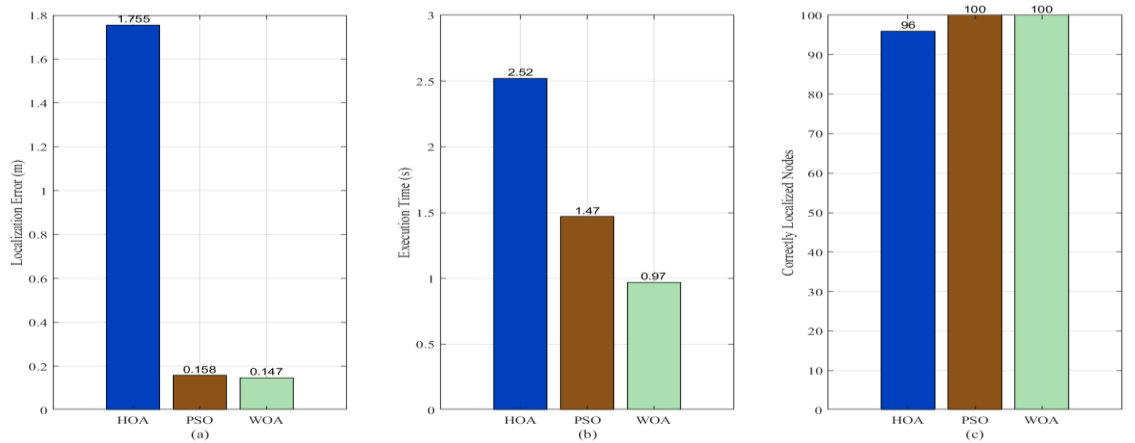


Figure 4: Performance comparison at iteration 100. (a) Localisation error (b) Execution time. (c) Number of correctly localised nodes.

These observations suggest that HOA's early exploration behaviour makes it more aggressive at the first iteration; thus, it has strong random exploration (i.e., peer-based movements and large random defence jumps to escape predators) that could place solutions closer to true node positions as compared to WOA, which has early weak exploitation and less randomness at iteration one. With more iterations, WOA converges faster with better accuracy due to its structured exploitation mechanism, while HOA continues with high exploration and low exploitation. These also account for the mean localisation error values. When it comes to execution time, HOA is consistently the worst with execution time and localisation accuracy due to the high computational cost associated with it, stemming from its multi-phase update mechanism, which requires multiple fitness evaluations for each iteration, and the inability to retain best solutions at the initial iterations. HOA, therefore, has limitations with convergence and speed as highlighted by other studies (Han et al., 2025; Pei et al., 2025).

A trend that is noticed is that, when the population is set to 100, even at 50 iterations, HOA performs well in terms of Correctly Localised Nodes, though it lags behind PSO and WOA, as shown in Figure 5.



*Figure 5: Performance comparison at population of 100, iteration 50, 40 anchor nodes, 100 unknown nodes. (a) Localisation error (b) Execution time. (c) Number of correctly localised nodes.*

This is because the candidate solutions have increased; thus, there is better coverage of the search space. Therefore, the algorithm is able to discover better candidate node positions even at higher iterations. However, owing to its exploration-dominant search strategy and multi-phase update, HOA still lags behind WOA and PSO in convergence. This suggests that the algorithm is sensitive to population size; as such, it may achieve competitive accuracy with larger population sizes.

## CONCLUSION

In this paper, we have explored the deployment of the recently proposed Hippopotamus Optimisation Algorithm for node localisation in WSNs. The performance evaluation of the deployment indicates that the algorithm is able to perform well and is suitable for node localisation in WSNs. Comparisons with other well-known algorithms suggest that, depending on certain factors, HOA has the potential to outperform other algorithms but maintains a high computational cost. Moreover, due to its exploration, it has an early advantage, but this diminishes with time owing to its weak exploitation.

To maintain this early advantage, modifications such as adaptive exploitation of promising solutions, preservation of best early solutions, increased search agent population, and local refinement of good solutions could be made to this algorithm to improve its performance on node localisation. The implication of our findings is that not all nature-inspired algorithms perform very well in addressing all problems. The original HOA seems less suited for low-dimensional, smooth, and continuous optimisation problems.



## REFERENCES

- Almuzaini, K. K., & Gulliver, A. (2010). Range-Based Localization in Wireless Networks Using Density-Based Outlier Detection. *Wireless Sensor Network*, 02(11), 807–814. <https://doi.org/10.4236/wsn.2010.211097>
- Amiri, M. H., Mehrabi Hashjin, N., Montazeri, M., Mirjalili, S., & Khodadadi, N. (2024). Hippopotamus optimization algorithm: a novel nature-inspired optimization algorithm. *Scientific Reports*, 14(1). <https://doi.org/10.1038/s41598-024-54910-3>
- Arora, S., & Singh, S. (2017). Node Localization in Wireless Sensor Networks Using Butterfly Optimization Algorithm. *Arabian Journal for Science and Engineering*, 42(8), 3325–3335. <https://doi.org/10.1007/s13369-017-2471-9>
- Aspnes, J., Eren, T., Goldenberg, D. K., Morse, A. S., Whiteley, W., Yang, Y. R., Anderson, B. D. O., & Belhumeur, P. N. (2006). *A Theory of Network Localization*. *IEEE Transactions on Mobile Computing*, 5(12), 1663-1678. [10.1109/TMC.2006.174](https://doi.org/10.1109/TMC.2006.174)
- Cheng, M., Qin, T., & Yang, J. (2022). Node Localization Algorithm Based on Modified Archimedes Optimization Algorithm in Wireless Sensor Networks. *Journal of Sensors*, 2022. <https://doi.org/10.1155/2022/7026728>
- Dao, T. K., Pan, J. S., Nguyen, T. T., Chu, S. C., Tran, H. T., Nguyen, T. D., & Vu, N. T. (2021, July). Node localization in wireless sensor network by ant lion optimization. In *Advances in Smart Vehicular Technology, Transportation, Communication and Applications: Proceeding of the Third International Conference on VTCA, 15–18 October 2019, Arad, Romania* (pp. 97-109). Singapore: Springer Singapore. [https://doi.org/10.1007/978-981-16-1209-1\\_10](https://doi.org/10.1007/978-981-16-1209-1_10)
- Fadel, E., Gungor, V. C., Nassef, L., Akkari, N., Abbas Malik, M. G., Almasri, S., & Akyildiz, I. F. (2015). A survey on wireless sensor networks for smart grid. *Computer Communications*, 71, 22–33. <https://doi.org/10.1016/j.comcom.2015.09.006>
- Goyal, S., & Patterh, M. S. (2014). Wireless sensor network localization based on cuckoo search algorithm. *Wireless personal communications*, 79(1), 223-234. <https://doi.org/10.1007/s11277-014-1850-8>
- Han, T., Wang, H., Li, T., Liu, Q., & Huang, Y. (2025). MHO: A modified hippopotamus optimization algorithm for global optimization and engineering design problems. *Biomimetics*, 10(2), 90. <https://doi.org/10.3390/biomimetics10020090>
- Hao, Z., Dang, J., Yan, Y., & Wang, X. (2021). A node localization algorithm based on Voronoi diagram and support vector machine for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 17(2). <https://doi.org/10.1177/1550147721993410>
- Harikrishnan, R., Jawahar Senthil Kumar, V., & Sridevi Ponmalar, P. (2016). Firefly algorithm approach for localization in wireless sensor networks. *Smart Innovation, Systems and Technologies*, 44, 209–214. [https://doi.org/10.1007/978-81-322-2529-4\\_21](https://doi.org/10.1007/978-81-322-2529-4_21)



- Kanoosh, H. M., Houssein, E. H., & Selim, M. M. (2019). Salp Swarm Algorithm for Node Localization in Wireless Sensor Networks. *Journal of Computer Networks and Communications*, 2019. <https://doi.org/10.1155/2019/1028723>
- Kumari, S., & Tyagi, A. K. (2024). Wireless sensor networks: An introduction. *Digital Twin and Blockchain for Smart Cities*, 495-528. <https://doi.org/10.1002/9781394303564.ch21>
- Lavanya, D., & Udgata, S. K. (2011). *Swarm Intelligence Based Localization in Wireless Sensor Networks*. In *International workshop on multi-disciplinary trends in artificial intelligence* (pp. 317-328). Berlin, Heidelberg: Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-642-25725-4\\_28](https://doi.org/10.1007/978-3-642-25725-4_28)
- Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, 95, 51-67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- Pei, S., Sun, G., & Tong, L. (2025). An improved hippopotamus optimization algorithm based on adaptive development and solution diversity enhancement. *PeerJ Computer Science*, 11, e2901. <https://doi.org/10.7717/peerj-cs.2901>
- Rajakumar, R., Amudhavel, J., Dhavachelvan, P., & Vengattaraman, T. (2017). GWO-LPWSN: Grey Wolf Optimization Algorithm for Node Localization Problem in Wireless Sensor Networks. *Journal of Computer Networks and Communications*, 2017. <https://doi.org/10.1155/2017/7348141>
- Singh, S. P., & Sharma, S. C. (2015). Range Free Localization Techniques in Wireless Sensor Networks: A Review. *Procedia Computer Science*, 57, 7–16. <https://doi.org/10.1016/j.procs.2015.07.357>
- Trigka, M., & Dritsas, E. (2025). Wireless sensor networks: From fundamentals and applications to innovations and future trends. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2025.3572328>
- Wang, J., Ghosh, R. K., & Das, S. K. (2010). A survey on sensor localization. *Journal of Control Theory and Applications*, 8(1), 2–11. <https://doi.org/10.1007/s11768-010-9187-7>
- Win, M. Z., Shen, Y., & Dai, W. (2018). A Theoretical Foundation of Network Localization and Navigation. *Proceedings of the IEEE*, 106(7), 1136–1165. <https://doi.org/10.1109/JPROC.2018.2844553>
- Zahia, L., Fouzi, S., & Samra, B. (2023). Node localization optimization in WSNs by using cat swarm optimization meta-heuristic. *Automatic Control and Computer Sciences*, 57(2), 177-184. <https://doi.org/10.3103/S0146411623020104>
- Zaidi, S., El Assaf, A., Affes, S., & Kandil, N. (2016). Accurate Range-Free Localization in Multi-Hop Wireless Sensor Networks. *IEEE Transactions on Communications*, 64(9), 3886–3900. <https://doi.org/10.1109/TCOMM.2016.2590436>
- Zhang, Q., Wang, J., Jin, C., Ye, J., Ma, C., & Zhang, W. (2008, October). Genetic algorithm based wireless sensor network localization. In *2008 Fourth International Conference*



***African Journal of Applied Research***

***Vol. 12, No. 3 (2026), pp. 245-264***

***<http://www.ajaronline.com>***

***<https://doi.org/10.26437/ajar.v12>***

*Received October 2, 2025*

*Peer reviewed: March 13, 2026*

*Revised: March 18, 2026*

*Published: April 2026*

*on Natural Computation* (Vol. 1, pp. 608-613). IEEE Computer Society.  
[10.1109/ICNC.2008.206](https://doi.org/10.1109/ICNC.2008.206)

ISSN: 2408-7920

Copyright © African Journal of Applied Research

Arca Academic

