




# A three-dimensional wireless sensor network with an improved localization algorithm based on orthogonal learning class topper optimization

Tapan Kumar Mohanta<sup>1</sup> · Dushmanta Kumar Das<sup>2</sup> 

Received: 28 December 2021 / Accepted: 2 January 2023 / Published online: 20 January 2023  
© Institut Mines-Télécom and Springer Nature Switzerland AG 2023

## Abstract

Numerous sensor network applications require accurate and rapid localization of randomly deployed sensor nodes. For wireless sensor network (WSN) localization, optimization methods can provide specific and reliable position estimates of a sensor node. The fixed density of beacons may be increase or decrease owing to various reasons, such as upkeep, lifespan, and breakdown. Because of its robustness, flexibility, and economic viability, the distance vector-hop (DV-Hop) algorithm is used to locate WSN nodes. Because of its high precision and fast computing speed, class topper optimization (CTO) is suitable to solve localization problems. This study proposes an orthogonal learning CTO-based DV-Hop localization algorithm for three-dimensional WSNs. Moreover, this study used a refined formula to calculate the minimum hop size of beacon nodes for reducing localization errors (LEs) in the approximated distance between the beacon and dumb nodes. Results revealed that our proposed method outperformed some existing algorithms in terms of reducing LEs (0.6%) and localization error variance (0.3%) and enhancing localization accuracy (0.4%) and coverage (0.7%).

**Keywords** 3D DV-Hop · OLCTO · WSN · Localization error · LabVIEW software

## 1 Introduction

Wireless sensor networks (WSNs) have attracted considerable research attention. WSNs are composed of numerous small sensor devices called sensor nodes that gather and communicate data from the environment [1]. WSN localization of collected usable data is a critical problem. Furthermore, without determining actual location-related information within each sensor node, the obtained data are not useful [2]. Moreover, the location of a data-sensing sensor is as crucial as the data itself. Thus, the data

and position-related information must be jointly communicated to determine the precise location of information collected by the sensor node. Several applications designed for use in extreme environments are increasingly using WSNs, such as those employed in battlegrounds, border patrols, and rain forests [3, 4] and those used for the remote control of hazardous areas and routing [5, 6]. Although the global positioning system is commonly used for sensor localization, it is not a viable choice because of the high cost of WSNs. Thus, researchers have proposed various WSN localization algorithms [7]. Numerous localization algorithms currently available are classified into two types: range based and range free [8].

To calculate a node's location, range-free localization algorithms are used to estimate the distance within nodes and do not require external hardware. For these types of algorithms, the closest node whose position is known is used to localize dumb nodes. The centroid algorithm [9], DV-Hop [10, 11], amorphous [12], approximate point in triangulation test [13], and multidimensional scaling techniques [14] are some of the well-known range-free localization algorithms. However, the existing range-free localization algorithms have many limitations. For example,

---

✉ Dushmanta Kumar Das  
dushmantakumardas29@gmail.com

Tapan Kumar Mohanta  
tapank838@gmail.com

<sup>1</sup> Department of ECE, ICFAI University Tripura,  
Kamalghat, 799210, India

<sup>2</sup> Department of Electrical and Electronics Engineering,  
National Institute of Technology Nagaland,  
Dimapur, 797103, India

these algorithms are not resistant to environmental noise interference, require long computational time to determine sensor node positions, and provide only coarse-grained precision. Range-based localization algorithms require precise measurement methods that necessitate expensive equipment to determine the distance or angle between a node and its counterpart, providing highly accurate location-related information [15]. Received signal strength indicator [16], time of arrival [17], angle of arrival of signals packets [18], and time difference of arrival [19] are some examples of range-based localization algorithms. We focus on range-free algorithms in this study because they do not require additional devices.

Range-free localization techniques do not require any hardware, making them less expensive than range-based localization techniques. Furthermore, to reduce the localization error (LE), appropriate localization optimization methodologies must be developed in conjunction with a range-free localization algorithm. Although a localization method involving the use of a high percentage of localized nodes would be adequate for localization, it does not guarantee fine-grained precision. Thus, optimization algorithms should be applied to the results of localization techniques to reduce the LE. Optimization algorithms are useful in practical applications. However, with the availability of an increasing amount of information in the IoT era, many real-world problems include multiple decision parameters and evaluation indicators. Single-objective optimization algorithms cannot efficiently solve such problems. Thus, multi-objective optimization algorithms have been proposed and are used in diverse fields. Most studies have used single-objective optimization algorithms to solve localization problems [20]. However, numerous studies have demonstrated the importance of multiobjective optimization algorithms in solving multiobjective complex problems [21]. Although traditional studies have focused on two-dimensional space, most sensors have been deployed in three-dimensional space for various applications, necessitating further research to overcome complex challenges.

Several optimization techniques have been developed to solve real-time optimization problems. However, these techniques are not useful for solving large-scale, complex problems in WSNs. The class topper optimization (CTO) method is proposed in this study to overcome existing limitations. The CTO algorithm presented in [22] is a population-based meta-heuristic optimization technique. On the basis of students' learning behaviors, a CTO is motivated to become the best student in the class (class topper). A class is divided into sections where students learn regarding various subjects. Exams are conducted each year to evaluate each student's performance in a specific class. Section toppers are students who receive the highest marks in a section, whereas class toppers receive the highest marks

in a class. Students are always striving to improve their knowledge to be the best student possible. To become a class topper, the CTO algorithm mimics students' unique learning behaviors. During the process of becoming a class topper in CTO, students enhance their knowledge with the help of the knowledge of their respective section topper.

Contribution of this study:

- i We discovered a novel formula to optimize the average hop-distance calculation among nodes in a 3D wireless sensor network.
- ii An orthogonal learning-based CTO (OLCTO) focused on a DV-Hop localization algorithm was developed to reduce the LE in 3D WSNs.
- iii The efficiency and reliability of the algorithm were improved by implementing the optimization process by using the OLCTO approach.
- iv Simulation results revealed that our new multiobjective OLCTO method outperforms some existing approaches.

## 2 Associated works

Numerous optimization-based algorithms have been developed to solve localization problems. In [20], an NSGA-II-based multiobjective DV-Hop localization algorithm was proposed; this study applied an improved constraint strategy to all beacon nodes to improve DV-Hop location estimation precision. This algorithm exhibited greater accuracy with a small increase in computational costs. Kaur et al. [23] proposed Wolf algorithm-based optimization to accurately estimate the average hop distance for each beacon node. The algorithm provided precise results with a small increase in the computation cost. Another study [24] recommended using multiobjective functions to solve difficult problems involving several objects and track them in order to reduce the model's size. However, that study measured the proportion of selection to determine the global optimum solution. The effect of radio irregularity was not examined in that study. Kanwar and Kumar [25] suggested a range-free localization approach using runner-root algorithms. They changed the average hop size of anchor nodes by refining a correction factor and optimized the hop size by using a line search algorithm. The precision of localization was improved using a runner-root method.

The genetic method was used in [26] to improve range-free localization. The authors introduced a correction factor to alter the hop size of anchor nodes, which was then further improved using a line search algorithm. However, holes, nonuniform node distribution, network sparsity, and irregular radio patterns may reduce the suggested algorithm's performance. Using PSO, Singh and Sharma [27] proposed an improved DV-Hop localization algorithm.

They measured the hop count and minimum hop size by using the proposed algorithm to determine the location and evaluate performance errors. The developed method utilized the hop size of the anchor from which the dumb node defined its reach. In addition, with the PSO strategy, the positions of unknown nodes can be determined. The dynamic-range average bisector approach was proposed in [26] to identify the location of nodes. In the system, each anchor node transmits data at two range levels. The correction factor was employed to prevent the overlapping of regions at the expense of increased computational costs.

A multipath routing scheme for homogeneous WSNs was presented in [28]. The proposed routing method was used to reduce energy consumption and balance load, resulting in increased network lifetime. We aim to reduce the packet loss rate in this work. Clustering network nodes, exploring paths between CHs, and maintaining paths are the three phases of the proposed routing method. A two-level routing algorithm with two phases, clustering and routing, was proposed in [29]. CHs are chosen during the clustering phase on the basis of residual energy, length from the BS, centrality, and the number of neighbors. BCHs and CHs are chosen during this phase. Each clustering setup phase has two clustering-steady phases, as determined by BCH selection. Thus, clustering's overhead is minimized. Routing is the second phase, which has two levels: intra cluster and inter cluster. Each cluster is divided into four sections, each with a CH centrality, and nodes are located in these four sections in intra cluster routing. In inter cluster routing, CHs are layered in accordance with their distance from the BS.

A safe data aggregation scheme was described in [30]. The goal of data patterns is to reduce redundant data in the network, reduce sensor node energy consumption, and maintain data accuracy and precision. Intra cluster data aggregation, inter cluster data aggregation, and data transfer are the three phases of a secure data aggregation method. In [31], a binary tree was used to organize sensor nodes. Subsequently, data aggregation was conducted on the tree's middle nodes, and analyzed data were sent to the root node. The data must first be approved to prevent unauthorized data aggregation requests. The aggregation process then begins, and an enhanced cyclic redundancy code is used hop by hop to ensure data aggregation accuracy and reliability. Meanwhile, data packets obtained from their offspring are subjected to cumulative functions imposed by intermediate nodes. As a result, the workload of sensors in the network is significantly reduced. This pattern continues, and aggregated data are eventually sent to the base station.

In [32], a secure combination data aggregation method called SHSDA was proposed; this method is based on a mixture of star and tree structures. The network is divided into four sections, each of which is geographically split into four equal components. A star structure is formed

among nodes. The best node in terms of residual energy and centrality is chosen as the root of a star structure in each section. An energy-efficient layer routing protocol was proposed by Hajipour and Barati [33]. This technique divides the network into circles that are concentric. The circles are then divided into eight equal sectors at a  $45^\circ$  angle. For each section, an agent is chosen. Each section's agent is in charge of gathering and aggregating data sensed by nodes in that section. When the agent receives data, it adds error detection and correction codes before sending the data to the lower layer of the same sector's agent. The process is repeated until the arrival of all data at the base station.

In [34], a two-level energy-aware routing and a clustering method were proposed. The network is clustered, and the most suitable CHs are chosen on the basis of main factors related to energy consumption. A rendezvous region is then built to create a communication substructure between CHs, and nodes in this region are referred to as backbone nodes. In this area, a tree is created to send aggregated data to the sink through CHs. This tree provides an appropriate substructure for sending data to the sink, allowing CHs to send data to the sink with the least amount of overhead and energy and in a short amount of time. Because of the use of this microstructure, CHs do not need to use route discovery to send data to the sink. These findings indicate that range-free localization algorithms still need improvement in terms of localization accuracy (LA). Thus, we propose a new algorithm for 3D WSNs that employs the multiobjective OLCTO approach.

### 3 Conventional 3D DV-hop algorithm

The free localization technique is based on a protocol for distance routing. To determine the number of hops of beacon nodes and the minimum hop distance apart from WSNs, the distance between dumb nodes or unknown nodes and beacon nodes is calculated. Different paths form in a network topology among dumb and beacon nodes that are not linear due to nonuniform connectivity with wireless sensor nodes. The 3D DV-Hop localization algorithm is an extension of the classic DV-Hop technique. Therefore, some errors have been identified in the algorithm for node positioning [25].

**Step 1** The minimum hop amount is defined for unknown and beacon nodes in step 1. By transmitting signals through beacon nodes through the vector protocol system, neighboring nodes can be shown their location. Information exists in the form of  $H_i$ ,  $a_j$ ,  $b_j$ , and  $c_j$  in which  $id$  is the identity;  $a_j$ ,  $b_j$ , and  $c_j$  are coordinates; and  $H_i$  is the hop count for the  $i$  beacon node. First, 0 is

set as the value of  $H_i$  [35]. The nodes obtain data from the broadcast and record the hop amount and localization of the vector's beacon nodes. The  $H_i$  value must be increased by 1 through this process [26]. In this update process, if any node receives the same  $id$  group, the new received data will be compared with the original value of  $H_i$ . The nodes obtain broadcast data and track the hop amount and localization for the beacon nodes of the vector.

**Step 2** The minimum hop count and average hop distance are determined to calculate the distance between unknown and beacon nodes. The average hop distance for the entire network is determined by obtaining the position and hop amount for beacon nodes, as described in the previous stage. Then, these data are transmitted to the entire network. Furthermore, most nodes require information on the average hop distance from a beacon node that is nearer to them [36]. The following equation provides the typical distance of the hop ( $jp_i$ ) and the hop range ( $p_i$ ) between the  $i(a_i, b_i, c_i)$  beacon node and the other (beacon) node  $j(a_j, b_j, c_j)$ :

$$jp_i = \frac{\sum_{i \neq j} \sqrt{(a_i - a_j)^2 + (b_i - b_j)^2 + (c_i - c_j)^2}}{\sum_{i \neq j} h(ij)}, \tag{1}$$

where  $jp_i$  is the minimum hop distance and  $h(ij)$  is the number of hops from beacon nodes  $i$  and  $j$ . The following formula can be used to calculate the distance between beacon and dumb nodes.

$$p_{iU} = jp_{iU} \times Hop_{min}, \tag{2}$$

where  $jp_{iU}$  is the step distance,  $Hop_{min}$  is the step count between the  $i$  beacon nodes, and  $p_{iU}$  denotes the estimated distance between the dumb node  $U$  and the beacon node  $i$ .

**Step 3** Let the coordinates of the  $U$  dumb node be  $(a, b, c)$ , and the  $i^{th}$  beacon node be  $(a_i, b_i, c_i)$  ( $1 \leq i \leq n$ ) in stage 3. Correspondingly, the distance between the  $i^{th}$  beacon node and the unknown or dumb node  $U$  is  $p_i$  ( $1 < i < n$ ). The dumb node's coordinates can be determined as follows:

$$\begin{aligned} (a - a_1)^2 + (b - b_1)^2 + (c - c_1)^2 &= p_1^2; \\ (a - a_2)^2 + (b - b_2)^2 + (c - c_2)^2 &= p_2^2; \\ \dots\dots\dots \\ (a - a_n)^2 + (b - b_n)^2 + (c - c_n)^2 &= p_n^2; \end{aligned} \tag{3}$$

$$a_1^2 - a_n^2 - 2(a_1 - a_n)a + b_1^2 - b_n^2 - 2(b_1 - b_n)b + c_1^2 - c_n^2 - 2(c_1 - c_n)c = p_1^2 - p_n^2; \tag{4}$$

$$\dots\dots\dots a_{n-1}^2 - a_n^2 - 2(a_{n-1} - a_n)a + b_{n-1}^2 - b_n^2 - 2(b_{n-1} - b_n)b + c_{n-1}^2 - c_n^2 - 2(c_{n-1} - c_n)c = p_{n-1}^2 - p_n^2;$$

(4) can be arranged in the matrix form as follows:

$$SA = T, \tag{5}$$

$$\text{where } S = \begin{bmatrix} 2(a_1 - a_n) & 2(b_1 - b_n) & 2(c_1 - c_n) \\ 2(a_2 - a_n) & 2(b_1 - b_n) & 2(c_1 - c_n) \\ \dots\dots\dots \\ 2(a_{n-1} - a_n) & 2(b_1 - b_n) & 2(c_1 - c_n) \end{bmatrix},$$

$$T = \begin{bmatrix} a_1^2 - a_n^2 + b_1^2 - b_n^2 + c_1^2 - c_n^2 + p_n^2 - p_1^2 \\ a_2^2 - a_n^2 + b_2^2 - b_n^2 + c_2^2 - c_n^2 + p_n^2 - p_2^2 \\ \dots\dots\dots \\ a_{n-1}^2 - a_n^2 - b_{n-1}^2 - b_n^2 - c_{n-1}^2 - c_n^2 + p_n^2 - p_{n-1}^2 \end{bmatrix},$$

$$A = \begin{bmatrix} a \\ b \\ c \end{bmatrix},$$

Finally, the least square approximation is used to calculate the dumb node coordinates as given below:

$$A = (S^T S)^{-1} S^T T. \tag{6}$$

## 4 Methodology:

### 4.1 Mathematical formulation of localization in 3D WSNs

Let us consider an update to a 3-dimensional network.  $(a_1, b_1, c_1), (a_2, b_2, c_2), (a_3, b_3, c_3), \dots\dots\dots(a_n, b_n, c_n)$  are the positions of beacon nodes. The positions of dumb nodes are  $(a_{s+1}, b_{s+1}, c_{s+1}), (a_{s+2}, b_{s+2}, c_{s+2}), (a_{s+3}, b_{s+3}, c_{s+3}), \dots\dots\dots(a_{s+t}, b_{s+t}, c_{s+t})$ . The following formula can be used to calculate the distance between beacon nodes:

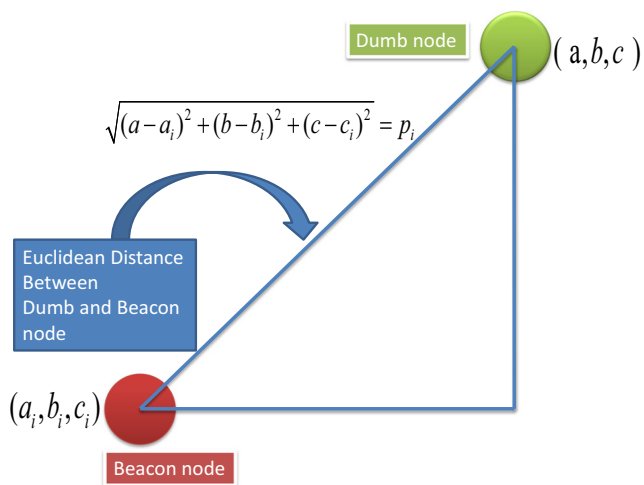
$$(a, b, c) = f_{i=1,2,\dots\dots,s}(a_i, b_i, c_i, p_i). \tag{7}$$

where  $(a, b, c)$  are the coordinate of the dumb node,  $(a_i, b_i, c_i)$  are the coordinates of the  $i^{th}$  beacon node position, and  $p_i$  denotes the distance between dumb nodes and the  $i^{th}$  beacon node. In general, localization systems work in two stages:

**Stage 1:** Internode distances can be estimated using the hop link (hop counting) or internode communication.

In addition, measurements can be used to calculate the actual physical distance.

**Stage 2:** Distances are calculated and converted into the known coordinates of the network's node. To estimate distance measurements, an optimization problem model



**Fig. 1** Euclidean distance between dumb nodes and beacons

that minimizes the total sensor position errors can be designed.

As presented in Fig. 1, the Pythagorean theorem can be utilized to calculate the distance between nodes. The Euclidean distance between the dumb and beacon nodes is given as follows:

$$\sqrt{(a - a_i)^2 + (b - b_i)^2 + (c - c_i)^2} = p_i. \tag{8}$$

$$f(a, b, c) = \min \left( \sum_{i=1,2,\dots,m} \left| \sqrt{(a - a_i)^2 + (b - b_i)^2 + (c - c_i)^2} - p_i \right| \right). \tag{9}$$

where  $f(a, b, c)$  is the objective function that represents the distance error. The optimization problem defined by (9) can be solved using diverse optimization approaches. The chosen method should effectively minimize the LE and optimize derived locations to improve LA. To overcome the challenge of localization, many scholars have proposed employing popular heuristic methods.

### 4.2 OLCTO algorithm

Meta-heuristic optimization algorithms are frequently utilized in both experimental and industrial settings because of their versatility, simplicity, and robustness. CTO is among the most recent and widely used algorithms in this field [37]. A chaotic dynamic is introduced in CTO with an adjustable inertia weight factor to enrich the search behavior and prevent local optimum problems [38]. The following equation can be used to describe a chaotic search strategy:

$$C S_i^{t+1} = 4C S_i^t(1 - C S_i^t), i = 1, 2, \dots, n, \tag{10}$$

The CTO approach includes the use of nonlinear dynamic acceleration coefficients to change the cognitive component  $C_1$  and the social component  $C_2$  as follows [39]:

$$C_1 = -(C_{1f} - C_{1i}) \times \left( \frac{iter}{iter_{max}} \right)^2 + C_{1f}, \tag{11}$$

$$C_2 = C_{1i} \times \left( 1 - \frac{iter}{iter_{max}} \right)^2 + C_{1f} \times \frac{iter}{iter_{max}}. \tag{12}$$

The CTO algorithm has several flaws, including the ability to trap in local minima and other limitations when dealing with multiobjective problems. Thus, OLCTO is currently being used to overcome the limitations of CTO. The orthogonal diagonalization (OD) technique is used by the OLCTO algorithm. This technique can be used to derive orthogonal guidance vectors in the active group. The diagonal matrix can be obtained using the OD process by converting the multiplication of three matrices into the diagonal matrix ( $DM$ ).  $DM$  is used to update the section topper, the class topper, and learn from vectors for all students in the class. To obtain  $DM$ , a square matrix  $P$  of size  $r \times r$  is converted into a  $DM$  of size  $r \times x \times r$  as follows:

$$P = RDMR^{-1}, \tag{13}$$

where  $R$  is a  $r \times x \times r$ -dimensional matrix composed of  $P$ 's eigenvector and  $DM$ 's diagonal elements. Because  $R$  is invertible, Eq. (13) can be defined as follows:

$$P = BDMB^{-1}, \tag{14}$$

where  $B$  is an orthogonal matrix; thus, (14) can be written in the following form:

$$DM = B^T P B, \tag{15}$$

Figure 2 shows the OLCTO flowchart.

The OLCTO procedure steps are as follows:

- Step 1. Initialize the vectors of position (students)  $S_i(0)$  at random and learn the form  $L_i(0)$  for each student  $i$ .  $i = 1, 2, \dots, m$ .
- Step 2. Calculate the performance index  $p(x)$  by using the position vector (students)  $S_i(0)$ .
- Step 3. Use the equation to initialize the section topper position vector of student  $i$  in the CTO algorithm:

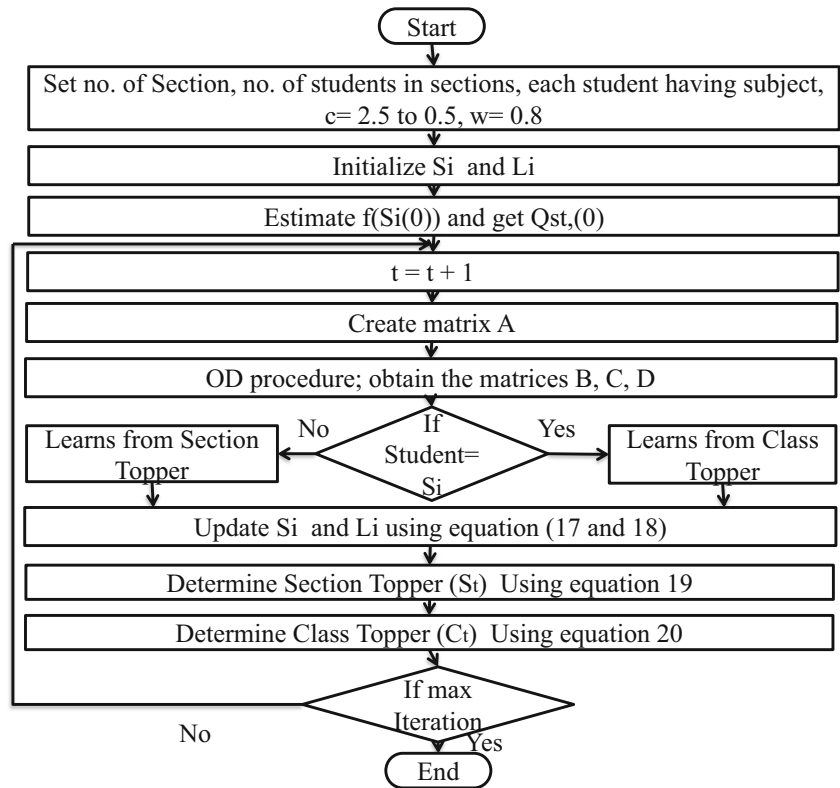
$$Q_{S_i,i} = S_i(0), \tag{16}$$

where  $Q_{S_i,i}$  denotes the section topper. To determine the section topper position vectors, use (16).

- Step 4. The  $m$  section topper position vectors can be ordered in ascending order by using the performance index value of  $p(x)$ .
- Step 5. Create a matrix  $B$  with the size  $m \times x \times p$ . Each entry in this matrix corresponds to one of the  $m$  section topper position vectors in same order as in Step 4.



Fig. 2 Flowchart of OLCTO



Step 6. To transform matrix  $B$  into a symmetric matrix  $A$  with a dimension of  $m \times x \times p$ , use the CTO pseudo code [22].

Step 7. Apply OD to matrix  $A$  to obtain a  $DM$  of size  $r \times x \times r$ .

Step 8. The equations below can be used to update the vectors:

$$L_i^t = L_i^{t-1} + C \times \phi \times [DM_i^t - S_i^{t-1}], \tag{17}$$

$$S_i^t = S_i^{t-1} + L_i^t, \tag{18}$$

$DM_i$  is the diagonal of the matrix  $DM$ , and  $i = 1, 2, \dots, r$ ,  $C$  is the coefficient of nonlinear dynamic acceleration, ranging from [2.5 to 0.5].

Step 9. Calculate  $Q_{S_r,i}^t$  (section topper) from the  $m$  students by using the following Eq. (19):

$$Q_{S_r,i}^t = \begin{cases} S_i^t, & \text{if } h(S_i^t) \leq h(Q_i^{t-1}), \\ Q_{S_r,i}^{t-1} & \text{otherwise,} \end{cases} \tag{19}$$

Then, calculate  $f(Q_{S_r,i}^t)$ ,  $i = 1, 2, \dots, m$ .

Step 10. The following method can be used to determine the class topper (best position). Choose  $Q_{S_r}$ , which corresponds to the minimum of  $f(Q_{S_r}^t)$ ,  $i = 1, 2, \dots, m$ .

Assess  $p(x)$  to find  $Q_{C_t}^t$ , which is the class topper (best position).

$$Q_{C_t}^t = \min Q_{S_r,i}^t, \tag{20}$$

Step 11. Finish the iterations,  $t = N_{iter}$

The following steps are included in the proposed algorithm:

**Step 1.** Dumb node locations are determined using the minimum hop distance calculated by each beacon (i.e., one-hop-size) from another beacon in the network. The greater the precision of this estimated distance is, the more effective the approximate positions are.

**Step 2.** Instead of using the conventional approach, we define and measure the average hop distance among beacons. Thus, we use the polynomial approximation to reduce the estimated LE and increase LA. The following polynomial is used to calculate the approximate hop distance from the  $i^{th}$  beacons and other beacons  $m$ .

$$r_{im} = \kappa_0 + \kappa_1 p_{im} + \kappa_2 p_{im}^2, \tag{21}$$

where  $\kappa_0$ ,  $\kappa_1$  and  $\kappa_2$  are the coefficients.

$$\begin{bmatrix} p_{i1}^2 & p_{i1} & 1 \\ p_{i2}^2 & p_{i2} & 1 \\ \vdots & \vdots & \vdots \\ p_{ij}^2 & p_{ij} & 1 \end{bmatrix} \begin{bmatrix} \kappa_2 \\ \kappa_1 \\ \kappa_0 \end{bmatrix} = \begin{bmatrix} r_{i1} \\ r_{i2} \\ \vdots \\ r_{ij} \end{bmatrix}, \tag{22}$$

We solve the following equation to determine the polynomial function estimate of the distance among nodes:

$$\kappa = (P^T P)^{-1} P^T R, \tag{23}$$

For each dumb node, the distance between itself and the beacon node is calculated as follows:

$$r_{ij} = \kappa_0 + \kappa_1 p_{ij} + \kappa_2 p_{ij}^2, \tag{24}$$

where  $r_{ij}$  and  $p_{ij}$  represent the distance as well as minimum number of hops between the  $i^{th}$  dumb node and the  $j^{th}$  beacon node, respectively. Thus, we obtain the following matrix form:

$$r_{est} = hop \times \kappa, \tag{25}$$

The actual distance between the  $i^{th}$  and  $j^{th}$  is calculated as

$$r_{true} = \sqrt{(a_i - a_j)^2 + (b_i - b_j)^2 + (c_i - c_j)^2}. \tag{26}$$

The error between the  $i^{th}$  and  $j^{th}$  nodes of the beacon is determined as follows:

$$r_{error} = r_{est} - r_{true}. \tag{27}$$

We now add a rectification factor, which is defined as follows:

$$\tau = \frac{r_{error}}{s}, \tag{28}$$

where  $s$  is the number of beacon nodes. The  $\tau$  rectification factor is used by adding it to the previous hop size to change the hop size of the beacon node. The adjusted distance between the  $i^{th}$  beacon nodes and the  $k^{th}$  dumb node is determined as [40]:

$$r_{ik}^{Mod} = (HopSize + \tau) \times H_{ik}. \tag{29}$$

**Step 3.** To enhance the location accuracy between sensor nodes, we present an orthogonal learning DV-Hop localization approach based on CTO. In orthogonal learning optimization, the optimal solution for result comparison is indicated as the minimum of the sum of two goal values. We used four single-objective functions and two multiobjective (orthogonal learning) functions

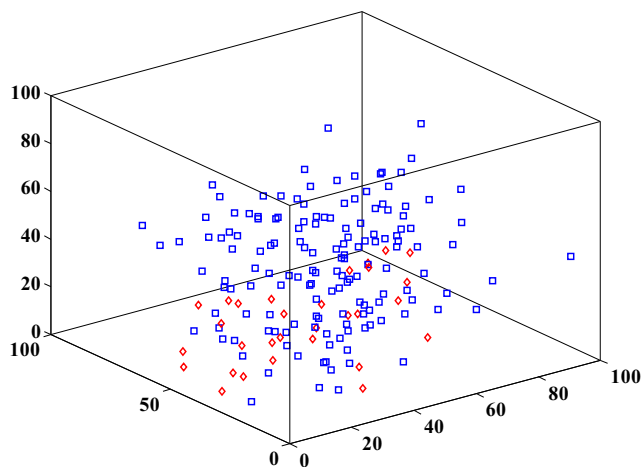


Fig. 3 Deployment of 3D sensor nodes

Table 1 Parameters for simulation

Parameters	Value
Border length	100 × 100 × 100 m <sup>3</sup> to 300 × 300 × 300 m <sup>3</sup>
Number of sensor nodes	200 to 450
Beacon nodes (M)	20 to 160
Range (R)	25 to 50
Algorithm of optimization	OLCTO
Maximum iterations	50
Simulation tool	LabVIEW©2015

to develop our proposed technique. The following is a proposed algorithm that employs single-objective optimization functions:

$$f_1(a, b, c) = \min \left( \sum_{i=1,2,\dots,M} \left| \sqrt{(a - a_i)^2 + (b - b_i)^2 + (c - c_i)^2} - r_{ik}^{Mod} \right| \right). \tag{30}$$

$$f_2(a, b, c) = \min \left( \sum_{i=1,2,\dots,M} \left| \sqrt{(a - a_i)^2 + (b - b_i)^2 + (c - c_i)^2} - r_{ik}^{Mod} \right|^2 \right). \tag{31}$$

$$f_3(a, b, c) = \min \left( \sum_{i=1,2,\dots,M} \left| \sqrt{(a - a_i)^2 + (b - b_i)^2 + (c - c_i)^2} - r_{it}^{Mod} \right| \right). \tag{32}$$

where  $r_{it}^{Mod} = hop \times \kappa$ ,  $\kappa = (P^T P)^{-1} P^T R$  and  $r^{error} = r_{est} - r_{true}$ .

$$f_4(a, b, c) = \min \left( \sum_{i=1,2,\dots,M} \left| \sqrt{(a - a_i)^2 + (b - b_i)^2 + (c - c_i)^2} - r_{it}^{Mod} \right|^2 \right). \tag{33}$$

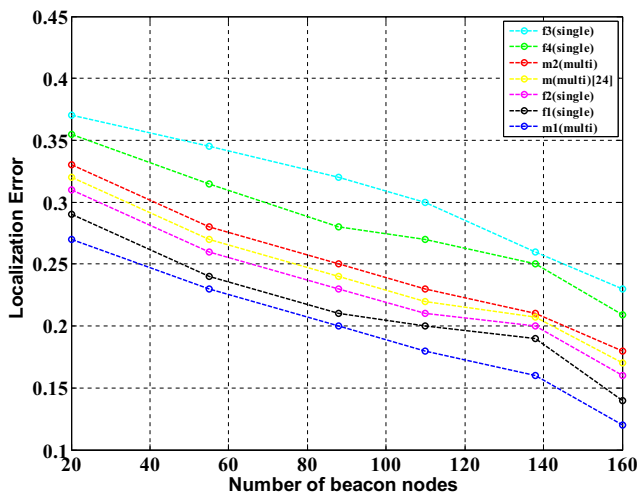


Fig. 4 Changes in LE with respect to changes in the number of beacon nodes

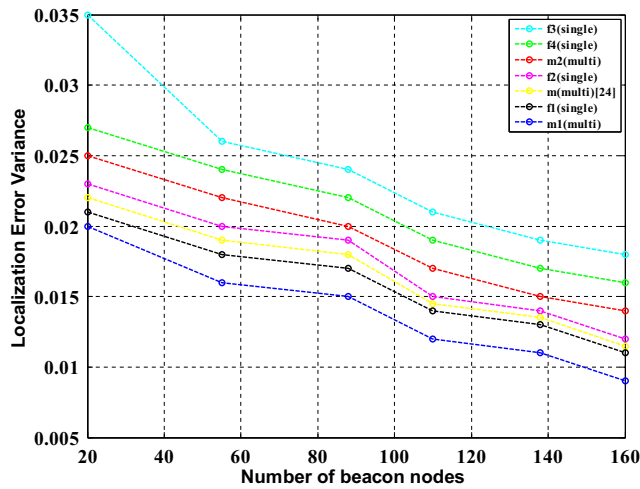


Fig. 5 Changes in LEV with respect to changes in the number of beacon nodes

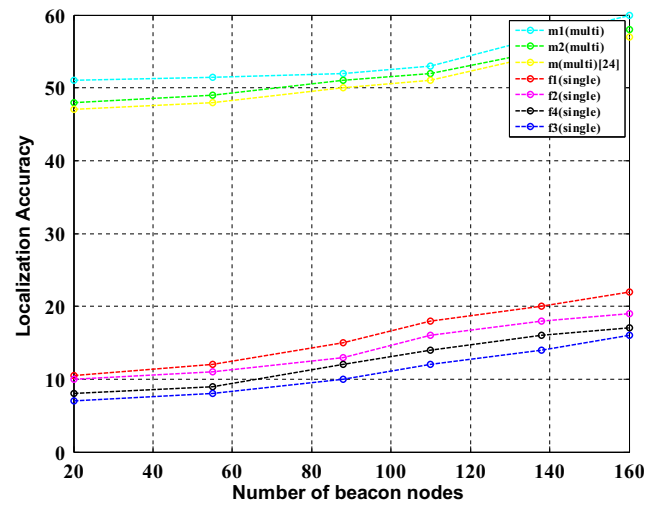


Fig. 6 Changes in LA with respect to changes in the number of beacon nodes

The proposed algorithm considers the following multiobjective (orthogonal learning) functions, where  $m_1$  is formulated by combining  $f_1$  and  $f_2$ , and  $m_1$  and  $m_1$  is represented as follows:

$$m_1(a, b, c) = \min \left( \sum_{i=1,2,\dots,M} \left| \sqrt{(a - a_i)^2 + (b - b_i)^2 + (c - c_i)^2} - r_{ik}^{Mod} \right| \right);$$

and

$$\min \left( \sum_{i=1,2,\dots,M} \left| \sqrt{(a - a_i)^2 + (b - b_i)^2 + (c - c_i)^2} - r_{ik}^{Mod} \right|^2 \right);$$

(34)

$m_2$  is the formulated by combining  $f_3$  and  $f_4$ , and  $m_2$  is represented as follows:

$$m_2(a, b, c) = \min \left( \sum_{i=1,2,\dots,M} \left| \sqrt{(a - a_i)^2 + (b - b_i)^2 + (c - c_i)^2} - r_{it}^{Mod} \right| \right);$$

and

$$\min \left( \sum_{i=1,2,\dots,M} \left| \sqrt{(a - a_i)^2 + (b - b_i)^2 + (c - c_i)^2} - r_{it}^{Mod} \right|^2 \right);$$

(35)

### 5 Metrics of performance

LE, localization error variation (LEV), LA, and coverage are parameters used to evaluate the performance of the proposed OLCTO-based DV-Hop algorithm.

Table 2 Comparison of the effects of the proposed (multiobjective) algorithm on LE and LEV beacon nodes with single-objective algorithms and multiobjective PSO [24]

Number of beacon nodes		20	50	80	100	130	150
$f_5$ (single objective)	LE	.37	.5	.6	.66	.62	.91
	LEV	.75	.625	.6	.75	.58	1
$f_4$ (single objective)	LE	.29	.47	.58	.6	.6	.74
	LEV	.35	.225	.2	.35	.28	.77
$m_2$ (multiobjective)	LE	.22	.27	.32	.48	.4	.5
	LEV	.25	.2	.18	.24	.2	.55
$m$ multiobjective [24]	LE	.156	.19	.25	.28	.27	.35
	LEV	.20	.16	.15	.17	.16	.31
$f_2$ (single objective)	LE	.148	.19	.24	.27	.26	.3
	LEV	.15	.1	.12	.14	.1	.33
$f_1$ (single objective)	LE	.074	.09	.12	.14	.13	.16
	LEV	.05	.04	.042	.046	.23	.22
$m_1$ (multiobjective)	LE	.068	.041	.1	.12	.11	.14
	LEV	.047	.035	.04	.043	.13	.2



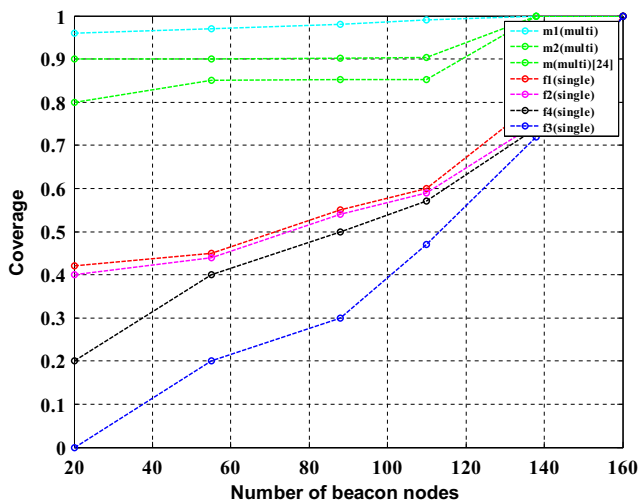


Fig. 7 Changes in coverage with respect to changes in the number of beacon nodes

5.1 LE

The LE for a 3D WSN can be computed as follows:

$$LE = \frac{\sum_{i=1}^t \sqrt{(a_i^{est} - a_i^{tr})^2 + (b_i^{est} - b_i^{tr})^2 + (c_i^{est} - c_i^{tr})^2}}{t \times R} \tag{36}$$

5.2 LEV

Variance measures how far each number in a set deviates from the mean. LEV for a 3D WSN can be computed as follows:

$$LEV = \sqrt{\frac{\sum_{i=1}^t \left( \sqrt{(a_i^{est} - a_i^{tr})^2 + (b_i^{est} - b_i^{tr})^2 + (c_i^{est} - c_i^{tr})^2} - LE \times R \right)^2}{t \times R^2}} \tag{37}$$

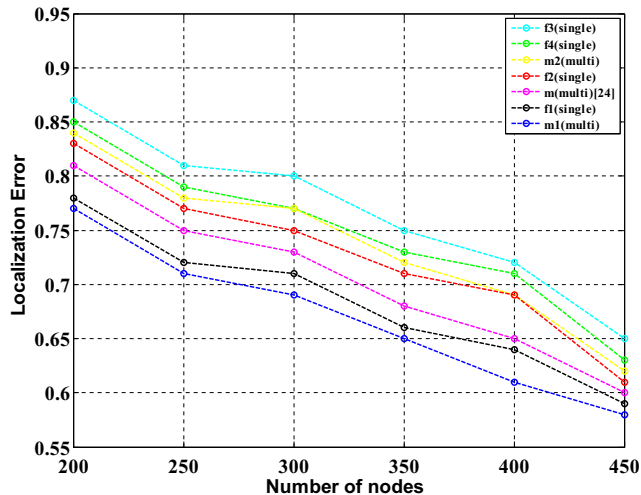


Fig. 8 Changes in LE with respect to changes in the number of nodes

5.3 LA and coverage

LA can be computed as follows:

$$LA = \frac{\sum_{i=1}^t \sqrt{(a_i^{est} - a_i^{tr})^2 + (b_i^{est} - b_i^{tr})^2 + (c_i^{est} - c_i^{tr})^2}}{t \times R^2} \tag{38}$$

Coverage can be computed as follows:

$$Coverage = \frac{\bigcup_{i=1}^t R^2(a_i, b_i, c_i)}{Area} \tag{39}$$

where  $t$  denotes the number of unknown or dumb nodes and  $(a_i^{tr}, b_i^{tr}, c_i^{tr})$  is true positions of unknown or dumb node  $i$ . The estimated dumb node positions are  $i^{th}$  is

Table 3 Comparison of the effects of the proposed (multiobjective) algorithm on LA and coverage beacon nodes with single-objective algorithms and multiobjective PSO [24] were conducted

Number of beacon nodes		20	50	80	100	130	150
m1 (multiobjective)	LA	0.062	0.071	0.019	0.018	0.052	0.033
	Coverage	0.066	0.075	0.025	0.023	0	0
m2 (multiobjective)	LA	0.058	0.061	0.011	0.011	0.042	0.023
	Coverage	0.055	0.065	0.015	0.013	0	0
m multiobjective [24]	LA	0.069	0.078	0.021	0.020	0.062	0.043
	Coverage	0.058	0.075	0.025	0.023	0	0
f1 (single objective)	LA	0.079	0.081	0.031	0.030	0.072	0.045
	Coverage	0.47	0.65	0.29	0.27	0.31	0
f2 (single objective)	LA	0.81	0.38	0.35	0.26	0.21	0.187
	Coverage	0.58	0.65	0.28	0.26	0.1	0
f4 (single objective)	LA	0.84	0.42	0.41	0.36	0.27	0.45
	Coverage	0.79	0.75	0.48	0.06	0.1	0
f3 (single objective)	LA	0.86	0.54	0.58	0.45	0.26	0.58
	Coverage	1	0.79	0.48	0.19	0.1	0

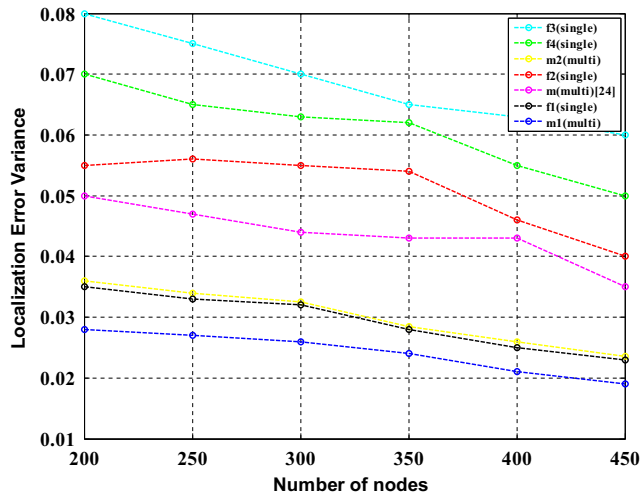


Fig. 9 Changes in LEV with respect to changes in the number of nodes

$(a_i^{est}, b_i^{est}, c_i^{est})$ .  $R$  is the sensor node’s communication range.

### 6 Simulation and results analysis

In a 3D environment, the proposed approach is implemented. For our system, we used the OLCTO algorithm with  $C = 4$ ,  $W = 2.5$  to  $0.05$ ,  $Itermax = 20$ , as well as random variables between 0 and 1. For a 3D, the minimum transmission range of a node is  $25 m^2$ . The value of the communication range is dependent on node density. The greater the communication range is, the lower is the node density. The communication range of 3D space is larger than that of 2D space. Figure 3 depicts the typical distribution of 150

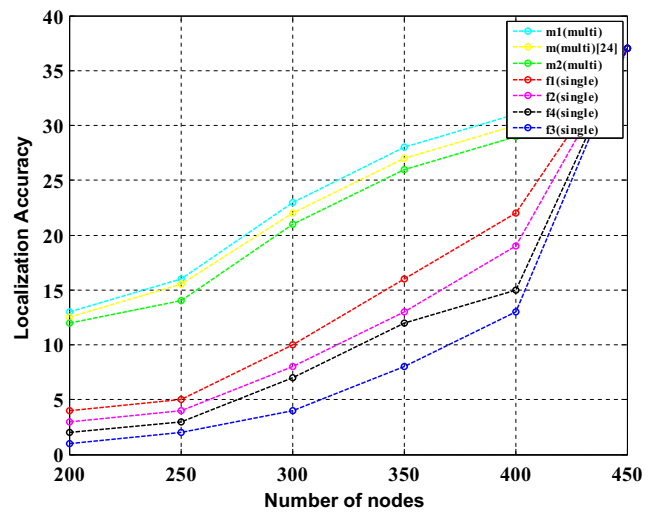


Fig. 10 Changes in LA with respect to changes in the number of nodes

dumb nodes and 30 beacon nodes in 3D space. Table 1 lists parameters used in the simulation. For the scenario, we used uniformly dispersed dumb nodes and beacon nodes, and the results were assessed using the following parameters: (i) the total number of nodes, (ii) the total number of beacon nodes, and (iii) communication range of nodes.

#### 6.1 Effects of beacon nodes on LE, LEV, LA, and coverage

Figures 4 and 5 present the effects of the beacon on LE and LEV, respectively. The number of beacon nodes varies from 20 to 160. As presented in Table 2, the  $m1$  had less LEs and LEV. For example, when the total number of beacon nodes is set to 20, the multiobjective function  $m1$  had

Table 4 Comparison of the effects of the proposed (multiobjective) algorithm on the LE and LEV of the number of nodes with single-objective algorithms and multiobjective PSO [24]

Number of nodes		200	250	300	350	400	450
$f5$ (single objective)	LE	0.144	0.14	0.028	0.036	0.037	0.04
	LEV	1.8	0.035	0.037	0.076	0.16	0.173
$f6$ (single objective)	LE	0.118	0.13	0.028	0.035	0.036	0.038
	LEV	1.5	0.037	0.038	0.083	0.19	0.21
$m2$ (multiobjective)	LE	0.105	0.056	0.058	0.065	0.065	0.072
	LEV	0.96	0.43	0.44	0.55	0.61	0.54
$f2$ (single objective)	LE	0.092	0.084	0.085	0.095	0.095	0.099
	LEV	0.78	0.77	0.78	0.88	0.95	0.59
$m$ multiobjective [24]	LE	0.065	0.08	0.075	0.085	0.085	0.089
	LEV	0.35	0.78	0.79	0.89	0.98	0.61
$f1$ (single objective)	LE	0.026	0.112	0.115	0.166	0.169	0.171
	LEV	0.32	0.037	0.038	0.083	1.22	0.58
$m1$ (multiobjective)	LE	0.025	0.140	0.159	0.20	0.19	0.21
	LEV	0.24	0.035	0.037	0.076	0.16	0.017

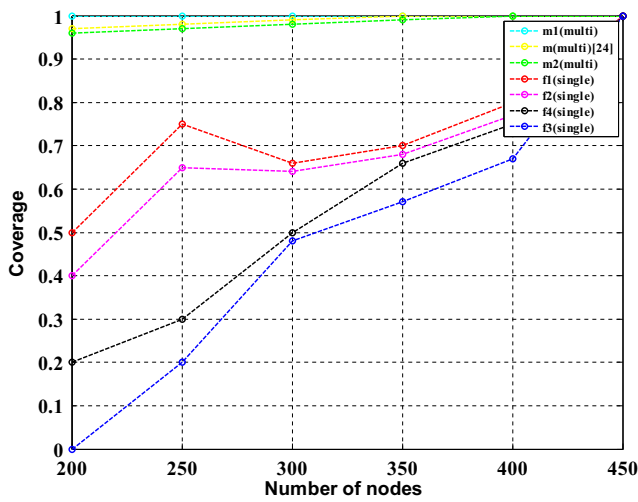


Fig. 11 Changes in coverage with respect to changes in the number of nodes

approximately 0.6%, 8%, 8.8%, 15.2%, 22.2%, and 30.2% less LEs in  $f1$ ,  $f2$ ,  $m$  [24],  $m2$ ,  $f4$ , and  $f5$ , respectively, and 0.3%, 10.3%, 15.3%, 20.3%, 30.3%, and 70.3% less LEV in  $f1$ ,  $f2$ ,  $m$  [24],  $m2$ ,  $f6$ , and  $f5$ , respectively. The proposed method has approximately 0.4%, 0.8%, and 0.9% less LEs in [30, 41] and [28], respectively, and 0.5%, 1.3%, and 1.9% less LEV in [30, 41] and [28], respectively.

Figures 6 and 7 present the effects of the beacon on LA and coverage, respectively, and Table 3 presents the analysis data. As indicated in Table 3, the multiobjective function  $m1$  exhibited improved LA and coverage. For example, when the total number of beacon nodes was set to 20, the multiobjective function  $m1$  had approximately 0.4%, 2%, 73.2%, 74.2%, 78.3%, and 80.02% improved LA in

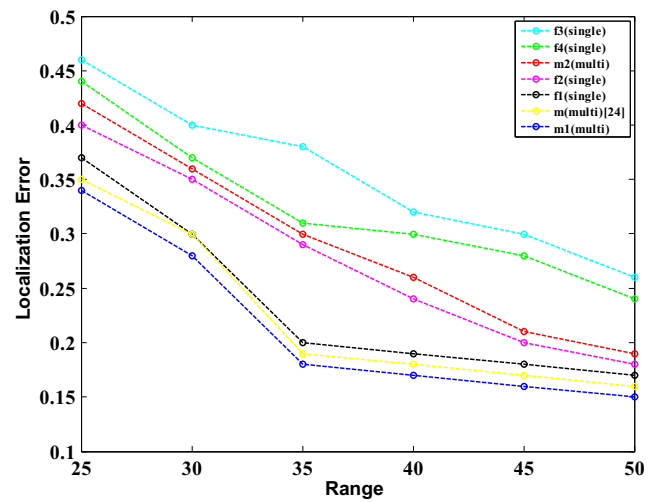


Fig. 12 Changes in LE with respect to range

$m2$ ,  $m$  [24],  $f1$ ,  $f2$ ,  $f4$ , and  $f3$  and 0.7%, 9.8%, 47.8%, 49.8%, 69.8%, and 89.8% improved coverage in  $m2$ ,  $m$  [24],  $f1$ ,  $f2$ ,  $f4$ , and  $f3$ , respectively. The proposed method had approximately 0.4%, 2%, and 8.02% improved LA in [30, 41] and [28], respectively, and 0.7%, 6.8%, and 9.8% improved coverage in [30, 41] and [28], respectively.

### 6.2 Effect the total number of nodes on localization errors, error variance, accuracy, and coverage

Figures 8 and 9 present the effect of number nodes on LE and LEV, respectively. As presented in Table 4, with an increase in the total number of nodes, the multiobjective function  $m1$  had the lowest LE and LEV. For example, when the total number of nodes was set to 200, the multiobjective

Table 5 Comparison of the effects of the proposed (multiobjective) algorithm on LA and coverage for the number of nodes with single-objective algorithms and multiobjective PSO [24]

Number of nodes		200	250	300	350	400	450
$m1$ (multiobjective)	LA	0.038	0.066	0.043	0.035	0.032	0
	Coverage	0.03	0.02	0.05	0.01	0	0
$m$ multiobjective [24]	LA	0.04	0.14	0.13	0.12	0.11	0
	Coverage	0.031	0.02	0.05	0.014	0	0
$m2$ (multiobjective)	LA	0.07	0.056	0.033	0.035	0.022	0
	Coverage	0.04	0.04	0.045	0.014	0	0
$f1$ (single objective)	LA	0.69	0.66	0.43	0.45	0.32	0
	Coverage	0.5	0.5	0.55	0.15	0.14	0
$f2$ (single objective)	LA	0.76	0.76	0.53	0.55	0.42	0
	Coverage	0.6	0.9	0.3	0.9	0.6	0
$f4$ (single objective)	LA	0.84	0.78	0.55	0.57	0.45	0
	Coverage	0.8	0.2	0.8	0.3	0.21	0
$f3$ (single objective)	LA	0.92	0.88	0.65	0.67	0.55	0
	Coverage	1	0.2	0.8	0.4	0.3	0

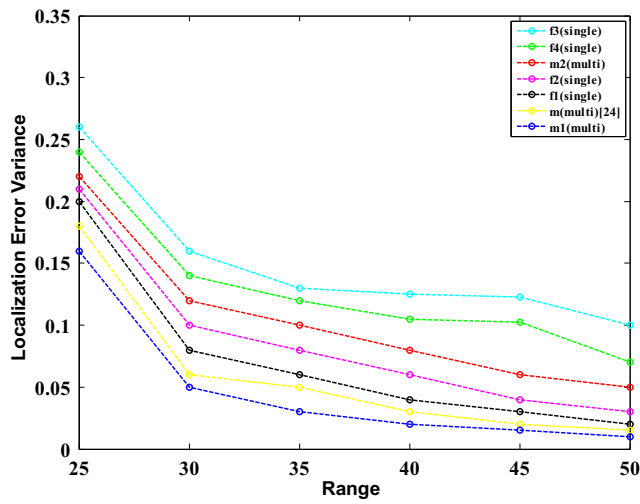


Fig. 13 Changes in LEV with respect to range

function *m1* had approximately 1%, 4%, 6.7%, 8%, 9.3%, and 11.9% less LEs in *f1*, *m* [24], *f2*, *m2*, *f6*, and *f5*, respectively, and 8%, 11%, 54%, 72%, 84%, and 84.6% less LEV in *f1*, *m* [24], *f2*, *m2*, *f6*, and *f5*, respectively. The proposed method had approximately 1.2%, 5%, and 6.9% less LEs in [30, 41] and [28], respectively, and 4%, 7%, and 8.6% less LEV in [30, 41] and [28], respectively.

Figures 10 and 11 present the effect of the total number of nodes on LA and coverage, respectively, and Table 5 presents the associated analysis of data. With an increase in the total number of nodes, the multiobjective function *m1* exhibited improved LA and coverage. For example, when the total number of nodes was 200, the multiobjective function *m1* exhibited approximately 0.2%, 3.2%, 65.2%, 72.2%, 80.2%, and 88.2% improved LA in *m* [24], *m2*, *f1*,

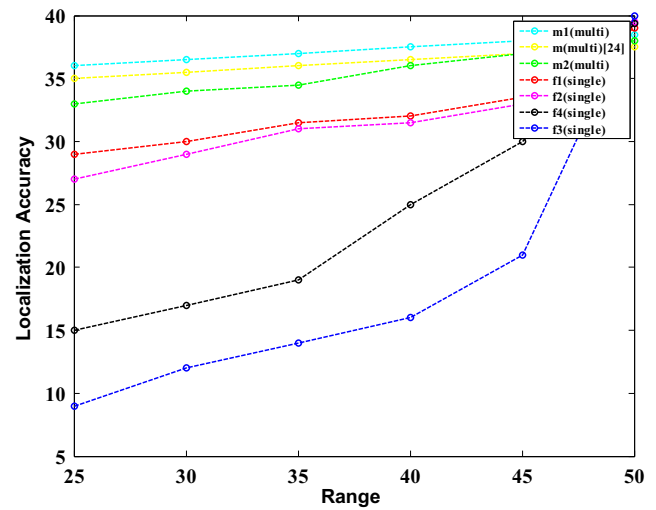


Fig. 14 Changes in LA with respect to range

*f2*, *f4*, and *f3*, respectively, and 0.1%, 1%, 47%, 57%, 77%, and 97% improved coverage in *m* [24], *m2*, *f1*, *f2*, *f4*, and *f3*, respectively. The proposed method resulted in approximately 0.5%, 2.2%, and 8.2% improved LA in [30, 41] and [28], respectively, and 0.2%, 3%, and 7% improved coverage in [30, 41] and [28], respectively.

### 6.3 Effect of LE, LEV, LA, and coverage on the communication range

Figures 12 and 13 present the effect of the transmission range on LE and LEV, respectively, and Table 6 presents the associated analysis of data. For example, when the transmission range was 25, the multiobjective function *m1* had approximately 2.8%, 5%, 15%, 19%, 22%, and 26%

**Table 6** Comparison of the effects of the proposed (multiobjective) algorithm on LE and LEV for the transmission range with single-objective algorithm and multiobjective PSO [24]

Transmission range (m)		25	30	35	40	45	50
<i>f3</i> (single objective)	LE	0.35	0.51	1.2	1.10	1	0.94
	LEV	0.68	2.6	3	4.3	4.6	4.3
<i>f4</i> (single objective)	LE	0.29	0.37	0.8	1	0.89	0.84
	LEV	0.5	2.2	2.75	3.1	3.5	3.2
<i>m2</i> (multiobjective)	LE	0.23	0.31	0.75	0.84	0.52	0.51
	LEV	0.43	1.8	2.1	3.7	2.1	2.5
<i>f2</i> (single objective)	LE	0.17	0.27	0.7	0.78	0.31	0.30
	LEV	0.31	0.84	1.95	2.9	1.6	1.3
<i>f1</i> (single objective)	LE	0.058	0.13	0.2	0.31	0.157	0.21
	LEV	0.25	0.65	1	1.7	0.97	0.92
<i>m</i> multiobjective [24]	LE	0.029	0.034	0.15	0.052	0.94	0.052
	LEV	0.12	0.43	0.73	0.92	0.35	0.34
<i>m1</i> (multiobjective)	LE	0.028	0.033	0.13	1	0.95	0.05
	LEV	0.11	0.28	0.33	0.25	0.142	0.142

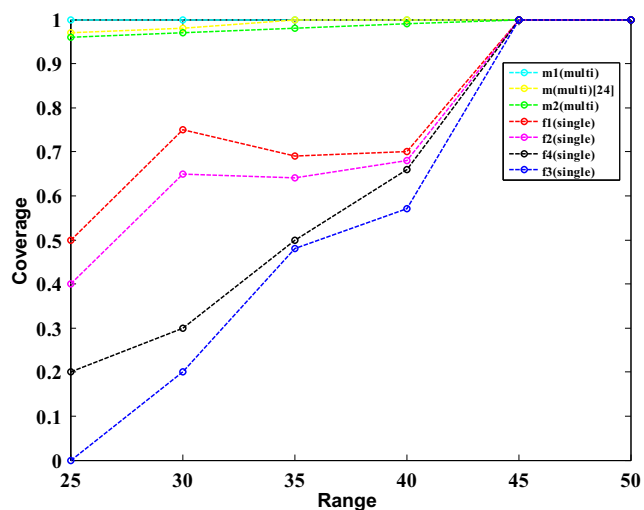


Fig. 15 Changes in coverage with respect to range

less LEs in  $m$  [24],  $f1$ ,  $f2$   $m2$ ,  $f6$ , and  $f5$  and 8%, 56%, 64%, 74%, 78%, and 83% less LEV in  $m$  [24],  $f1$ ,  $f2$   $m2$ ,  $f6$ , and  $f5$ , respectively. The proposed method had approximately 1.8%, 3%, and 6% less LEs in [30, 41] and [28], respectively, and 6%, 6.6%, and 8.3% less LEV in [30, 41] and [28], respectively.

Figures 14 and 15 present the effect of the total number of nodes on LA and coverage, respectively, and Table 7 presents the associated analysis of data. As indicated in Table 7 with an increase in the transmission range, the multiobjective function  $m1$  exhibited improved LA and coverage. For example, when the total number of nodes was 25, the multiobjective function  $m1$  had approximately 7%, 15%, 36%, 50%, 78%, and 86% improved LA in  $m$  [24],

$m2$ ,  $f1$ ,  $f2$ ,  $f4$ , and  $f3$  and 9%, 14%, 94%, 95%, 96%, and 97% improved coverage in  $m$  [24],  $m2$ ,  $f1$ ,  $f2$ ,  $f4$ , and  $f3$ , respectively. The proposed method exhibited approximately 7%, 7.8%, and 8.6% improved LA in [30, 41] and [28] and 9%, 9.5%, and 9.8% improved coverage in [30, 41] and [28].

### 7 Conclusion

In this study, we present a range-free DV-Hop localization method in 3D space by utilizing the OLCTO optimization. The suggested approach converts single-objective functions based on the DV-Hop localization algorithm into two multiobjective functions to reduce LEs. The goal of multiobjective localization is to reduce LEs while increasing positional precision. Among all the proposed objective functions, the multiobjective function  $m1$  had the least LEs and LEV as well as improved LA. Simulation results revealed that the multiobjective function  $m1$  had the highest placement accuracy among all single- and multiobjective functions and multiobjective PSO [24]. The proposed method’s main limitations are in terms of energy consumption, network lifetime, and network reliability. In our next study, we will focus on improving the convergence rate and reducing energy consumption to enhance LA. The following study focuses on reducing communication between dumb and beacon nodes by computing the hop size of all beacons at dumb nodes. This approach can reduce computational time while reducing node energy consumption. Future studies can develop circular and spiral beacon nodes for the deployment of the multi-hop-based approach in a 3D system and on a physical staging ground.

Table 7 Comparison of the effects of the proposed (multiobjective) algorithm on LA and coverage for the transmission range with single-objective algorithm and multiobjective PSO [24]

Transmission range (m)		25	30	35	40	45	50
$m1$ (multiobjective)	LA	3	2.03	1.64	1.8	1.37	0.069
	Coverage	0.03	0.025	0.02	0.015	0	0
$m$ multiobjective [24]	LA	2.8	2.71	2.73	1.64	0.75	0.04
	Coverage	0.3	0.025	0.02	0.015	0	0
$m2$ (multiobjective)	LA	2.6	2.01	2.53	1.24	0.70	0.05
	Coverage	0.035	0.03	0.025	0.022	0	0
$f1$ (single objective)	LA	2.2	0.9	0.85	0.74	0.09	0.025
	Coverage	0.5	0.25	0.3	0.26	0	0
$f2$ (single objective)	LA	2	0.8	0.76	0.72	0.07	0.012
	Coverage	0.6	0.38	0.37	0.35	0	0
$f4$ (single objective)	LA	0.66	0.52	0.46	0.44	0.08	0.025
	Coverage	0.8	0.7	0.6	0.4	0	0
$f3$ (single objective)	LA	0.4	0.3	0.25	0.4	0.07	0.024
	Coverage	1	0.85	0.6	0.45	0	0

**Data availability** Not applicable

**Code availability** Not applicable

## Declarations

**Competing interests** The authors declare no competing interests.

## References

- Dai H, Chen A, Gu X, He L (2011) Localisation algorithm for large-scale and low-density wireless sensor networks. *Electr Lett* 47(15):881–883
- Najeh T, Sassi H, Liouane N (2018) A novel range free localization algorithm in wireless sensor networks based on connectivity and genetic algorithms. *Int J Wireless Inf Netw* 25(1):88–97
- Joshi YK, Younis M (2016) Restoring connectivity in a resource constrained WSN. *J Netw Comput Appl* 66:151–165
- Lee S-M, Cha H, Ha R (2007) Energy-aware location error handling for object tracking applications in wireless sensor networks. *Comput Commun* 30(7):1443–1450
- Gavish B, Neuman I (1992) Routing in a network with unreliable components. *IEEE Trans Commun* 40(7):1248–1258
- Liao W-H, Sheu J-P, Tseng Y-C (2001) Grid: a fully location-aware routing protocol for mobile ad hoc networks. *Telecommun Syst* 18(1):37–60
- Song G, Tam D (2015) Two novel DV-Hop localization algorithms for randomly deployed wireless sensor networks. *Int J Distr Sensor Netw* 11(7):187670
- Gao GQ, Lei L (2010) An improved node localization algorithm based on DV-Hop in WSN. In: 2010 2nd International conference on advanced computer control. *IEEE*, vol 4, pp 321–324
- Bulusu N, Heidemann J, Estrin D (2000) GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Commun* 7(5):28–34
- Niculescu D, Nath B (2001) Ad hoc positioning system (APS). In: *GLOBECOM'01. IEEE global telecommunications conference* (Cat. No. 01CH37270). *IEEE*, vol 5, pp 2926–2931
- Niculescu D (2003) DV based positioning in ad hoc networks. *Telecommun Syst* 22(1-4):267–280
- Nagpal R et al (1999) Organizing a global coordinate system from local information on an amorphous computer
- He T, Huang C, Blum BM, Stankovic JA, Abdelzaher T (2003) Range-free localization schemes for large scale sensor networks. In: *Proceedings of the 9th annual international conference on Mobile computing and networking*, pp 81–95
- Shang Y, Ruml W, Zhang Y, Fromherz MP (2003) Localization from mere connectivity
- Kaur A, Kumar P, Gupta GP (2019) A weighted centroid localization algorithm for randomly deployed wireless sensor networks. *J King Saud Univ-Comput Inf Sci* 31(1):82–91
- Viani F, Lizzi L, Rocca P, Benedetti M, Donelli M, Massa A (2008) Object tracking through RSSI measurements in wireless sensor networks. *Electron Lett* 44(10):653–654
- Girod L, Estrin D (2001) Robust range estimation using acoustic and multimodal sensing. In: *Proceedings 2001 IEEE/RSJ international conference on intelligent robots and systems. Expanding the societal role of robotics in the next millennium* (Cat. No. 01CH37180). *IEEE*, vol 3, pp 1312–1320
- Priyantha NB, Chakraborty A, Balakrishnan H (2000) The cricket location-support system. In: *Proceedings of the 6th annual international conference on mobile computing and networking*, pp 32–43
- Niculescu D, Nath B (2003) Ad hoc positioning system (APS) using aoa. In: *IEEE INFOCOM 2003. Twenty-second annual joint conference of the IEEE computer and communications societies* (IEEE Cat. No. 03CH37428). *Ieee*, vol 3, pp 1734–1743
- Wang P, Xue F, Li H, Cui Z, Xie L, Chen J (2019) A multi-objective DV-Hop localization algorithm based on NSGA-II in Internet of Things. *Mathematics* 7(2):184
- Sun Z, Tao L, Wang X, Zhou Z (2015) Localization algorithm in wireless sensor networks based on multiobjective particle swarm optimization. *Int J Distr Sensor Netw* 11(8):716291
- Das P, Das DK, Dey S (2018) A new class toppler optimization algorithm with an application to data clustering. *IEEE Trans Emerging Topics Comput*
- Kaur A, Kumar P, Gupta GP (2018) Nature inspired algorithm-based improved variants of DV-Hop algorithm for randomly deployed 2D and 3D wireless sensor networks. *Wirel Pers Commun* 101(1):567–582
- Kanwar V, Kumar A (2021) Range free localization for three dimensional wireless sensor networks using multi objective particle swarm optimization. *Wirel Pers Commun* 117(2):901–921
- Kanwar V (2020) DV-Hop-based range-free localization algorithm for wireless sensor network using runner-root optimization. *J Supercomput*:1–18
- Sharma G, Kumar A (2018) Improved DV-Hop localization algorithm using teaching learning based optimization for wireless sensor networks. *Telecommun Syst* 67(2):163–178
- Singh SP, Sharma SC (2019) Implementation of a PSO based improved localization algorithm for wireless sensor networks. *IETE J Res* 65(4):502–514
- Shahbaz AN, Barati H, Barati A (2021) Multipath routing through the firefly algorithm and fuzzy logic in wireless sensor networks. *Peer-to-Peer Netw Appl* 14(2):541–558
- Mosavifard A, Barati H (2020) An energy-aware clustering and two-level routing method in wireless sensor networks. *Computing* 102(7):1653–1671
- Yousefpoor E, Barati H, Barati A (2021) A hierarchical secure data aggregation method using the dragonfly algorithm in wireless sensor networks. *Peer-to-Peer Netw Appl* 14(4):1917–1942
- Hasheminejad E, Barati H (2021) A reliable tree-based data aggregation method in wireless sensor networks. *Peer-to-Peer Netw Appl* 14(2):873–887
- Naghbi M, Barati H (2021) SHSDA: secure hybrid structure data aggregation method in wireless sensor networks. *J Ambient Intell Humanized Comput* 12(12):10769–10788
- Hajipour Z, Barati H (2021) EELRP: energy efficient layered routing protocol in wireless sensor networks. *Computing* 103(12):2789–2809
- Sharifi SS, Barati H (2021) A method for routing and data aggregating in cluster-based wireless sensor networks. *Int J Commun Syst* 34(7):e4754
- Kumar S, Lobiyal D (2013) An advanced DV-Hop localization algorithm for wireless sensor networks. *Wireless Personal Commun* 71(2):1365–1385
- Kanwar V, Kumar A (2020) Multiobjective optimization-based DV-Hop localization using NSGA-II algorithm for wireless sensor networks. *Int J Commun Syst* 33(11):e4431
- Mohanta TK, Das DK (2021) Class toppler optimization based improved localization algorithm in wireless sensor network. *Wirel Pers Commun*:1–20
- Liu B, Wang L, Jin Y-H, Tang F, Huang D-X (2005) Improved particle swarm optimization combined with chaos. *Chaos, Solitons Fractals* 25(5):1261–1271



39. Tian D, Zhao X, Shi Z (2019) Chaotic particle swarm optimization with sigmoid-based acceleration coefficients for numerical function optimization. *Swarm Evolution Computat* 51:100573
40. Vesterstrom J, Thomsen R (2004) A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In: *Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753)*. IEEE, vol 2, pp 1980–1987
41. Mohanta TK, Das DK (2022) Advanced localization algorithm for wireless sensor networks using fractional order class topper optimization. *J Supercomput*:1–29

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.