



# Enhanced DV-Hop using shuffled shepherd algorithm for localizing sensor nodes in 3-D space

Ahmed El. Gouda<sup>1</sup> · Yasmine Abouelseoud<sup>1</sup> · Ahmed H. Abd El-Malek<sup>2</sup> · Radwa Ahmed Osman<sup>3</sup>

Accepted: 18 July 2023 / Published online: 24 August 2023  
© The Author(s) 2023

## Abstract

For many wireless sensor networks applications, a powerful localization algorithm is required to determine the exact positions of sensor nodes. In this paper, a new localization algorithm is presented which combines the distance vector hop (DV-Hop) algorithm with the shuffled shepherd optimization algorithm to determine the position for each unknown node within 3-D space. Studying the localization problem in 3-D space is more realistic, but more challenging due to the enlarged search space. Comparison of our proposed algorithm with its alternatives in literature shows that it offers improved accuracy and more stable performance with respect to changes in nodes density and communication range.

**Keywords** DV-Hop algorithm · Shuffled shepherd optimization algorithm · Wireless sensor networks · Anchor nodes

## 1 Introduction

Wireless sensor networks (WSNs) have many applications such as in forest fire detection, area monitoring to identify the position of the enemy, and also in monitoring the conditions of patients. Wireless sensor networks consist of small devices which are called sensors and one of their important functions is collecting and sending data. For

sending data, it is essential to know the location of the transmitting node, which is done by using techniques of localization [1, 2]. The objective of localization techniques is to determine the coordinates of each node in a specific sensing area.

There are many different ways for node localization, one of these is to add a Global Positioning System (GPS) to each sensor node [3]. Adding a GPS receiver to each node is not a good solution due to its large power consumption, and high cost. Different localization algorithms for sensor networks have been reviewed in [4], where only a few nodes are equipped with GPS with known locations referred to as anchor nodes, and the remaining nodes are referred to as unknown nodes.

The localization techniques can be categorized as:

- Centralized localization techniques.
- Distributed localization techniques.

Where the distributed techniques can be further classified into:

- Range-based schemes.
- Range-free schemes.

In centralized localization techniques, all the measurements are collected at a central base station (BS), where the computation takes place [5–7]. Subsequently, the results are forwarded back to the nodes. The transmission of data through the network causes latency, more energy and bandwidth consumption [8]. The benefit of this technique is

---

Yasmine Abouelseoud, Ahmed H. Abd El-Malek, and Radwa Ahmed Osman have contributed equally to this work.

---

✉ Ahmed El. Gouda  
ahmed.gouda@alexu.edu.eg; goda13@hotmail.com

Yasmine Abouelseoud  
yasmine.abouelseoud@gmail.com;  
yasmine.abouelseoud@alexu.edu.eg

Ahmed H. Abd El-Malek  
ahmed.abdelmalek@ejust.edu.eg

Radwa Ahmed Osman  
radwa.ahmed@aast.edu

<sup>1</sup> Department of Engineering Mathematics and Physics, Alexandria University, Alexandria 21544, Egypt

<sup>2</sup> Electronics and Communications Engineering Department, Egypt-Japan University of Science and Technology (E-JUST), Alexandria 21934, Egypt

<sup>3</sup> Department of Basic and Applied Science, Arab Academy for Science, Technology and Maritime Transport, AbuQuir Campus, P.O. Box 1029, Alexandria, Egypt

the elimination of the problem of computation at every node, which is usually a device with limited resources. The disadvantage of this scheme is the lack of the ability to access data in a proper way as well as inadequate scaling properties. The popular centralized scheme employs Semi-Definite Programming (SDP).

In distributed localization, sensor nodes perform the required computation themselves and communicate with each other to get their own location in the network [9–12]. The range-based schemes are defined by protocols that use absolute distance estimation for the location computation. Range-based techniques compute the precise distance between transmitting and receiving nodes on the basis of distance estimation methods [13, 14]. Thus, these techniques involve diverse estimation methods to compute the inter-node distance to measure their locality and then to calculate the position with the help of some principles of geometry. The range-free schemes make no assumptions about the accessibility of such information [8, 15]. Due to hardware restrictions of sensor nodes, solutions in range-free schemes are considered as cost effective such as the distance-vector (DV)-hop algorithm.

Many different optimization techniques are used to solve the problem of localization in wireless sensor networks. The most well-known optimization techniques for estimating coordinates of unknown nodes are bees optimization algorithm, bat algorithm, particle swarm optimization (PSO), butterfly optimization algorithm, genetic algorithms, squirrel search algorithm, shuffled shepherd optimization algorithm, etc.

In this paper, we focus on the three-dimensional space localization problem and choose the DV-Hop localization algorithm among other existing alternatives for the following reasons. First, three-dimensional (3-D) space in WSNs research is more realistic but is computationally more complicated. In real applications sensor nodes are distributed in 3-D space like in forests, in buildings, on mountains and in deep water. So, we were determined to stick to 3-D models and search for an approach that would reduce their computational complexity to an acceptable level. Second, the DV-Hop algorithm is our preferred localization algorithm because it is range free and distributed.

The contributions of this paper can be summarized as follows:

- Proposing an algorithm to solve the challenges of localization in 3-D space and obtaining better solutions in terms of accuracy compared to other existing standard algorithms. As a result of the improved accuracy, the cost is reduced since fewer number of anchor nodes is required which in turn leads to minimizing the power consumption as well.

- Our proposed algorithm is an enhancement to the existing DV-Hop algorithm that uses the shuffled shepherd algorithm in 3-D space in place of the least squares method employed in the traditional DV-Hop algorithm.
- In our model, a weighting factor of the squared error is included in the fitness function. This factor increases the accuracy of our proposed algorithm compared to other algorithms such as [16, 17] and [18].

The rest of this paper is organized as follows. Section 2 presents a literature review for some techniques used for solving the localization problem. The DV-Hop algorithm and the shuffled shepherd optimization algorithm are described in Sect. 3. Our proposed model is introduced in Sect. 4. Localization algorithms related performance measures and simulation results are provided in Sect. 5. Section 6 concludes the main results of the work.

## 2 Literature review

In this paper, our main focus is range-free based localization algorithms. Range-free schemes include five well-known schemes, which are the base for all more recent developments. These five schemes are:

- Approximate Point in Triangle (APIT) scheme.
- Distance Vector (DV)-Hop scheme.
- Multi-Hop scheme.
- Centroid scheme.
- Gradient scheme.

Approximate Point in Triangle scheme (APIT) performs localization in an area by dividing it into triangular zones between anchor nodes. The locations of anchor nodes are received and tested to determine whether they are outside or inside a common triangle. If it is inside, add to the set of insiders. When the target accuracy is reached, estimate the position of the unknown node as the center of gravity [19].

This paper focuses on the DV-Hop localization algorithm, which is considered as the classical distance vector routing method. Dragos Niculescu proposed the basic idea is to estimate the distance between the unknown nodes and the reference or anchor nodes as the product of the average hop size by the hop count. Each anchor node broadcasts a message which contains its position with hop count initially set equal to 1. Each receiving unknown node retransmits the received message to its neighbors after adjusting the hop count appropriately. Each unknown node keeps the minimum value received regarding the hop count. Consequently, it ignores other messages with higher values for the hop count [16, 20, 21]. Messages are

broadcast out with an increment of the hop count value at every middle hop.

Multi-Hop algorithm [22] uses connectivity information considering the nodes which are within the communication range. This scheme has three steps. First, the distance estimation between each viable pair of nodes is done. After that, multi-dimensional scaling (MDS) is used for deriving the locations to fit the estimated distance. Finally, optimization is done by putting the known locations into account. In large scale sensor networks, there are several kind of MDS methods such as metric, non-metric, classical, and weighted methods. In the centroid algorithm [23], a node location is computed on the basis of several neighboring anchor nodes positions. After receiving the location of all anchor nodes, an unknown node estimates its position as the average of the coordinates of all neighboring anchor nodes [8]. In the gradient localization algorithm [24, 25], unknown nodes obtain their locations through multi-iteration. It also uses hop count, which is initially set to zero and incremented as it is propagated to other nearby nodes. This algorithm consists of the following steps. First, each anchor node sends a message containing its coordinates and hop count value. Then, an unknown node determines the shortest path between itself and anchor nodes from which it receives beacon messages. The estimated distance can then be calculated and finally, minimum error is evaluated for each unknown node.

To solve the localization multi-dimensional optimization problem, numerous bio-inspired algorithms [26] have been considered. PSO is a population-based search algorithm in which individual particles work together to solve a given optimization problem [27]. The search is guided based on the global best fitness and all personal best fitness values. A genetic algorithm (GA) is another population-based algorithm that consists of four steps: initialization of population, fitness function evaluation for each individual, selection according to the fitness function, and finally recombination (mutation and crossover) [28]. As for the squirrel search algorithm (SSA) [29], it mimics the behavior of flying squirrels foraging. The squirrels search for food resources by gliding from one tree to the other when the weather is bad. There are three types of trees which are a hickory tree, acorn trees, and normal trees. SSA consists of the following stages: Random initialization, fitness evaluation, sorting, declaration, random selection, generation of new locations, seasonal monitoring condition, and random relocation at the end of the winter season. SSA has been successfully applied to solve the localization problem in [18].

The bacterial foraging algorithm (BFA) has been applied to solve the localization problem, where the nodes

which were localized in an iteration act as reference nodes for localizing the remaining nodes [30]. The benefit of the proposed technique was diminishing the amount of transmitted information to the base station which conserves the node energy.

To minimize the summation of the squared range error between the unknown nodes and anchor nodes, two dissimilar scattering of reference nodes were considered for the simulation of bees optimization algorithm (BOA) [31]. Moreover, for localizing the nodes in a 3-D area, the stochastic particle swarm optimization (SPSO) algorithm was proposed and the results show that it was more effective compared with PSO [32]. Furthermore, for improving the convergence rate and localization accuracy, the combination between particle swarm optimization and bacterial foraging optimization (BFO) was proposed in [33]. In addition, the genetic algorithm [34] was employed to reduce the total cost by equipping a small number of nodes with GPS systems. A variant of the cuckoo search algorithm was introduced in [35] to find the global optimum position of sensor nodes. Additionally, binary particle swarm optimization was used to reduce the total time of computation by using the received signal strength indicator (RSSI) to calculate the distance between unknown nodes and anchor nodes in [36].

For localizing more nodes and reducing the processing time, an improved bat algorithm has been introduced in [37]. A Parallel Firefly Algorithm (PFA) which uses the received signal strength (RSS) measurements for accurate self-localization of nodes was introduced in [38]. The objective function of their suggested algorithm was modified by taking distance factor into account. The processing time and global optimum value of the objective function of the proposed method were reduced compared to the PSO algorithm [38]. Artificial bee colony (ABC) is one of the optimization algorithms, which also provided a higher accuracy compared with PSO but with more processing time [39]. Additionally, to improve the performance of localization in WSNs, the butterfly optimization algorithm (BOA) was used in [40]. The simulation results showed that BOA improved the accuracy and decreased the computational time compared with the other algorithms.

A range-free based firefly optimization method for localizing the nodes in a 3-D environment was proposed in [41]. The sensor nodes were placed randomly over three levels. The reference nodes were scattered arbitrarily over the top level and unknown nodes were positioned in the intermediate and lowest levels. To decrease the computational complexity, the nonlinearity in the relation between the received signal strength indicator and distance was expressed using fuzzy logic. Furthermore, the location of

an unknown node was calculated using optimized edge weights between the anchor and unknown nodes. Their proposed algorithm accurately localized more unknown nodes as compared to existing techniques [41]. Moreover, for minimizing the total estimation error and for increasing the position accuracy, a modified DV-Hop algorithm was presented in [42] which was based on an improved hop size estimation.

### 3 Background

Localization approaches focus mainly on 2-D (two-dimensional) space. But, in real practice, nodes are often deployed in three-dimensional (3-D) space in WSNs such as in forests, in buildings, on mountains, in deep water, etc. In [43], a comprehensive survey of WSNs deployment in urban areas has been provided, from which it has been noticed that in almost all practical applications, nodes are deployed in 3-D space. Thus, localization in 3-D scenario is of more practical significance. The 3-D node localization problem in WSNs is much more complex and complicated in terms of computations. In the proposed model, 3-D space is considered, where the deployed area includes unknown nodes and anchor nodes. The proposed algorithm is based on the DV-Hop algorithm which is used to estimate the distance between the unknown nodes and the anchor nodes and it also determines the minimum hop count between the nodes. For minimizing the error in distance estimation between the nodes, we combined the DV-Hop algorithm with the shuffled shepherd optimization algorithm (SSOA), which is a newly developed swarm-based algorithm [44, 45].

In this section, we review the two building blocks of our proposed solution to the localization problem in 3-D space.

#### 3.1 Basic DV-Hop algorithm

The DV-Hop algorithm is a popular, range-free, distributed localization algorithm. This localization algorithm consists of the following three stages [16].

In stage 1, each anchor node floods the network with a beacon packet. This packet contains the coordinates of the anchor's position and hop-count which is initially set to 1. Each node maintains a table for all the received packets. This table contains the locations of the transmitting anchors and the minimum hop-count between the node and each anchor  $(x_i, y_i, z_i, hopcount_i)$  [16].

In stage 2, each anchor node utilizes the gathered information during the previous step to calculate its average hop size, where  $HopSize_i$  is the average hop size of anchor  $i$ ,  $m$  is the number of neighboring anchors to anchor  $i$ ,  $HopCount_{ij}$  is the minimum hop-count value between anchor  $i$  with coordinates  $x_i; y_i; z_i$  and anchor  $j$  with coordinates  $x_j; y_j; z_j$ . When every anchor node calculates its average hop size, it broadcasts it in the WSN [16]. Additionally, each unknown node will consider the first received average hop size as its hop size ( $HopSize_u$ ), which is assumed to be to the nearest anchor node, and ignores any latecomers.

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

The distance between each unknown node and every anchor node can be estimated as follows:

$$d_{ui} = HopSize_u \times HopCount_{ui}, \quad (2)$$

where  $d_{ui}$  and  $Hopcount_{ui}$  are the distance and the minimum hop-count value between the unknown node  $u$  and the anchor node  $i$ , respectively.

In stage 3, trilateration or multilateration is used with the least squares method to find the location of the unknown node. Assuming that each unknown node identifies  $n$  neighboring anchors with coordinates  $x_i; y_i; z_i$  and with distance  $d_{ui}$ , for  $i : 1; 2; 3; \dots; n$ , and  $n > 2$ ,

$$\begin{aligned} (x_u - x_1)^2 + (y_u - y_1)^2 + (z_u - z_1)^2 &= d_{u1}^2, \\ (x_u - x_2)^2 + (y_u - y_2)^2 + (z_u - z_2)^2 &= d_{u2}^2, \\ &\vdots \\ (x_u - x_n)^2 + (y_u - y_n)^2 + (z_u - z_n)^2 &= d_{un}^2. \end{aligned} \quad (3)$$

This linear system of equations can be written in matrix form as

$$\mathbf{Ax} = \mathbf{B}, \quad (4)$$

$$\mathbf{A}_{(n-1) \times 3} = \begin{pmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) & 2(z_1 - z_n) \\ 2(x_2 - x_n) & 2(y_2 - y_n) & 2(z_2 - z_n) \\ \vdots & \vdots & \vdots \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) & 2(z_{n-1} - z_n) \end{pmatrix}, \quad (5)$$

$$\mathbf{B}_{(n-1) \times 1} = \begin{pmatrix} x_1^2 + y_1^2 + z_1^2 - d_{u,1}^2 + d_{u,n}^2 - x_n^2 + y_n^2 + z_n^2 \\ x_2^2 + y_2^2 + z_2^2 - d_{u,2}^2 + d_{u,n}^2 - x_n^2 + y_n^2 + z_n^2 \\ \vdots \\ x_{n-1}^2 + y_{n-1}^2 + z_{n-1}^2 - d_{u,n-1}^2 + d_{u,n}^2 - x_n^2 + y_n^2 + z_n^2 \end{pmatrix}, \tag{6}$$

$$\mathbf{x}_{3 \times 1} = \begin{pmatrix} x_u \\ y_u \\ z_u \end{pmatrix}. \tag{7}$$

The solution of this inconsistent system represents the estimated locations of unknown nodes which can be given as:

$$\mathbf{x}^* = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{B}. \tag{8}$$

### 3.2 Shuffled shepherd optimization algorithm

The shuffled shepherd optimization algorithm (SSOA) [44, 45] is a new optimization algorithm that simulates animal instinct to find the best way to go to pasture. A shepherd uses one or more horses in a herd to move the tools and find the way. Moreover, it tries to guide sheep behind horses to pasture and bring them back. The sheep are divided into a number of herds and all the sheep are sorted by their objective function values in an ascending order and distributed randomly in each herd. Each herd has one shepherd and one horse and the movement of each sheep is updated according to the fitness function value. Next, based on that all the sheep in herds are merged again, sorted in ascending order and all these steps are repeated until the maximum number of iterations is reached.

The reasons for using the shuffled shepherd algorithm in our proposed model are summarized in what follows. First, the agents are distributed among more than one community, which helps providing better exploration of the enlarged search space due to the 3-D nature of the localization problem under study. The agents are then merged in each iteration and sorted according to the fitness function to balance exploration and exploitation. Second, the shuffled shepherd optimization algorithm considers both bad and good agents which leads to an overall improved performance [45].

A detailed description of the shuffled shepherd optimization algorithm is provided in what follows. In SSOA [44, 45], the sheep are divided into  $h$  herds and each herd has  $n$  sheep and the total number of sheep is  $s = h \times n$ . All

the sheep are sorted in ascending order according to the fitness function and the first  $h$  are selected and distributed randomly in herds and this is repeated for the next  $h$  sheep till all sheep are distributed. In each herd, sheep are sorted according to their fitness values. One agent is selected and considered a shepherd with an index  $i$ . There are some sheep worse and others better compared with the shepherd. The better are called horses and the others are considered the remaining sheep which need a guide. The shepherd leads the sheep to the horse which guides them to the pasture. The horse is denoted by  $d$  and the remaining sheep are denoted by  $j$ . The movement vector can be computed for each sheep as follows:

$$stepsize_i = \beta \times rand \times (X_d - X_i) + \alpha \times rand \times (X_j - X_i), \tag{9}$$

where  $X_d, X_i, X_j$  are the position vectors of the horse, the shepherd, and the selected sheep, respectively [45].  $rand$  is a random value in the interval  $[0, 1]$ ;  $\alpha$  parameter is equal to  $\alpha_0$  at the start of the algorithm, then it decreases with the increment in the iteration number of the algorithm to zero and can be obtained according to

$$\alpha = \alpha_0 - \frac{\alpha_0}{max\_iteration} \times iteration; \tag{10}$$

$\beta$  parameter is equal to  $\beta_0$  at the beginning of the algorithm then increases with the advance in iteration number of the algorithm until reaching  $\beta_{max}$ ;  $\beta$  can be calculated as

$$\beta = \beta_0 - \frac{\beta_{max} - \beta_0}{max\_iteration} \times iteration. \tag{11}$$

Now,  $X_i^{temporary}$  is computed as the temporary position vector for each sheep after calculating the step size as follows:

$$X_i^{temporary} = X_i^{old} + stepsize_i \tag{12}$$

If the objective function value of the temporary vector is not worse than the old objective function value, then the position of the sheep is updated, so we have  $X_i^{new} = X_i^{temporary}$ , otherwise  $X_i^{new} = X_i^{old}$ .

The Steps of the Shuffled Shepherd Optimization Algorithm (SSOA) are summarized as follows:

1. The parameters  $\alpha$  and  $\beta$  are initialized, and the initial position of the  $i$ th sheep is determined randomly in an  $m$ -dimensional search space according to the following equation [44]:

$$X_i^0 = X_{min} + rand \times (X_{max} - X_{min}), \quad i = 1, 2, \dots, m \tag{13}$$

where  $X_i^0$  is the initial position vector of the  $i$ th sheep,  $X_{max}$  and  $X_{min}$  are the maximum and minimum boundaries, respectively.

2. The fitness function for each sheep is evaluated and the sheep are sorted in ascending order of their fitness values.
3. Build the herds as we discussed before.
4. The step size for each sheep and the temporary position vector are calculated and compared with the old position depending on the fitness function.
5. Update the position of each sheep; if the temporary objective function is not worse than the old objective function, then the position of the sheep is updated, otherwise keep the previous position.
6. Finally, merge the herds and sort the sheep in ascending order according to the fitness function and update the parameters values  $\alpha$  and  $\beta$ .

These steps are repeated until the maximum number of iterations is reached to obtain the best solution.

### 4 Proposed localization algorithm

The key step to developing our proposed localization methodology is to select a suitable fitness function to guide the search in the shuffled shepherd algorithm. Towards this end, we proceed to evaluate the real or actual distance between anchors  $i$  and  $j$ , which is given by

$$d_{ij}^r = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}, \tag{14}$$

and compare it with the estimated distance, which is computed as in [46]

$$d_{ij}^{est} = Hopsize_i^l \times Hopcount_{ij}, \tag{15}$$

where  $Hopsize_i^l$  is the improved anchor node’s average hop size. The improved average hop size of an anchor node is defined by

$$Hopsize_i^l = \frac{\sum_{j \neq i, j=1}^m d_{ij}^r \times Hopcount_{ij}}{\sum_{j \neq i, j=1}^m Hopcount_{ij}^2}. \tag{16}$$

The average improved hop size of the unknown node can be calculated as:

$$Hopsize_u^l = \frac{1}{n} \sum_{i=1}^n Hopsize_i^l. \tag{17}$$

The expression for the improved anchor node hop size is obtained by minimizing the sum of squared errors between the real distances and the estimated distances for anchor node  $i$  to all its  $m$  neighboring anchors:

$$\sum_{j \neq i, j=1}^m e_{ij}^2 = \sum_{j=1}^m \left( d_{ij}^r - Hopsize_i^l \times Hopcount_{ij} \right)^2. \tag{18}$$

The  $Hopsize_i^l$  can be evaluated by taking derivative with respect to  $Hopsize_i^l$  and equating it to zero, then we get

$$\sum_{j \neq i, j=1}^m \left( d_{ij}^r \times Hopcount_{ij} - Hopsize_i^l \times Hopcount_{ij}^2 \right) = 0. \tag{19}$$

The fitness function for the proposed shuffled shepherd based methodology is given as

$$f(x_u^*, y_u^*, z_u^*) = \sum_{i=1}^n w_i \left( \sqrt{(x_u^* - x_i)^2 + (y_u^* - y_i)^2 + (z_u^* - z_i)^2} - d_{ui} \right)^2, \tag{20}$$

where the number of neighboring anchor nodes is at least 2 for each unknown node.  $w_i$  is a weighting factor of the squared error,  $(x_u^*, y_u^*, z_u^*)$  represents the coordinates of the unknown node  $u$  and  $d_{ui}$  is the estimated distance between anchor node  $i$  and unknown node  $u$ , respectively. The weighting factor of anchor node  $i$  can be obtained as:

$$w_i = \left( \frac{Hopcount_{ij}^2}{n \times Hopcount_{ui}} \right)^{\frac{R}{Hopsize_u^l}}, \tag{21}$$

where  $R$  is the communication range [18]. The weighting factor is used to increase the impact of nearby nodes and diminish the effect of far away nodes.

The fitness value is then calculated for every sheep by considering its position vector as the coordinates of the unknown node  $(x_u^*, y_u^*, z_u^*)$ . Algorithm 1 provides the pseudo-code of our proposed method referred to as SSOIDV-Hop in the rest of the paper.

**Algorithm 1** The SSODV-Hop pseudocode

---

**Input:** Parameters of WSN and SSOA (see Table 1 and Table 2)  
**Output:** The unknown nodes estimated position

---

- 1: Input the parameters of WSN
- 2: Sensor nodes are deployed randomly and the anchor nodes broadcast their messages
- 3: Anchor nodes calculate their improved average hop size by (16) and broadcast them in the network
- 4: **for**  $u = 1$  : number of unknown nodes **do**
- 5:     Calculate the improved average hop size of the unknown node ( $HopSize_u^I$ ) using (17)
- 6:     Calculate the distances between the unknown node and all the neighboring anchors using (2)
- 7:     **if** unknown node estimates the distance to more than two anchors **then**
- 8:         Apply SSOA
- 9:         The parameters  $\alpha$  and  $\beta$  are initialized
- 10:         the initial position of the  $i$ th sheep is determined randomly (13)
- 11:         The fitness function for each sheep is evaluated
- 12:         The sheep are sorted in ascending order of their fitness values using (20)
- 13:         **for**  $t=1:T_{max}$  **do**
- 14:             Build the herds as we discussed before
- 15:             The step size for each sheep and the temporary position vector are calculated
- 16:             **if** the temporary objective function is not worse than the old objective function **then**
- 17:                 The position of the sheep is updated
- 18:                 else keep the previous position
- 19:             **end if**
- 20:             Merge the herds and sort the sheep in ascending order according to fitness function
- 21:             Update the parameters values  $\alpha$  and  $\beta$
- 22:             **end for**
- 23:         The estimated positions of the unknown nodes is the location of the sheep with the lowest fitness function
- 24:     **end if**
- 25: **end for**

---

## 5 Experimental results

The proposed model is compared with the standard DV-Hop algorithm, improved DV-Hop, and squirrel search algorithm based improved DV-Hop algorithm in 3-D space. The error in the estimated location for each unknown node is evaluated. In our simulations, the nodes are randomly located in a deployment region  $100 \text{ m} \times 100 \text{ m} \times 100 \text{ m}$ , as shown in Fig. 1.

### 5.1 Accuracy measures

Comparing the algorithms is important for some applications to prove the efficiency of the proposed model. Moreover, accuracy is an important metric for measuring the degree of matching between the estimated and actual locations for each unknown node. The location error (LE) is evaluated as follows:

$$LE = \sqrt{(x_u^* - x_u)^2 + (y_u^* - y_u)^2 + (z_u^* - z_u)^2}, \quad (22)$$

where  $(x_u^*, y_u^*, z_u^*)$  and  $(x_u, y_u, z_u)$  are the estimated and the actual location of the unknown node  $u$ , respectively. The perfect indication for the accuracy is measured by finding

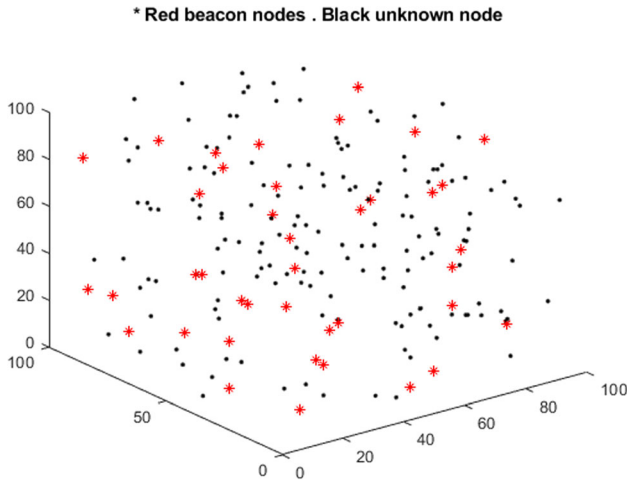


Fig. 1 Nodes distribution (anchor nodes and unknown nodes)

the average localization error (ALE), which can be calculated as follows:

$$ALE = \frac{1}{R N} \sum_{u=1}^N \sqrt{(x_u^* - x_u)^2 + (y_u^* - y_u)^2 + (z_u^* - z_u)^2}, \tag{23}$$

where  $N$  is the number of unknown nodes.

### 5.2 Simulation parameters

In the simulation, localization algorithms are implemented using MATLAB R2017B. The average number of independent runs of each result is 100. The parameters of the WSN, the parameters of the proposed model, and the parameters of SSA in 3-D are presented in Tables 1, 2, and 3, respectively. The parameters of other compared algorithms are kept as in their original papers.

### 5.3 Simulation results

The effect of varying some parameters on the performance of the proposed model is considered in this subsection. These include varying the ratio of the anchor nodes, varying nodes count, and the communication range. The

Table 1 The parameters of the WSN

Parameter	Value
Deployment area	100 m × 100 m × 100 m
Number of nodes ( $N$ )	50–400
Ratio of anchors	5–40%
Communication range ( $R$ )	15–45 (m)

Table 2 The parameters of the SSOA

Parameter	Value
Number of sheep ( $s$ )	50
Number of herds ( $h$ )	5
$\alpha_0$	0.5
$\beta_0$	1
$\beta_{max}$	3.5
Max number of iterations	200

Table 3 The parameters of the SSA

Parameter	Value
Number of FSs ( $NP$ )	50
Number of nutritious food source ( $N_{fs}$ )	5
Max number of iterations ( $T_{max}$ )	200

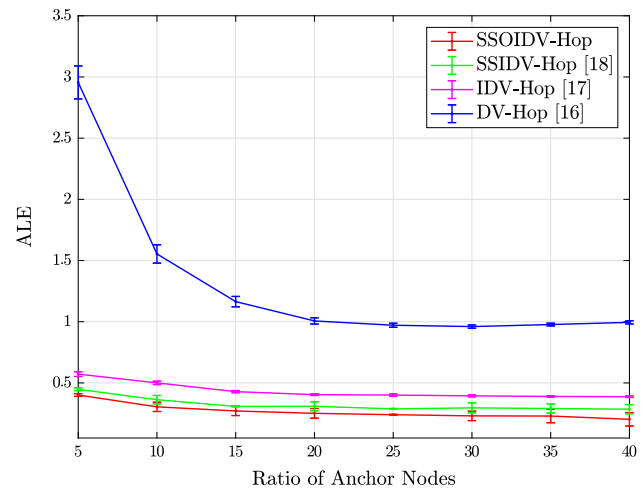


Fig. 2 Percentage of anchor nodes effect on the ALE

comparison considers the proposed model, standard DV-Hop, IDV-Hop, and squirrel search algorithm (SSA).

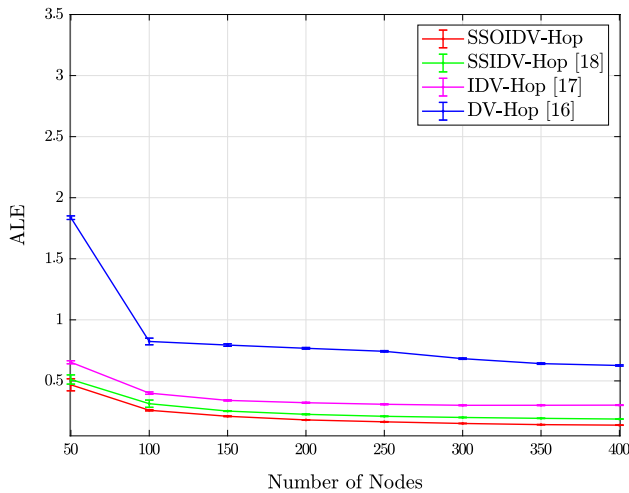
#### 5.3.1 The effect of varying the ratio of anchor nodes

When the number of anchor nodes increases in a WSN, the unknown node collects the data from more neighboring anchor nodes. Therefore, the anchor nodes with a small hop-count to the unknown node increase, and the hop-count value decreases, which is expected to reduce the ALE. In the simulation, 200 nodes are distributed with a communication range of 30 m. The percentage of anchor nodes is varied from 5% to 40%. Figure 2 presents a 95% confidence interval (CI) variation with the anchor node



**Table 4** Mean and tolerance of ALE for varying beacon nodes ratio

Localization algorithm	DV-Hop	IDV-Hop	SSIDV-Hop	SSOIDV-Hop
Average value of ALE	1.3255	0.4348	0.3236	0.2715
Average value of tolerance for 95% CI	±0.0414	±0.0099	±0.0257	±0.0346



**Fig. 3** Number of nodes effect on the ALE

ratio. The small width of the interval is an indicator of the stability of the algorithm. The standard DV-Hop achieves the highest ALE due to the performance of multilateration step. On the other hand, the lowest average localization error is achieved by our proposed SSOA.

In Table 4, the average values of the ALE for the different localization algorithms and the average values of the tolerance, given by ( $tolerance = \frac{1.96 * \sigma}{\sqrt{N}}$ ), are presented. Our proposed SSOIDV-Hop improved the accuracy on average by 80% compared to DV-Hop, 38% compared to the improved DV-Hop, and 16% compared to SSIDV-Hop.

**Table 5** Mean and tolerance of ALE for varying beacon nodes ratio

Localization algorithm	DV-Hop	IDV-Hop	SSIDV-Hop	SSOIDV-Hop
Average value of ALE	0.9753	0.36565	0.262	0.21805
Average value of tolerance for 95% CI	±0.01022	±0.005496	±0.0101	±0.00814

### 5.3.2 The effect of varying the number of nodes

The minimum hop-count paths between nodes take the form of straight lines when the number of nodes increases. This improves the estimation of the average hop size of each anchor node which is expected to decrease the ALE. Figure 3 shows the average localization error for different nodes density. ALE is calculated based on the variation of node density from 50 to 400 with 20% anchor nodes. The communication range for each node is considered as 40 m. It can be observed from Fig. 3 that the average localization error decreases with the increase in the number of sensor nodes. This is due to the fact that when the number of sensor nodes increases, this causes an increase in network connectivity which leads to the collection of large information amount about the location of nodes, which improves the localization accuracy. It can be seen from the result that the proposed algorithm performs better in terms of localization accuracy as compared to other algorithms.

Table 5 presents the average values of the ALE for the different localization algorithms and the average values of tolerance. Our proposed SSOIDV-Hop has better accuracy on average by 77.6% compared to standard DV-Hop, 40.4% compared with improved DV-Hop, and 16.8% compared to SSIDV-Hop. The tolerance value is small enough to ensure the stability of the proposed methodology.

### 5.3.3 Variation in communication range effect on ALE

All localization algorithms exhibit an improvement in accuracy when the communication range of a node is increased from 20 to 30 m. The reason for this is that the

number of one-hop neighbor anchors increases when the communication range (R) increases. Also, it has an effect on decreasing the minimum hop-count value between nodes. In order to study the effect of variation in communication range on localization accuracy, we considered 400 sensor nodes with 20 % anchor nodes. To examine the performance of our proposed algorithm, ALE is calculated with variation in the communication range of nodes from 20 to 45 m. Figure 4 shows the average localization error for different communication ranges of the sensor nodes. Simulation results show that the localization error decreases with the increase in communication range. It is due to the fact that as the communication range increases, network connectivity also increases, which leads to better coverage and good location estimation of nodes in the network.

In Table 6, the average values of the ALE for the different localization algorithms and the average values of tolerance are provided. Our proposed SSOIDV-Hop improves accuracy on average by 58.7% compared to DV-Hop, 20% compared with improved DV-Hop and 7.3% compared to SSIDV-Hop.

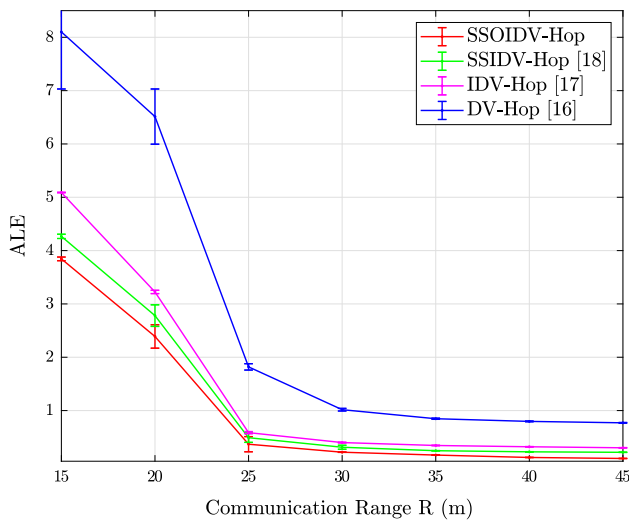


Fig. 4 Varying in communication range effect on the ALE

Table 6 Mean and tolerance of ALE for varying beacon nodes ratio

Localization algorithm	DV-Hop	IDV-Hop	SSIDV-Hop	SSOIDV-Hop
Average value of ALE	1.1728	1.2203	1.4669	2.8378
Average value of tolerance for 95% CI	±0.05784	±0.053645	±0.01074	±0.24227

## 6 Conclusions

The proposed model is based on the DV-Hop algorithm with the meta-heuristic technique called the shuffled shepherd optimization algorithm, which improved the estimation of the average hop size of each unknown node. The fitness function employed depends on weighting factors of the squared error. Our model considers the localization problem in 3-D space, which is a more realistic assumption. The proposed model achieves the least average localization error compared to the standard DV-Hop algorithm, improved DV-Hop, and the squirrel search algorithm. The proposed model shows more accuracy and more stability compared to other algorithms.

**Funding** Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

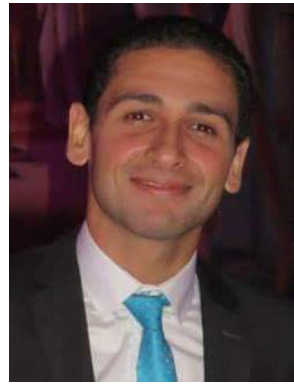
## References

- Aldeer, M. (2013). A summary survey on recent applications of wireless sensor networks (pp. 485–490). <https://doi.org/10.1109/SCoREd.2013.7002637>.
- Kulkarni, R. V., & Venayagamoorthy, G. K. (2011). Particle swarm optimization in wireless-sensor networks: A brief survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(2), 262–267. <https://doi.org/10.1109/TSMCC.2010.2054080>
- Hofmann-Wellenhof, B., Lichtenegger, H., & Collins, J. (2001). *Global Positioning System*. <https://doi.org/10.1007/978-3-7091-6199-9>.

4. Mohar, S.S., Goyal, S., & Kaur, R. (2018). A survey of localization in wireless sensor network using optimization techniques. In *2018 4th International Conference on Computing Communication and Automation (ICCCA)* (pp. 1–6). <https://doi.org/10.1109/CCAA.2018.8777624>
5. Shang, Y., Ruml, W., Zhang, Y., & Fromherz, M.P.J. (2003). Localization from mere connectivity. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing*.
6. Kannan, A., Mao, G., & Vucetic, B. (2006). Simulated annealing based wireless sensor network localization. *Journal of Computers*. <https://doi.org/10.4304/jcp.1.2.15-22>
7. Alippi, C., & Vanini, G. (2006). A rssi-based and calibrated centralized localization technique for wireless sensor networks. In: *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)* (pp. 5–305). <https://doi.org/10.1109/PERCOMW.2006.13>
8. 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>
9. Langendoen, K., & Reijers, N. (2003). Distributed localization in wireless sensor networks: A quantitative comparison. *Computer Networks*, 43(4), 499–518. [https://doi.org/10.1016/S1389-1286\(03\)00356-6](https://doi.org/10.1016/S1389-1286(03)00356-6)
10. Moore, D., Leonard, J., Rus, D., & Teller, S. (2004). Robust distributed network localization with noisy range measurements. In *SenSys'04 - Proceedings of the Second International Conference on Embedded Networked Sensor Systems*. <https://doi.org/10.1145/1031495.1031502>
11. Savarese, C., Langendoen, K., & Rabaey, J. (2002). Robust positioning algorithms for distributed Ad-Hoc wireless sensor networks. In: *2002 USENIX Annual Technical Conference (USENIX ATC 02)*. USENIX Association, Monterey, CA. <https://www.usenix.org/conference/2002-usenix-annual-technical-conference/robust-positioning-algorithms-distributed-ad-hoc>
12. Alrajeh, N. A., Bashir, M., & Shams, B. (2013). Localization techniques in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 9(6), 304628. <https://doi.org/10.1155/2013/304628>
13. Hussein, A., Al-Qaysi, H., & Rahman, T. (2014). A survey on secure range based localization algorithms in wireless sensor networks. *Research Journal of Recent Sciences*, 3(11), 103–109.
14. Youssef, A.M., & Youssef, M. (2007). A taxonomy of localization schemes for wireless sensor networks. In *ICWN* (pp. 444–450).
15. Nagaraju, S., Gudino, L., Tripathi, N., Vidhyadharan, S., & Ramesha, C. K. (2018). Mobility assisted localization for mission critical wireless sensor network applications using hybrid area exploration approach. *Journal of King Saud University-Computer and Information Sciences*. <https://doi.org/10.1016/j.jksuci.2018.04.008>
16. Kumar, S., & Lobiyal, D. K. (2017). Novel DV-Hop localization algorithm for wireless sensor networks. *Telecommunication Systems: Modelling, Analysis, Design and Management*, 64(3), 509–524. <https://doi.org/10.1007/s11235-016-0189-8>
17. Chen, Y., Li, X., Ding, Y., Xu, J., & Liu, Z. (2018). An improved dv-hop localization algorithm for wireless sensor networks. In *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)* (pp. 1831–1836). <https://doi.org/10.1109/ICIEA.2018.8398006>
18. Abd El Ghafour, M. G., Kamel, S. H., & Abouelseoud, Y. (2021). Improved dv-hop based on squirrel search algorithm for localization in wireless sensor networks. *Wireless Networks*, 27(4), 2743–2759.
19. He, T., Huang, C., Blum, B.M., Stankovic, J.A., & Abdelzaher, T. (2003). Range-free localization schemes for large scale sensor networks. In *Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking, MobiCom 2003 ; Conference date: 14-09-2003 Through 19-09-2003* (pp 81–95). <https://doi.org/10.1145/938985.938995>
20. Niculescu, D., & Nath, B. (2003). DV based positioning in ad hoc networks. *Telecommunication Systems*, 22, 267–280. <https://doi.org/10.1023/A:1023403323460>
21. Salama, M., & Kandil, M. (2016). An improved dv-hop localization algorithm based on modified hop-size. In *2016 World Symposium on Computer Applications & Research (WSCAR)* (pp. 83–86).
22. Savvides, A., Park, H., & Srivastava, M.B. (2002). The bits and flops of the n-hop multilateration primitive for node localization problems (pp. 112–121). <https://doi.org/10.1145/570738.570755>
23. Bulusu, N., Heidemann, J., & Estrin, D. (2000). Gps-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5), 28–34.
24. Poovendran, R., Roy, S., & Wang, C. (2007). *Secure localization and time synchronization for wireless sensor and Ad Hoc networks*. Springer.
25. Priyantha, N.B., Balakrishnan, H., Demaine, E., & Teller, S. (2003). Anchor-free distributed localization in sensor networks (pp. 340–341). <https://doi.org/10.1145/958491.958550>
26. Singh, A., Sharma, S., & Singh, J. (2021). Nature-inspired algorithms for wireless sensor networks: A comprehensive survey. *Computer Science Review*, 39, 100342.
27. Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks* vol. 4, (pp. 1942–1948). IEEE.
28. Holland, J. H. (1980). Adaptive algorithms for discovering and using general patterns in growing knowledge bases. *International Journal of Policy Analysis and Information Systems*, 4(3), 245–268.
29. Jain, M., Singh, V., & Rani, A. (2019). A novel nature-inspired algorithm for optimization: Squirrel search algorithm. *Swarm and Evolutionary Computation*, 44, 148–175. <https://doi.org/10.1016/j.swevo.2018.02.013>
30. Kulkarni, R., Venayagamoorthy, G., & Cheng, M. (2009). Bio-inspired node localization in wireless sensor networks. In *Proceedings 2009 IEEE International Conference on Systems, Man and Cybernetics, SMC 2009* (pp. 205–210). <https://doi.org/10.1109/ICSMC.2009.5346107>
31. Moussa, A., & El-Sheimy, N. (2010). Localization of wireless sensor network using bees optimization algorithm. In *The 10th IEEE International Symposium on Signal Processing and Information Technology* (pp. 478–481).
32. Zhang, Z., & Cui, H. (2012). Localization in 3D sensor networks using stochastic particle swarm optimization. *Wuhan University Journal of Natural Sciences*. <https://doi.org/10.1007/s11859-012-0884-6>
33. Rani, S., Babbar, H., Kaur, P., Alshehri, M. D., & Shah, S. H. A. (2022). An optimized approach of dynamic target nodes in wireless sensor network using bio inspired algorithms for maritime rescue. *IEEE Transactions on Intelligent Transportation Systems*. <https://doi.org/10.1109/TITS.2021.3129914>
34. Assis, A.F., Vieira, L.F.M., Rodrigues, M.T.R., & Pappa, G.L. (2013). A genetic algorithm for the minimum cost localization problem in wireless sensor networks. In *2013 IEEE Congress on Evolutionary Computation* (pp. 797–804). <https://doi.org/10.1109/CEC.2013.6557650>
35. Zefan, C., & Xiaodong, Y. (2017). Cuckoo search algorithm with deep search. In *2017 3rd IEEE International Conference on Computer and Communications (ICCC)* (pp. 2241–2246). <https://doi.org/10.1109/CompComm.2017.8322934>

36. Zain, I.F.M., & Shin, S.Y. (2014). Distributed localization for wireless sensor networks using binary particle swarm optimization (bpso). In *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)* (pp. 1–5). <https://doi.org/10.1109/VTCSpring.2014.7022886>.
37. Goyal, S., & Patterh, M. (2015). Modified bat algorithm for localization of wireless sensor network. *Wireless Personal Communications*. <https://doi.org/10.1007/s11277-015-2950-9>
38. Sai, V.-O., Shieh, C.-S., Nguyen, T.-T., Lin, Y.-C., Horng, M.-F., & Le, Q.-D. (2015). Parallel firefly algorithm for localization algorithm in wireless sensor network. In *2015 Third International Conference on Robot, Vision and Signal Processing (RVSP)* (pp. 300–305) <https://doi.org/10.1109/RVSP.2015.78>.
39. Kulkarni, V.R., Desai, V., & Kulkarni, R.V. (2016). Multistage localization in wireless sensor networks using artificial bee colony algorithm. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1–8). <https://doi.org/10.1109/SSCI.2016.7850273>.
40. Mohar, S.S., Goyal, S., & Kaur, R. (2018). A survey of localization in wireless sensor network using optimization techniques. In *2018 4th International Conference on Computing Communication and Automation (ICCCA)* (pp. 1–6). <https://doi.org/10.1109/CCAA.2018.8777624>.
41. Kumar, A., & Karampal. (2015). Optimized range-free 3d node localization in wireless sensor networks using firefly algorithm. In *2015 International Conference on Signal Processing and Communication (ICSC)* (pp. 14–19). <https://doi.org/10.1109/ICSPCom.2015.7150611>.
42. Peng, B., & Li, L. (2015). An improved localization algorithm based on genetic algorithm in wireless sensor networks. *Cognitive Neurodynamics*, 9, 249–256.
43. Rashid, B., & Rehmani, M. H. (2016). Applications of wireless sensor networks for urban areas: A survey. *Journal of Network and Computer Applications*, 60, 192–219. <https://doi.org/10.1016/j.jnca.2015.09.008>
44. Kaveh, A., & Zaerreza, A. (2020). Shuffled shepherd optimization method: A new meta-heuristic algorithm. *Engineering Computations*, 37, 2357–2389. <https://doi.org/10.1108/EC-10-2019-0481>
45. Kaveh, A. (2014). *Advances in metaheuristic algorithms for optimal design of structures*. Springer.
46. Li, G., Zhao, S., Wu, J., Li, C., & Liu, Y. (2019). DV-Hop localization algorithm based on minimum mean square error in internet of things. *Procedia Computer Science*, 147, 458–462. <https://doi.org/10.1016/j.procs.2019.01.272>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Ahmed El. Gouda** was born in Alexandria, Egypt in 1986 and received the B.Sc. received in Communications and Electronics Engineering and M.Sc.Eng. degrees in Engineering Mathematics from the Department of Engineering Mathematics and Physics from the Faculty of Engineering, Alexandria University, Egypt in 2008 and 2018, respectively. Currently, he is pursuing his Ph.D. degree in Engineering Mathematics at Alexandria University. Engineering Ahmed Gouda is a Teaching Assistant (part time) with the Basic and Applied Science institute in the college of Engineering, Arab Academy For Science, Technology and Maritime Transport. His research interests include applied mathematics, optimization techniques and wireless sensor networks.



**Yasmine Abouelseoud** was born in Alexandria, Egypt, in 1978. She received the B.Sc. degree in Computer Science, M.Sc. and Ph.D. degrees in Engineering Mathematics from Alexandria University in Egypt in 2001, 2005, and 2008, respectively. She is currently a full professor at the Engineering Mathematics and Physics Department, Faculty of Engineering, Alexandria University, Egypt. Her main research interests are in information security and engineering optimization. She has over sixty publications in these fields. Her papers co-authored with her students received best papers/best student paper awards in several international conferences (IEOM 2012, IEOM 2015, WCECS, 2012).



**Ahmed H. Abd El-Malek** received the B.Sc. and M.Sc. degrees in electrical engineering from Alexandria University, Egypt in 2007 and 2010, respectively. He received his Ph.D. degree from Electrical Engineering Department, King Fahd University of Petroleum & Mineral (KFUPM), Saudi Arabia in 2016. Currently, he is an associate professor in the Electronics and Communications Engineering Department, Egypt-Japan University of Science and Technology. His research interests are cognitive radio, design and analysis of wireless networks, network coding, physical layer security and interference cancellation.



of Engineering, Arab Academy For Science, Technology and

**Radwa Ahmed Osman** received the B.Sc.Eng. and M.Sc.Eng. degrees in Electrical and Communications Engineering from Arab Academy For Science, Technology and Maritime Transport, Alexandria, Egypt in 2007 and 2010, respectively, and the Ph.D. degree in Electronic Engineering from the University of Aston, Birmingham, U.K., in 2017. Dr. Ahmed Osman is an associate professor with the Basic and Applied Science institute in the college

Maritime Transport. Her research interests include wireless communications & networks, vehicular communications, optimization, machine learning, deep learning, quality assessment and numerical solutions.