

Improved Wireless Sensor Network Localization Algorithm based on Selective Opposition Class Topper Optimization (SOCTO)

TAPAN KUMAR MOHANTA

ICFAI University Tripura

DUSHMANTA KR DAS (✉ dushmantakumardas29@gmail.com)

National Institute of Technology Nagaland <https://orcid.org/0000-0001-8030-6242>

Research Article

Keywords: DV-Hop, SOCTO, WSN, Localization error, LabVIEW software

Posted Date: February 15th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-861354/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Improved Wireless Sensor Network Localization Algorithm based on Selective Opposition Class Topper Optimization (SOCTO)

Tapan Kumar Mohanta¹ ·
Dushmanta Kumar Das^{2*}

Abstract One of the most important challenges in wireless sensor networks is the issue of localisation. Furthermore, it is critical to monitor and evaluate the data gathered. With the aid of a collection of constructed nodes known as beacons, the localization technique can measure node location. For a variety of factors, such as upkeep, lifespan, and breakdown, the fixed density of these beacons may be increased or decreased. Because of its robustness, flexibility, and economic viability, a well-known technique for locating wireless sensor network nodes is the Distance Vector-Hop (DV-Hop) algorithm. As a result, researchers continue to look for ways to develop it. Based on Selective Opposition Class Topper optimization (SOCTO), we propose an enhanced DV-Hop localization algorithm, it also focuses through an optimized formulation to compute the average hop-size with weight of beacon nodes in order to reduce the localization error within estimated distance between the beacon and the dumb node, due to improved localization accuracy. The range-free localization algorithm in uniform multi-hop wireless sensor networks is the subject of recent study. In various implementation scenarios, the results of simulations show that our proposed algorithm performs admirably the DV-Hop technique and DV-Hop related techniques.

Keywords DV-Hop • SOCTO • WSN • Localization error • LabVIEW software

Tapan Kumar Mohanta¹
Department of ECE, ICFAI University
Kamalgat, Tripura, 799210, India, e-mail: tapank838@gmail.com
Dushmanta Kumar Das^{2*}
Department of Electrical and Electronics Engineering, National Institute of Technology
Nagaland, 797103, India. e-mail: dushmantakumardas29@gmail.com.

1 Introduction

WSNs (wireless sensor networks) are becoming a common technology that is attracting research attention. WSNs are made up of a large number of tiny sensors called sensor nodes that assemble and communicate data from their sensing environment [1]. WSN localization is a critical problem that holds the collected data more usable for a variety of applications. Furthermore, without knowing the actual location information within each sensor node, the obtained data is meaningless [2]. Furthermore, the location of a data-sensing sensor is just as critical as the data itself. As a result, the data and position information must be communicated jointly such that the precise location of the collected information by the sensor node can be determined. Several applications designed to work in extreme environments, such as battlefields, border surveillance, rain forests [3,4], remote control of hazardous areas, routing [5–8], and so on, are increasingly using WSN. Although the Global Positioning System (GPS) is commonly used for sensor localization, it is not a viable choice due to WSN's cost constraints. As a result, researchers have proposed a variety of WSN localization algorithms [9]. There are numerous localization algorithms available today, that can be classified into two parts: range-based and range-free algorithms [10]. To calculate a node's location, range-free localization algorithms estimate the distance within nodes and do not need extra hardware. Under these types of algorithms, the closest nodes that are aware of their positions are used to localize dumb nodes. The range-based approach is split into few types: Time Of Arrival (TOA) [11], Time Difference Of Arrival (TDOA) [12], Received Signal Strength Indicator (RSSI) [13], and Arrival Of Angle (AOA) [14]. We're interested in range-free algorithms in our paper because they don't require any additional devices to range.

Nature has made a huge contribution to success of the modern algorithms for optimization and has also offered a solution to several complex engineering problems. There are numerous algorithms inspired by nature, just like Genetic Algorithm (GA), Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO), Simulated Annealing (SA) and Ant Colony Optimization (ACO). It has been found that the PSO is algorithm of quick processing. However, it is difficult for the algorithm to escape from local optima. SA has random initial disadvantages and there's less chance of finding global solutions. It is also noted that Genetic Algorithm (GA) has advantages over PSO. However, Genetic Algorithm (GA) has disadvantages of being sluggish and has better scope. From the above discussion, It is observed that the usefulness of these strategies depends also on problem under consideration. As one methodology can behave well during one kind of problem, it could be worse in many other problems. Because of their versatility, simplicity, and reliability, meta-heuristic optimization algorithms are widely used in both science and industrial problems. One of most current and commonly used algorithms in this field is Class Topper Optimization (CTO). Opposition-based learning (OBL) feature is included on CTO in this study to improve aggressive behaviour while holding

the convergence rate high. Instead of opposing all of the parameters in the class, only a few of the parameters are chosen to be opposed. This aids in avoiding pointless experimentation and achieving rapid convergence without jeopardizing the likelihood of discovering optimal solutions. The benefits of SOCTO are that the agents produce both large and small motions within, allowing for the simultaneous exploration of personal and global optima. Another important distinction between SOCTO and other algorithms is that the agents in SOCTO do not communicate with each other. As a result, class topper and section topper are both created by this algorithm to find the personal and global search at each iteration. We proposed a selective opposition class topper optimization based improved DV-Hop algorithm for WSNs to resolve the disadvantages of conventional DV-Hop localization.

The following are the most significant contributions of this work:

1. Using the least square process, the average hop distance between nodes is measured.
2. In the proposed algorithm, a beacon node weight is included to change the hop size of the beacon nodes.
3. The SOCTO algorithm is proposed to correct the approximate locations of the dumb nodes, resulting in a smaller localization error.
4. This involves detailed comparisons of the proposed optimized solution to current optimized DV-Hop-based algorithms. When compared to other algorithms, the results show that the proposed solution is more accurate.

2 Associated works

Because of its simplicity, feasibility, and low hardware requirements, DV-Hop is a common choice among WSN localization algorithms. However, the precision with DV-Hop is insufficient to meet more stringent requirements. Therefore, researchers have suggested a variety of improved DV-Hop-based schemes. In this section, some enhanced algorithms based on DV-Hop are briefly mentioned.

For DV-Hop, a reference node selection method and two improved algorithms were proposed by Gui et al. in [15]. The low precision of DV-Hop is attributable towards its multi-hop design and the method of measuring the faulty position, according to Gui. The authors looked into the third stage of DV-Hop and discovered that, if separate beacon nodes are added as the least square process's last reference node, the localization results are different. Then two new DV-Hop-based algorithms are presented, i.e., GOS DV-Hop and RAS DV-Hop. With the last reference node of a least square process, RAS DV-Hop uses every beacon node to take part in the localization of dumb nodes.

Also the applicant coordinates that have the closest distance to all beacons are chosen as the end result of the position. GOS DV-Hop employs a global enhancement technique inside the third stage for the DV-Hop and is capable

of greater localization over RAS DV-Hop. However, the difficulty is also increased by GOS DV-Hop. The final localization accuracy can be influenced by the fact that all beacon nodes are used as nodes of reference, which can contribute to some position error expansion. In [16], According to Kaur et al., the high localization reliability of DV-Hop would be due to the use of a linear approach, including the least square method, for solving non-linear formulas during localization. The authors then suggested an improved DV-Hop method that uses a non-linear approach called the Gauss-Newton process to decrease the DV-Hop accuracy. The initial two steps with the proposed methodology are equivalent to the original DV-Hop. The sum of the three beacon node coordinates closest to the dumb node is used as the approximate coordinates of the dumb node as in third stage of localization. In order to refine the dumb node's coordinates, the Gauss-Newton technique is then applied. This method increases the position accuracy of dumb nodes, but because of the unknown node's approximate distance error, the three beacon nodes selected are often not the closest ones. In [17], Tao et al. thought about that after studying how the DV-Hop measures the separation between nodes, various beacon nodes have different effects on the unknown node localization performance. The authors developed the field of weighted coefficients. The measured coefficient in DV-Hop represents the beacon node's effect on the measurement of dumb node coordinates. The measured coefficient is proportional to the number with dumb beacon node hop counts as well as the rise in hop counts, it decreases. The impact of the weighted coefficient is considered when measuring the hop distance from the dumb to beacon nodes, which encourages the calculated distance more accurately and enhances the accuracy of the localization. In heterogeneous networks, the weighted coefficient is inaccurate because of the variety of network topology, which contributes to the inaccuracy of the calculated distance. A new precise localization algorithm based on DV-Hop and differential evolution called DECHDV-Hop for WSN was suggested by Cui et al. in [18]. The number with hops between nodes is modified transforming the discrete value into the continuous value in the DECHDV-Hop algorithm, as a result of which the approximated range from the dumb node to the different beacon nodes is not the same when the number of hops among them is same. The precision of the localization outcomes is also improved because of the improved precision of the estimated distance. In addition, the localization process with the dumb node is abstracted by DECHDV-Hop as a problem of optimization reduces the calculated weighted square error with approximate distance from the dumb to the beacon node and to solve the issue, the problem is solved using the difference evolution algorithm.

In [19], an improved DV-Hop localization algorithm based on evolutionary algorithms is presented. The authors used the Shuffled Frog Leaping Algorithm (SFLA) to rectify the beacon node's age hop distance in the second step of DV-Hop to reduce the localization error. Various beacon nodes are given various weights when adjusting the minimum hop distance. More weights are allocated to the beacon nodes with greater localization influences. The hybrid genetic-particle swarm algorithm is implemented in the third stage to solve the

issue of the least square method's excessive localization error. The precision of localization has been improved, but it is important to change the control parameters of evolutionary algorithms. Different parameters have a huge effect on the accuracy of localization. Although the algorithm shows good efficiency, the algorithm spends more computational time using SFLA and hybrid Genetic-PSO. In [20], a better scheme called IDV-Hopping with TLBO is suggested. The minimum hop distance with the beacon node is corrected making use of a correction factor in the improved algorithm. The collinearity principle is added to reduce the position error induced by node collinearity. The unknown node that was effectively positioned in the initial round is promoted to a auxiliary beacon node in order to localize further unknown nodes. At last, to enhance localization precision, the unknown node's coordinates calculation is derived as an optimization problem, as well as Teaching-Learning-Based Optimization (TLBO) algorithm is used to resolve the issue, which is parameter-independent [20].

In the literature, there was a suggestion for a quickest recursive path routing technique for sensor networks localization [21,22]. In multi-hop wireless sensor networks, the quoted recursive algorithm objectives are to calculate an estimate of ranges among any two sensors. This method involves calculating all possible connections between the dumb node (sensor of origin) and the beacon node (sensor of destination) with least number of hops utilizing recursive functions. The mean of all paths is then calculated using a distance matrix in order to approximate distance between dumb nodes and beacons. Both recursive and ad hoc routing dependent localization are used in sensor network techniques [23], Guided Position Estimation [24,25] algorithms, and Recursive Position Estimation [26,27] are some of the recursive models that have been improved in the literature.

To our understanding, there is no optimization algorithm that employs both an optimization model, as well as an adjustment to the minimum hop as an enhancement of the DV-Hop algorithm behind the research. For improved estimation accuracy, we suggest a selective opposition class topper optimization DV-Hop location method with minimum hop optimization in this paper.

3 Algorithm of conventional DV-Hop

The localization technique is essentially a free localization technique that is based on a protocol for distance routing. For number of hops of beacon nodes as well as the minimum hop distance besides WSNs, calculation of the distance between dumb nodes or unknown nodes and beacon nodes is done. Different paths form in a network topology among dumb nodes and beacon nodes that are not linear due to non-uniform connectivity with wireless sensor nodes. Therefore, Some errors have been identified at the period of the algorithm in

the node position method [28].

Step 1 The minimum hop amount is defined for unknown nodes and beacon nodes of step 1. By transmitting signals through beacon nodes by vector protocol system, the neighbouring nodes can be shown their location, Information exists in the form of H_i, a_i, b_i, id , in which id will be the identity, a_i, b_i have been coordinates, and H_i has been the hop count for the i^{th} beacon node.

First, 0 is set of the value for H_i [29]. The nodes obtain data from the broadcast and record the hop amount and localization of the vector's beacon nodes. The value of H_i must be increased by 1 through this process [20]. In this updation 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 obtained broadcast data and keep track of the hop amount and localization for the beacon nodes of the vector.

Step 2 Minimum hop count and average hop distance were determined to find the distance between unknown nodes and beacon nodes. In this method, the average hop distance for the entire network can be determined by obtaining the position and hop amount for nodes of beacons, as described in the previous stage. Then this data is transmitted to the entire network. For most nodes, The minimum hop distance data from the beacon node closer to them is also necessary to obtain [30]. The following equation provides the typical distance of the hop (jp_i) and the hop range (p_i) between the $i(a_i, b_i)$ beacon node and the other (beacon) node $j(a_j, b_j)$ can just be computed as:

$$jp_i = \frac{\sum \sqrt{(a_i - a_j)^2 + (b_i - b_j)^2}}{\sum h(ij)}. \quad (1)$$

The distance between beacon nodes and dumb nodes is expressed in the following formula.

$$p_{iu} = jp_{iu} \times Hop_{min}, \quad (2)$$

where jp_i is the average hop distance, Hop_{min} is the hop count between the i beacon nodes and the u dumb nodes.

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

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

$$\begin{aligned}
 a_1^2 - a_n^2 + 2(a_1 - a_n)a + b_1^2 - b_n^2 - 2(b_1 - b_n)b \\
 = p_1^2 - p_n^2; \\
 &\dots\dots\dots \\
 a_{n-1}^2 - a_n^2 + 2(a_{n-1} - a_n)a + b_{n-1}^2 - b_n^2 - 2(b_{n-1} \\
 - b_n)b = p_{n-1}^2 - p_n^2;
 \end{aligned}
 \tag{4}$$

(3) can be arranged from $SA = T$ in the matrix, where

$$\begin{aligned}
 S &= \begin{bmatrix} 2(a_1 - a_n) & 2(b_1 - b_n) \\ 2(a_2 - a_n) & 2(b_1 - b_n) \\ \dots\dots\dots \\ 2(a_{n-1} - a_n) & 2(b_1 - b_n) \end{bmatrix}, \\
 T &= \begin{bmatrix} a_1^2 - a_n^2 + b_1^2 - b_n^2 + p_n^2 - p_1^2 \\ a_2^2 - a_n^2 + b_2^2 - b_n^2 + p_n^2 - p_2^2 \\ \dots\dots\dots \\ a_{n-1}^2 - a_n^2 + b_{n-1}^2 - b_n^2 + p_n^2 - p_{n-1}^2 \end{bmatrix}, \\
 A &= \begin{bmatrix} a \\ b \end{bmatrix},
 \end{aligned}$$

Finally, the last square approximation is used to calculate the unknown node coordinates as given below:

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

4 A Selective opposition class topper optimization (SOCTO) Algorithm

Because of their versatility, simplicity, and robustness, metaheuristics are widely used for optimization in both experimental and industrial problems. One of the most recent and widely used algorithms in this field is Class Topper Optimization (CTO) [31]. Opposition-based learning (OBL) [32] is coupled with CTO in this study to improve exploratory behavior while keeping the convergence rate high. Selective Opposition based Class Topper Optimization is the

name given to the proposed algorithm (SOCTO). The class is initialized with the amount with search agents, their top and bottom bounds, and their dimensions. The class vectors are chosen at random while keeping the constraints in mind. Using opposition to generate a new student solution (class topper or section topper), we can look for the class's opposite location (with respect to the student whose opposition is made). With this in mind, it is reasonable to conclude that the best solution (class topper) location is the best in the class. The students will miss out on the chance to learn from the class topper. The following is a list of a student's computation knowledge:

Each student in each section learns the following in accordance with their respective section topper (S_t):

$$L_1^{t+1} = W \times L_1^t + C \times \phi_1 \times (S_t - S), \quad (5)$$

$$S_t^{t+1} = S_t^t + L_1^{t+1}. \quad (6)$$

There is now a section topper (S_t) to allow for progress away from the class topper (C_t). This empowers a more targeted application of Opposition Based Learning (OBL). As a result, the section topper has become the best student in the section. Section Topper (S_t) is taught by Class Topper (C_t).

$$L_2^{t+1} = W \times L_2^t + C \times \phi_2 \times (C_t - S_t), \quad (7)$$

$$S_t^{t+1} = S_t^t + L_2^{t+1}. \quad (8)$$

A student's learning improves the section's consistency. Some constants in the algorithm will be calculated in advance, such as W , C , ϕ_1 , ϕ_2 . Every student's cause is drawn towards the performance index (PI) by the stochastic acceleration term C , and the inertial weight factor W . It successfully accelerates as well as increases the likelihood of reaching the global optimum by employing a selective OBL strategy in CTO. After each updated version, a fresh set of student solutions consisting of Class Topper (C_t), Section Topper (S_t) and least fit student s . The best candidate among these solutions is class topper; then section topper and the least fit student. As a result, it's possible that the (C_t) student is in the opposite place in the class from the optimum. As a result, the ranked correlation coefficient of each of the (S_t) student solutions with respect to the (C_t) solution is determined after each iteration. If the coefficient value is below zero, indicating that the points are negatively correlated, the ct search agent's stance is opposed. The Spearman coefficient aids in the selection of students to be objected, as well as the aspects in which they are to be opposed, thus narrowing and controlling our search and increasing its efficiency. For each iteration, the value of a threshold parameter (threshold) is reduced linearly. This value indicates whether the S students' attribute is near or far from the (C_t). The difference between the given vector z and the vector z_{ct} is determined using each aspect (i) with a given vector z as well as the vector z_{ct} , the distinction among them is determined with the following formula:

$$diff(i) = |z(i) - z_{ct}(i)|. \quad (9)$$

If the value of $diff(i)$ exceeds the threshold, that number of parameters with an excerpted j_{no} is increased, i.e. $j_{no} = j_{no} + 1$. Let n be the complete dimension. If $(n - j_{no}) < j_{no}$, the dimensions with a discrepancy between z and z_{ct} is greater than the threshold are in opposition used in following equation:

$$\bar{z}(j) = ua(j) + la(j) - z(j). \quad (10)$$

where $j \in \{i : diff(i) \leq threshold\}$, \bar{z} is the subsequent vector, $ua(j)$ is the maximum bound of j^{th} dimension with in the existing population, and $la(j)$ is the minimum bound of j^{th} dimension in the existing population. In general, OBL is applied by using the function's maximum and minimum bounds. Let \bar{z} be the ideal dimension value and $\bar{z} + \delta$ be the current value, where δ is a very small amount. On opposition, the value becomes $b_j + a_j - \bar{z} - \delta$, which may or may not be close to the best solution unless $\bar{z} = (b_j + a_j)/2$. As a result, applying opposition learning when the method converges can cause needless exploration, which can be detrimental. The bounds, which are derived from the class are used for opposition of a student to achieve a balance between discovery and exploitation.

The chosen S students are then pitted against each other. If the fitness is higher, the vectors are replaced by opposition vectors; otherwise, the vectors remain unchanged. The updated set of student solutions is used to evaluate the C_t and S_t vectors. Furthermore, the opposition is focused on the student's upper and lower limits rather than the class, this drawback is further remedied.

4.1 Proposed selective opposition class topper optimization (SOCTO) for enhanced DV-Hop localization

We propose a five-step enhanced DV-Hop localization algorithm based on selective opposition class topper optimization (SOCTO) in this section of the paper.

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, the effective will be the approximate positions.

Step 2. Instead of using the conventional approach, we define to measure the average hop distance among beacons. As a result, we use the polynomial approximation to reduce estimated location error and increase localization accuracy. The following polynomial is used to calculate the approximate hop distance from beacons i and other beacons m .

$$r_{ij} = \kappa_0 + \kappa_1 p_{ij} + \kappa_2 p_{ij}^2, j = 1, 2, 3, \dots, m \quad (11)$$

where κ_0 , κ_1 and κ_2 are the coefficients Eq. (11) can be represented in matrix form as follows:

$$\begin{bmatrix} p_{i1}^2 & p_{i1} & 1 \\ p_{i2}^2 & p_{i2} & 1 \\ \vdots & \vdots & \vdots \\ p_{ij}^2 & p_{ij} & 1 \end{bmatrix} \begin{bmatrix} \kappa_2 \\ \kappa_1 \\ \kappa_0 \end{bmatrix} = \begin{bmatrix} r_{i1} \\ r_{i2} \\ \vdots \\ r_{ij} \end{bmatrix}, \quad (12)$$

where p is the number of hops among beacon nodes. We solve the following equation to find its polynomial function of estimate of distance among nodes:

$$\kappa = (P^T P)^{-1} P^T R, \quad (13)$$

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

$$r_{ij} = \kappa_0 + \kappa_1 p_{ij} + \kappa_2 p_{ij}^2, \quad (14)$$

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. As a result, we get the following matrix form:

$$r_{est} = hop \times \kappa, \quad (15)$$

We can use least square approximation [34] to solve Eq. (21). The true distance between beacon nodes i and j shall be calculated by

$$r_{true} = \sqrt{(a_i - a_j)^2 + (b_i - b_j)^2}. \quad (16)$$

Error between nodes i and j of the beacon is given as:

$$r_{error} = r_{est} - r_{true}. \quad (17)$$

Now, we have added a rectification factor, and this is defined as:

$$\tau = \frac{r_{error}}{s}, \quad (18)$$

where s is the number of beacon nodes.

The rectification factor τ 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 j 'th dumb node is determined as [33]:

$$r_{ik}^{Mod} = (HopSize + \tau) \times H_{ik}. \quad (19)$$

.

The following example demonstrates the benefits of our suggested methodology for correction. Assume a collection of sensor nodes, as shown in Fig.

1, consisting of beacons $B1$, $B2$, $B3$, and $B4$, denoted by red star symbols, and dumb nodes denoted by pink star symbols. The above sensor nodes are distributed at random within a 20×20 meters sensing area with a connectivity range meters. Beacons are aware of their locations, distances, as well as the number of hops between them. Table 1 and 2 illustrate the shortest hop count and the actual distances among beacons. All of the following simulations are carried out using LabVIEW@2015. The suggested optimization relates to the HopSize estimate. We estimate the hop ranges (HopSize) of beacons in the network model of this example using Eq. (1) of the initial DV-Hop algorithm and displays in Table 3.

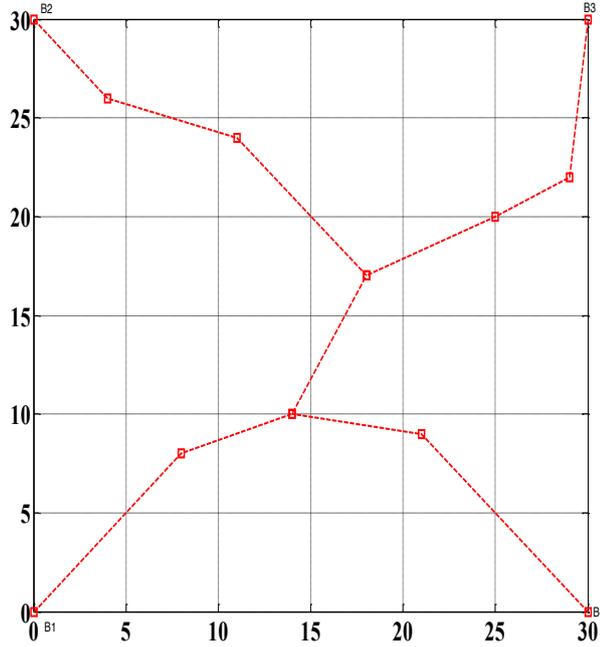


Fig. 1: The hop-size optimization method.

Now, using our suggested optimization, we hope to determine approximate distances among beacons using polynomial approximation in Eq. (11). The following are the actual distances among beacons:

$$30 = \kappa_0 + 6\kappa_1 + 36\kappa_2;$$

$$42.42 = \kappa_0 + 6\kappa_1 + 36\kappa_2;$$

Table 1: The number of hops among beacons should be kept to a minimum.

Count the hops	B1	B2	B3	B4
B1	0	6	6	4
B2	6	0	6	6
B3	6	6	0	6
B4	4	6	6	0

Table 2: The actual distance among beacons.

True distance (meters)	B1	B2	B3	B4
B1	0	30	42.42	30
B2	30	0	30	42.42
B3	42.42	30	0	30
B4	30	42.42	30	0

Table 3: Beacons' hopSize.

Beacon	B1	B2	B3	B4
HopSize	6.40	5.69	5.69	5.69

Table 4: DV-Hop distance estimation among beacons.

r_{Dv-Hop}^{est}	B1	B2	B3	B4
B1	0	60.10	34.14	72.76
B2	60.10	0	84.14	51.04
B3	34.14	84.14	0	55.14
B4	72.76	51.04	55.14	0

$$\begin{aligned}
30 &= \kappa_0 + 4\kappa_1 + 16\kappa_2; \\
30 &= \kappa_0 + 4\kappa_1 + 16\kappa_2; \\
42.42 &= \kappa_0 + 6\kappa_1 + 36\kappa_2; \\
30 &= \kappa_0 + 6\kappa_1 + 36\kappa_2.
\end{aligned} \tag{20}$$

Eq. (12) can be written as follows:

$$\begin{bmatrix} 30 \\ 42.42 \\ 30 \\ 30 \\ 42.42 \\ 30 \end{bmatrix} = \begin{bmatrix} 1 & 6 & 36 \\ 1 & 6 & 36 \\ 1 & 4 & 16 \\ 1 & 4 & 16 \\ 1 & 6 & 36 \\ 1 & 6 & 36 \end{bmatrix} \begin{bmatrix} \kappa_0 \\ \kappa_1 \\ \kappa_2 \end{bmatrix} \tag{21}$$

Table 5: Estimation of approach distance among beacons proposed.

r_{ik}^{Mod}	B1	B2	B3	B4
B1	0	28.10	40.14	28.76
B2	28.10	0	40.14	28.04
B3	28.76	28.04	0	26.14
B4	27.6	28.04	26.14	0

A least square approximation is used to measure coefficients κ_0 , κ_1 and κ_2 as shown in Eq. (13):

$$\begin{cases} \kappa_0 = -26.76; \\ \kappa_1 = 25; \\ \kappa_2 = -2.45. \end{cases}$$

We calculate approximate distances among beacons after measuring polynomial approximation coefficients including HopSize of beacons. Table 4 and 5 display outcomes for the DV-Hop technique, r_{Dv-Hop}^{est} , and the proposed technique, r_{ik}^{Mod} , respectively. These tables show that the estimated ranges resulting from our suggested technique are nearer distance than those obtained by using the DV-Hop technique. It is possible to conclude that HopSize values are incorrect, and thus the DV-Hop technique will not provide accurate approximate locations of dumb nodes.

Step 3. To calculate the location of dumb nodes, the $2D$ hyperbolic position technique is used. Let (a_u, b_u) be the location of the dumb node and (a_p, b_p) be the position of the i^{th} beacon node. The following formula is used to calculate the distance between these nodes:

$$r_{pu}^2 = (a_p - a_u)^2 + (b_p - b_u)^2. \quad (22)$$

If $X_p = a_p^2 + b_p^2$ and $Y_u = a_u^2 + b_u^2$,

$$r_{pu}^2 - X_p = -2a_p a_u - 2b_p b_u Y_u. \quad (23)$$

(23) can be expressed as:

$$AZ = q, \quad (24)$$

$$Z = [a_u, b_u, Y_u]^T, \\ A = \begin{bmatrix} -2a_1 & -2b_1 & 1 \\ -2a_2 & -2b_2 & 1 \\ \dots & \dots & \dots \\ -2a_n & -2b_n & 1 \end{bmatrix}, q = \begin{bmatrix} r_{1u}^2 - X_1 \\ r_{2u}^2 - X_2 \\ \cdot \\ \cdot \\ r_{nu}^2 - X_n \end{bmatrix}.$$

From (24), we can obtain Z value with the help of least mean square estimated method:

$$Z = (A^T A)^{-1} A^T q. \quad (25)$$

Next, the coordinates of dumb node u are

$$\begin{aligned} a_u &= z(1), \\ b_u &= z(2). \end{aligned} \quad (26)$$

Step 4. To rectify the approximate positions of a dumb nodes, we used the SOCTO algorithm. SOCTO's DV-Hop localization objective feature is as follows:

$$f(a, b) = \min \left(\sum_{i=1,2,\dots,M} \left| \sqrt{(a - a_i)^2 + (b - b_i)^2} - p_{ik}^{Mod} \right| \right). \quad (27)$$

Fig. 2 illustrates a flowchart of the proposed localization algorithm.

5 Simulation and Performance evaluation

Under this segment, we compare the efficiency of the suggested algorithm to the initial DV-Hop algorithm as well as other algorithms inside the literature (UDV- Hop [26] and IR- DV-Hop [27] localization algorithm). For the execution and modeling with our algorithm, as well as DV-Hop and its enhancements, we use LabVIEW@2015. To improve simulation, we run 100 randomly chosen node companies plan and average the values for all evaluations. Sensors (beacons and dumb nodes) are chosen randomly inside an isotropic $250 \times 250 m^2$ sensor network in our computations. For every node has same 15 meter messaging radius. We try three different installations for the distribution with beacon nodes with in area: random, spiral, and circular deployment.

The node distribution is viewed in Fig. 3, 4 and 5 the red circle represents the beacon node, while the blue pentagram represents the dumb nodes. The approximation error with localization is represented by the blue straight line. The positioning error used to evaluate the performance with localization techniques written as follows:

$$ALE = \frac{\sum_{i=1}^{N-M} \sqrt{(a_{iest} - a_{iact})^2 + (b_{iest} - b_{iact})^2}}{R * (N - M)}, \quad (28)$$

where the approximate and true coordinates with the i^{th} dumb node are indicated by (a_{iest}, b_{iest}) and (a_{iact}, b_{iact}) , the connectivity range of a sensor nodes is denoted by R , the total number of nodes is denoted by N , and the number of beacon node is denoted by M . Any localization algorithm can be shown to have higher precision with lower ALE values. According to Eq. (28), the localization error is proportional to the number with dumb nodes, the number of beacons, as well as the connectivity radius with nodes. As an outcome, our simulated results are based on three parameters. Throughout all studies, we use random anchor deployment and same sensor node configurations to obtain

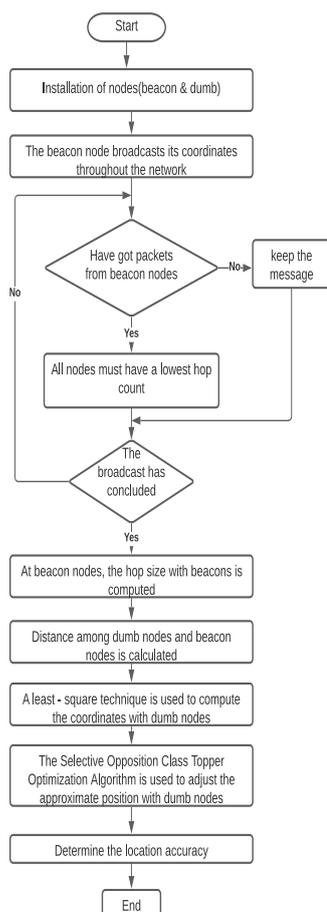


Fig. 2: The suggested algorithm's flowchart

a comparative analysis of our algorithm's localization accuracy with initial DV-Hop [34], UDV-Hop [26], and IR- DV-Hop [27] localization algorithms. We compare our proposed approach to the DV-Hop and IR-DV algorithms by adopting different beacon node deployment strategies, such as random, circular, and spiral distributions, as shown in Fig. 3, 4 and 5 respectively.

Fig. 6, 7 and 8 shows the average localization error produced by the DV-Hop, IR-DV, and suggested methods when random beacon nodes, circular beacon nodes, and spiral beacon nodes are deployed versus the total number of nodes. The suggested technique outperforms other current algorithms with various beacon node deployment schemes, as shown in Table 6.

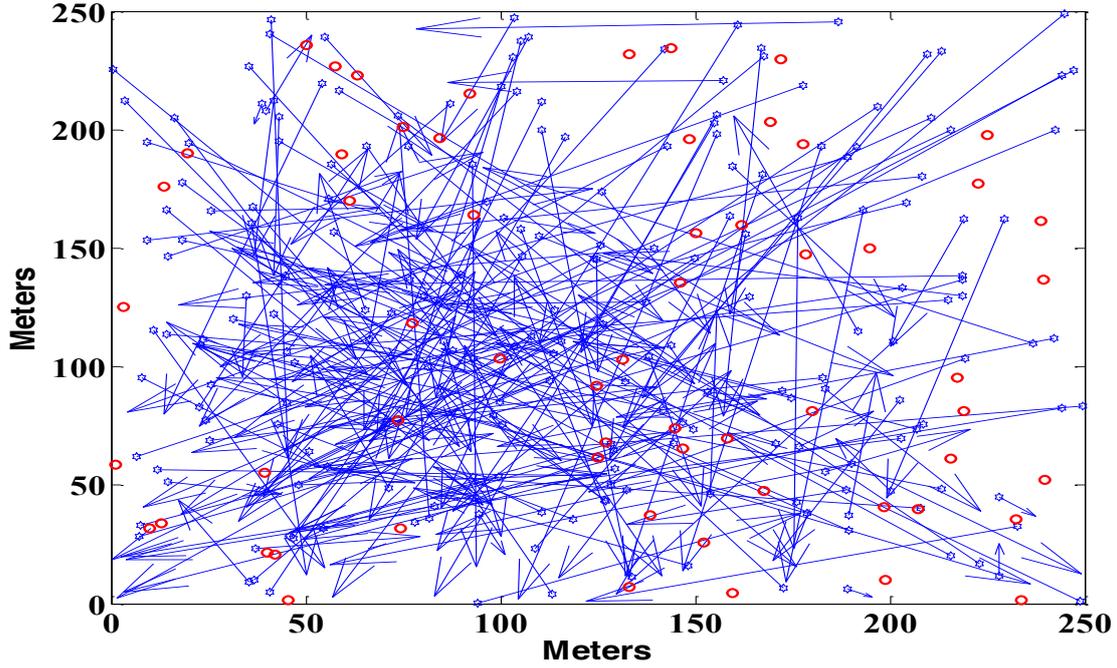


Fig. 3: The distribution of sensor nodes produced a localization result with a random deployment of beacon nodes.

Table 6: Comparative analysis of the suggested algorithm's impact on beacons from random, circular and spiral beacon nodes deployment with the help of certain existing algorithms.

No of nodes		200	250	300	350	400	450	500
Proposed	Random (%)	0.2	0.19	0.16	0.13	0.14	0.048	0.048
	Circular (%)	0.18	0.18	0.15	0.15	0.19	0.14	0.1
	Spiral (%)	0.23	0.11	0.21	0.25	0.25	0.06	0.17
IR-DV [27]	Random (%)	0.25	0.23	0.2	0.15	0.16	0.051	0.051
	Circular (%)	0.22	0.22	0.17	0.17	0.23	0.16	0.11
	Spiral (%)	0.31	0.13	0.26	0.33	0.33	0.07	0.20
DV [34]	Random (%)	1.65	1	1	1.07	1.16	0.89	1

5.1 The cumulative number of nodes impacted by localization error for random distribution deployment of beacons

As the number of nodes rises, Table 7 illustrates that the suggested algorithm is superior to other existing methods [34,26,27]. It may be seen in Fig. 3. The proposed method offers a significantly higher accuracy gain than its competi-

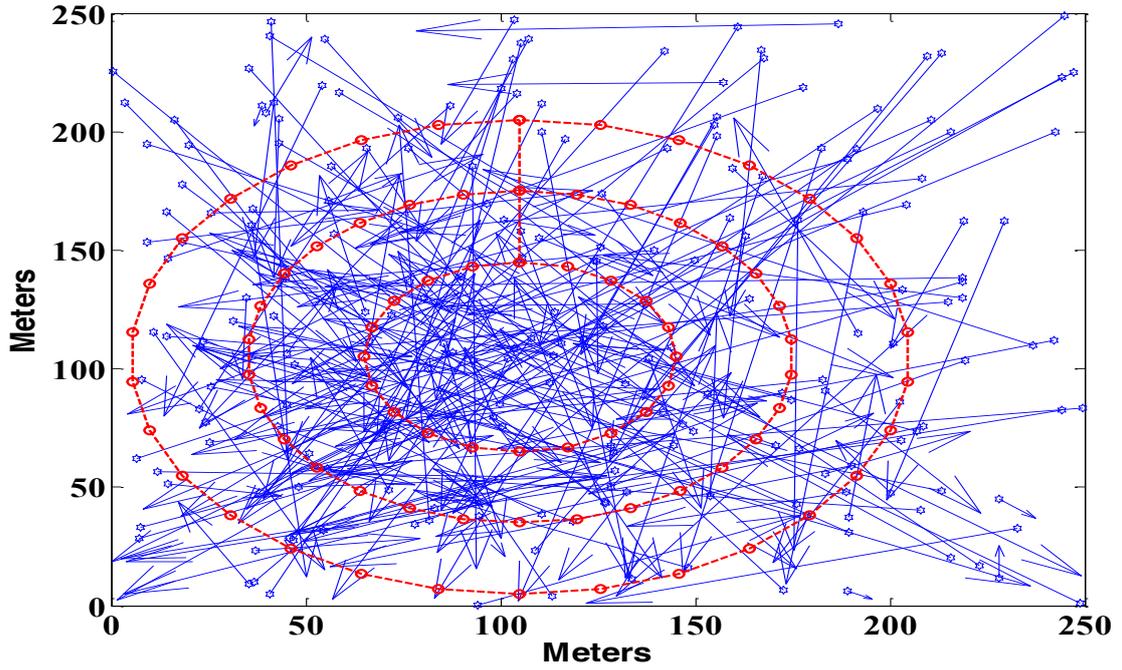


Fig. 4: The distribution of sensor nodes was localized using a circular beacon nodes deployment.

Table 7: A comparison of the effects of the proposed algorithm on random beacon node deployment with some existing algorithms.

No of nodes		200	250	300	350	400	450	500
Proposed	ALE (%)	.07	.018	.04	.04	.1	.019	.125
IR DV Hop [27]	ALE (%)	.08	.019	.04	.05	.11	.02	0.14
UDV Hop [26]	ALE (%)	.32	.038	.04	.2	.16	.025	.21
Basic DV Hop [34]	ALE (%)	1.12	.61	.66	1	1.19	0.95	1.75

tors. In fact, the suggested methodology allows sensors to assess their locations with a lower localization error.

5.2 The effect of localization error for random distribution deployment by ratio of beacons

Table 8 shows that the proposed algorithm outperforms other existing algorithms as the ratio of beacon nodes increases. The suggested algorithm's and

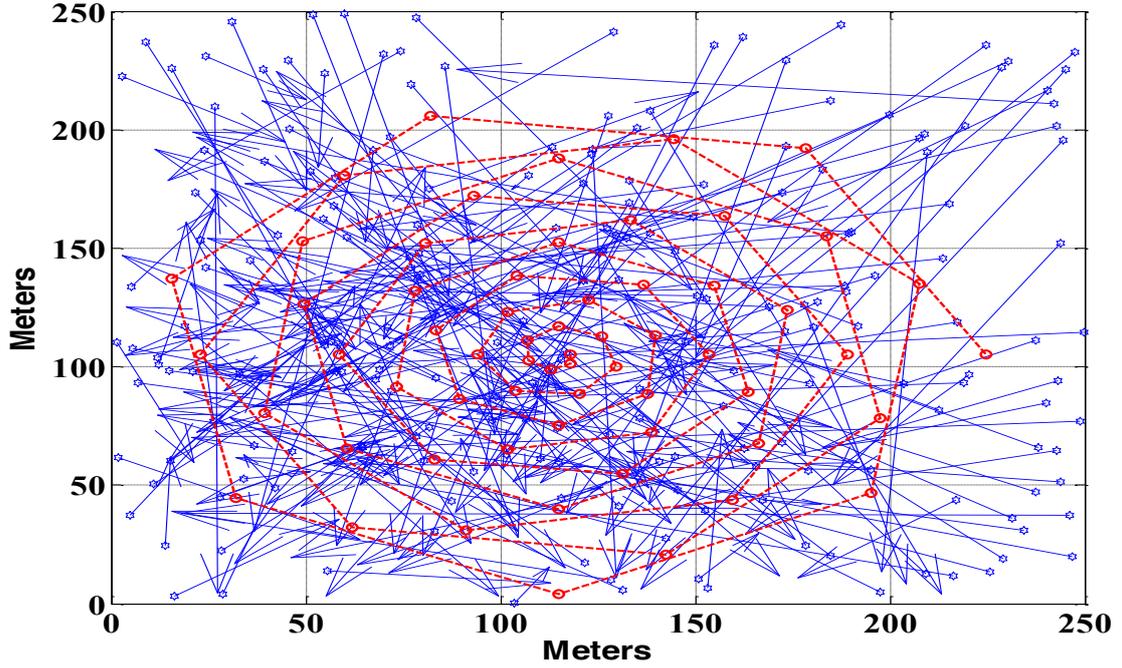


Fig. 5: The distribution of sensor nodes was localized using spiral beacon nodes deployment.

Table 8: A comparison of the proposed algorithm's effect on beacon node percentage from ALE with some existing algorithms.

percentage of beacon nodes	5	10	15	20	25	30
Proposed	.14	.09	.04	.15	.15	.22
IR DV Hop [27]	.17	.1	.05	.17	.18	.29
UDV Hop [26]	.26	.15	.10	.23	.25	.35
DV Hop [34]	1.04	1	1.10	1.23	1.43	1.45

its equivalents' localization errors are plotted against the proportion of beacon nodes in Fig. 10.

5.3 Communication radius of nodes effect by localization error for random distribution deployment

Table 9 shows that the proposed algorithm outperforms other existing algorithms when the broadcast range is considered. Fig. 10 depicts the fluctuation

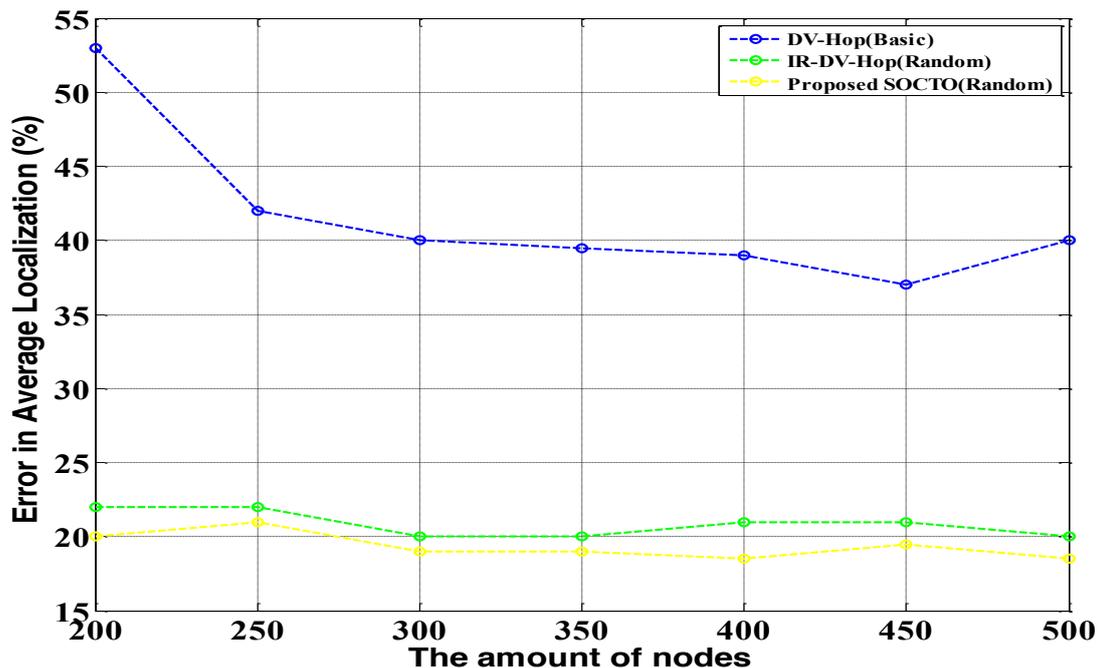


Fig. 6: The DV-Hop, IR-DV, and suggested algorithms' localization errors vs the total number of nodes with random distributions of beacon nodes.

Table 9: A comparison of the proposed algorithm's effect on range of communication from ALE with some existing algorithms

percentage of beacon nodes	15	20	25	30	35	40
Proposed	.09	.11	.14	.2	.23	.23
IR DV Hop [27]	.10	.13	.17	.25	.30	.30
UDV Hop [26]	.26	.53	.44	.66	.73	.73
DV Hop [34]	1.21	1.4	1.4	1.8	1.81	1.82

of the communication radius of nodes versus the localization error achieved by our proposed localization method and other algorithms (beacons and dumb nodes).

6 Conclusion

Selective Opposition Class Topper Optimization (SOCTO) algorithm is proposed for multi-hop WSNs in this work. This study improves on the original

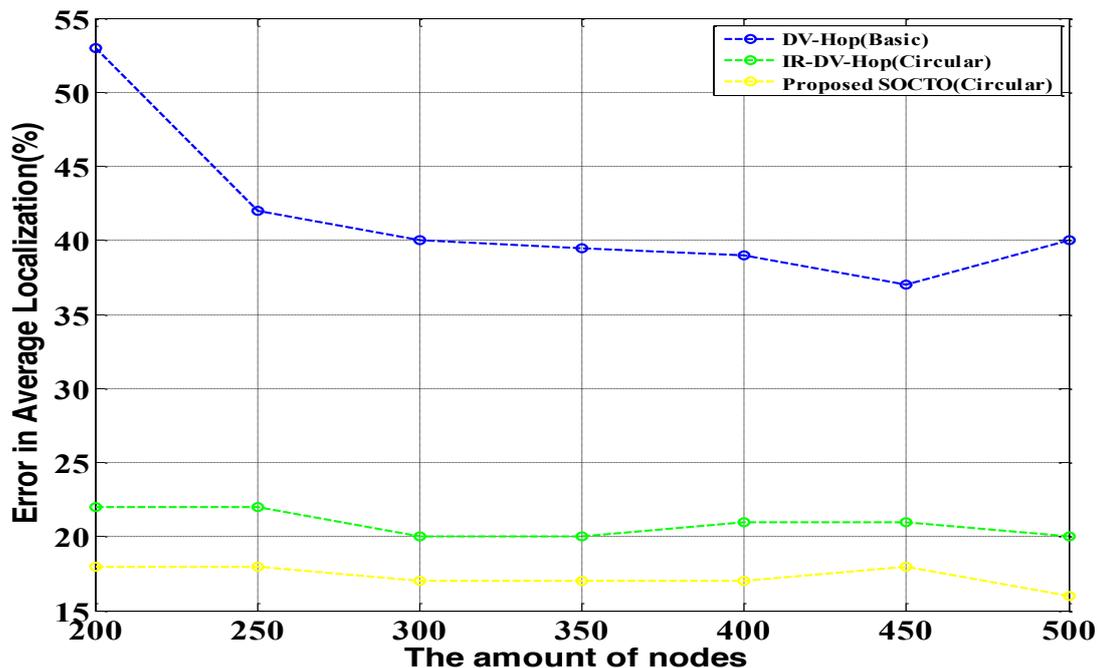


Fig. 7: The DV-Hop, IR-DV, and suggested algorithms' localization errors vs the total number of nodes with circular distributions of beacon nodes.

DV-Hop localization algorithm by using a polynomial approximation to optimize the average distance of hop computation of beacon nodes. The proposed approach achieves a smaller localization error and beats the original DV-Hop and other localization algorithms inside the literature in terms of accuracy using simulation results. We also check the result of our algorithm's localisation under 3 distinct beacon deployments: random, circular, and spiral. According to simulation data, the spiral anchor deployment produces better localization accuracy than the latter. Development of the spiral beacon nodes deployment of the multi-hop-based approach in a three-dimensional system and on the physical staging ground is also one of the potential research topics to be worked out.

7 Declaration

7.1 Funding

: Not applicable

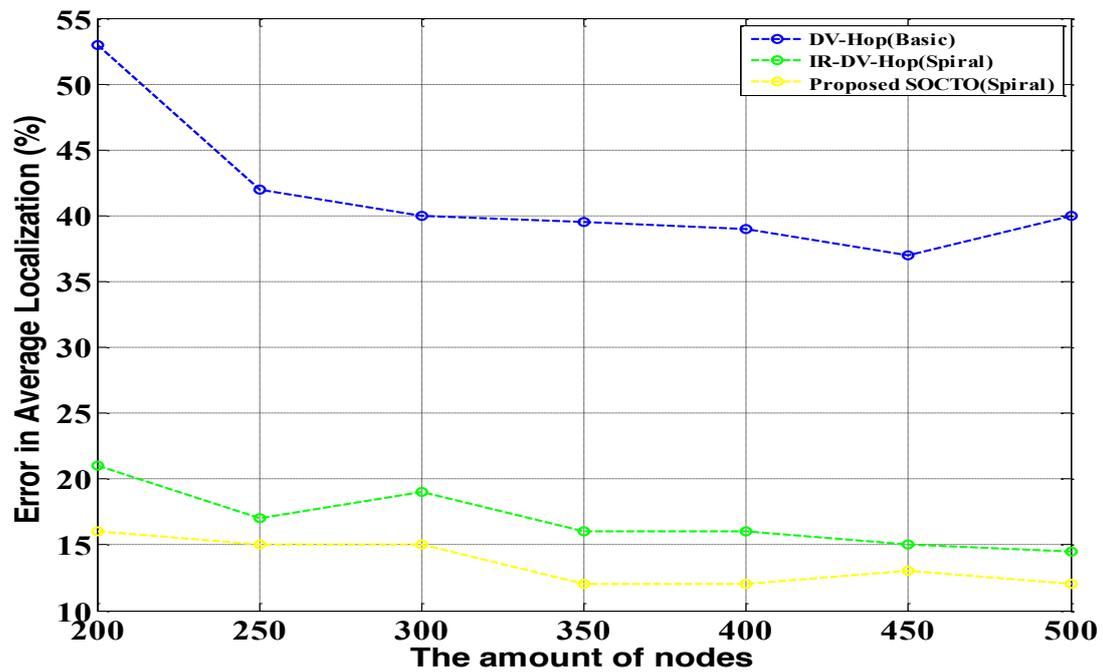


Fig. 8: The DV-Hop, IR-DV, and suggested algorithms' localization errors vs the total number of nodes with spiral distributions of beacon nodes.

7.2 Conflicts of interest/Competing interests

: Not applicable

7.3 Availability of data and material

: Not applicable

7.4 Code availability

: Not applicable

7.5 Authors' contributions

: Not applicable

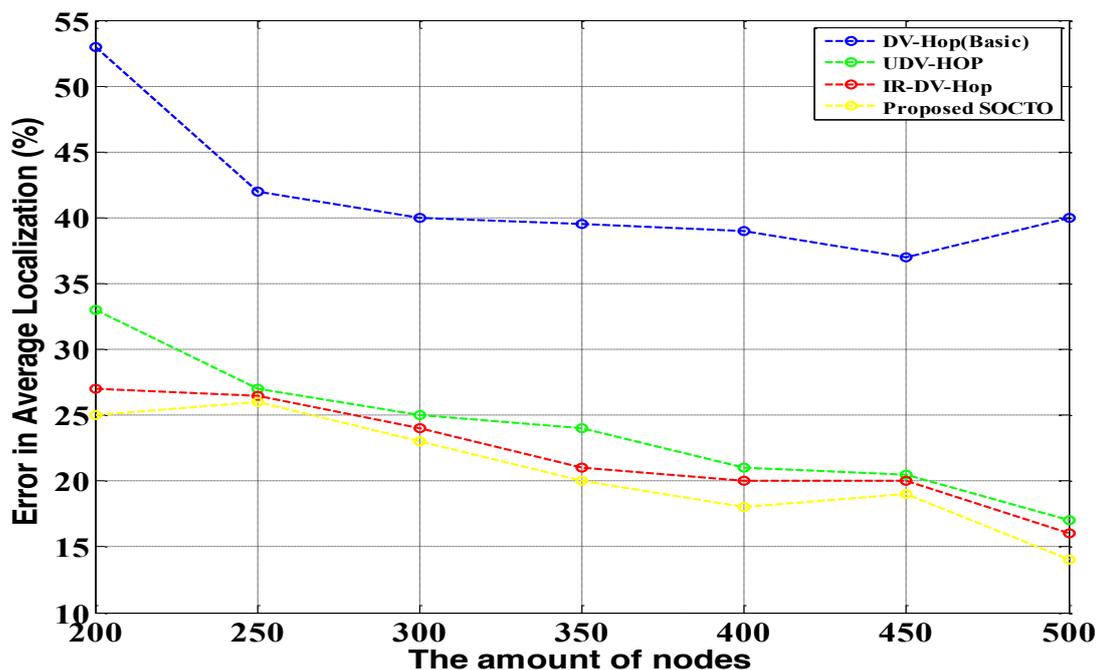


Fig. 9: Total number of nodes versus error in localization.

References

- [1] H. Dai, A. Chen, X. Gu, and L. He, "Localisation algorithm for large-scale and low-density wireless sensor networks," *Electronics letters*, vol. 47, no. 15, pp. 881–883, 2011.
- [2] T. Najeh, H. Sassi, and N. Liouane, "A novel range free localization algorithm in wireless sensor networks based on connectivity and genetic algorithms," *International Journal of Wireless Information Networks*, vol. 25, no. 1, pp. 88–97, 2018.
- [3] Y. K. Joshi and M. Younis, "Restoring connectivity in a resource constrained wsn," *Journal of Network and Computer Applications*, vol. 66, pp. 151–165, 2016.
- [4] S.-M. Lee, H. Cha, and R. Ha, "Energy-aware location error handling for object tracking applications in wireless sensor networks," *Computer Communications*, vol. 30, no. 7, pp. 1443–1450, 2007.
- [5] B. Gavish and I. Neuman, "Routing in a network with unreliable components," *IEEE Transactions on Communications*, vol. 40, no. 7, pp. 1248–1258, 1992.
- [6] W.-H. Liao, J.-P. Sheu, and Y.-C. Tseng, "Grid: A fully location-aware routing protocol for mobile ad hoc networks," *Telecommunication systems*, vol. 18, no. 1, pp. 37–60, 2001.
- [7] P. A. Contla and M. Stojmenovic, "Estimating hop counts in position based routing schemes for ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1, pp. 109–118, 2003.
- [8] P. K. Sahu, E. H.-K. Wu, J. Sahoo, and M. Gerla, "Ddor: Destination discovery oriented routing in highway/freeway vanets+," *Telecommunication Systems*, vol. 50, no. 4, pp. 267–284, 2012.

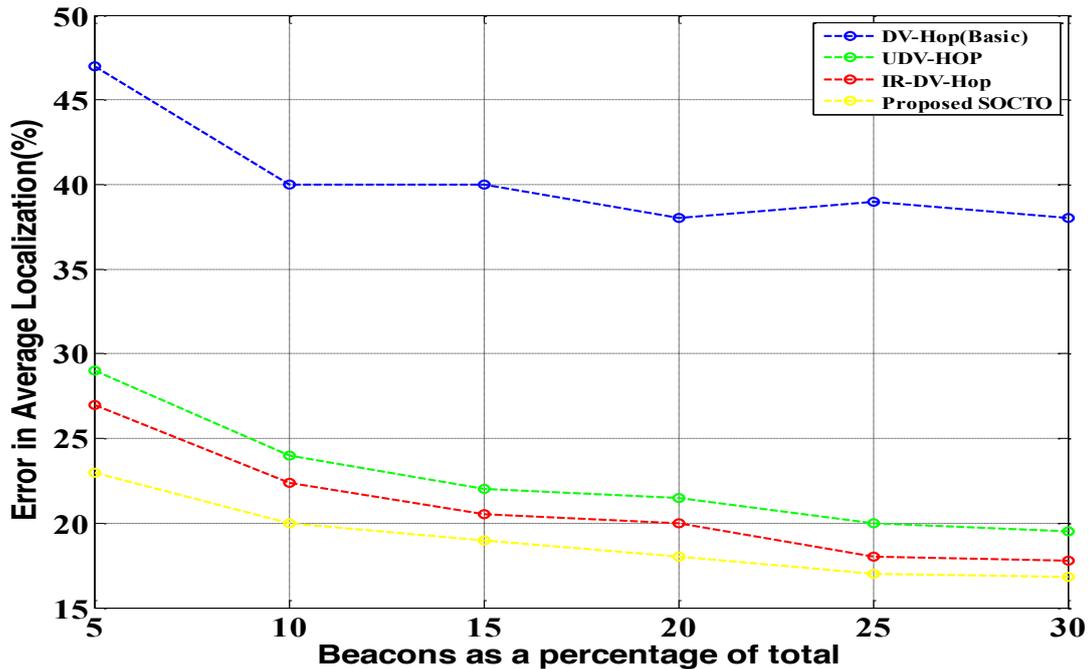


Fig. 10: beacon node percentage against localization error.

- [9] G. Song and D. Tam, "Two novel dv-hop localization algorithms for randomly deployed wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 11, no. 7, p. 187670, 2015.
- [10] G. Q. Gao and L. Lei, "An improved node localization algorithm based on dv-hop in wsn," in *2010 2nd international conference on advanced computer control*, vol. 4. IEEE, 2010, pp. 321–324.
- [11] L. Girod and D. Estrin, "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 the Next Millennium (Cat. No. 01CH37180)*, vol. 3. IEEE, 2001, pp. 1312–1320.
- [12] D. Niculescu and B. Nath, "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)*, vol. 3. Ieee, 2003, pp. 1734–1743.
- [13] F. Viani, L. Lizzi, P. Rocca, M. Benedetti, M. Donelli, and A. Massa, "Object tracking through rssi measurements in wireless sensor networks," *Electronics Letters*, vol. 44, no. 10, pp. 653–654, 2008.
- [14] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000, pp. 32–43.
- [15] L. Gui, X. Zhang, Q. Ding, F. Shu, and A. Wei, "Reference anchor selection and global optimized solution for dv-hop localization in wireless sensor networks," *Wireless Personal Communications*, vol. 96, no. 4, pp. 5995–6005, 2017.

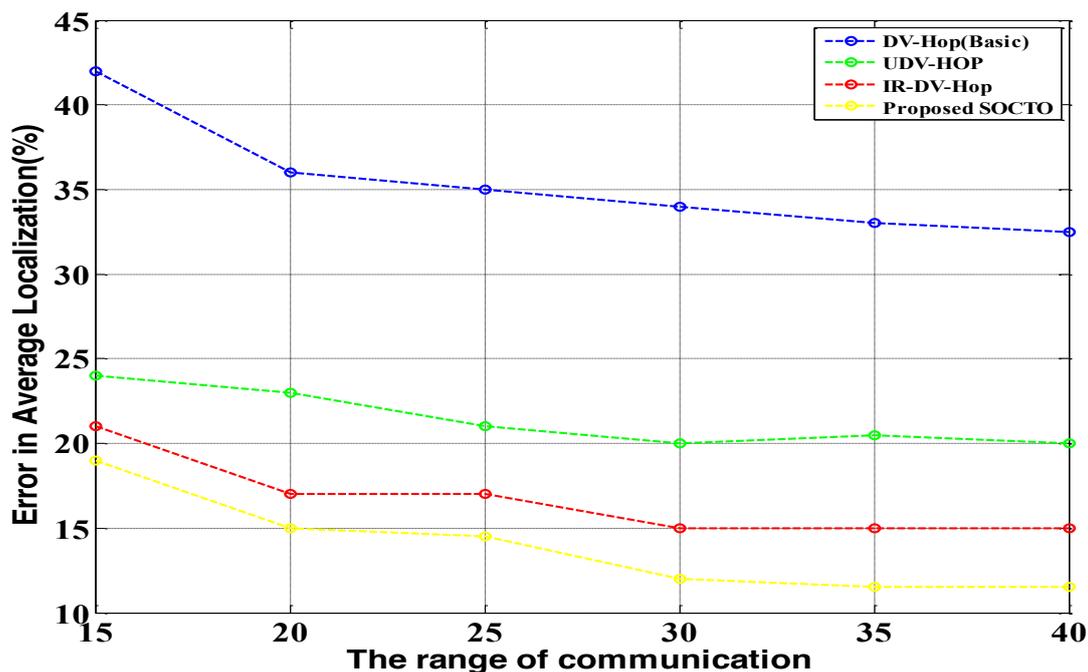


Fig. 11: The range of communication versus the error in localization.

- [16] A. Kaur, P. Kumar, and G. P. Gupta, "A novel dv-hop algorithm based on gauss-newton method," in *2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*. IEEE, 2016, pp. 625–629.
- [17] Q. Tao and L. Zhang, "Enhancement of dv-hop by weighted hop distance. in 2016 ieee advanced information management, communicates, electronic and automation control conference (imcec),(pp. 1577–1580)," 2016.
- [18] L. Cui, C. Xu, G. Li, Z. Ming, Y. Feng, and N. Lu, "A high accurate localization algorithm with dv-hop and differential evolution for wireless sensor network," *Applied Soft Computing*, vol. 68, pp. 39–52, 2018.
- [19] M. Mehrabi, H. Taheri, and P. Taghdiri, "An improved dv-hop localization algorithm based on evolutionary algorithms," *Telecommunication Systems*, vol. 64, no. 4, pp. 639–647, 2017.
- [20] G. Sharma and A. Kumar, "Improved dv-hop localization algorithm using teaching learning based optimization for wireless sensor networks," *Telecommunication Systems*, vol. 67, no. 2, pp. 163–178, 2018.
- [21] J. Cota-Ruiz, P. Rivas-Perea, E. Sifuentes, and R. Gonzalez-Landaeta, "A recursive shortest path routing algorithm with application for wireless sensor network localization," *IEEE Sensors Journal*, vol. 16, no. 11, pp. 4631–4637, 2016.
- [22] S. Zhang, M. J. Er, B. Zhang, and Y. Naderahmadian, "A novel heuristic algorithm for node localization in anisotropic wireless sensor networks with holes," *Signal Processing*, vol. 138, pp. 27–34, 2017.
- [23] P. Kirci and H. Chaouchi, "Recursive and ad hoc routing based localization in wireless sensor networks," *Computer Standards & Interfaces*, vol. 44, pp. 258–263, 2016.

-
- [24] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE wireless communications*, vol. 11, no. 6, pp. 6–28, 2004.
- [25] H. A. de Oliveira, E. F. Nakamura, A. A. F. Loureiro, and A. Boukerche, "Directed position estimation: A recursive localization approach for wireless sensor networks," in *Proceedings. 14th International Conference on Computer Communications and Networks, 2005. ICCCN 2005.* IEEE, 2005, pp. 557–562.
- [26] S. Kumar and D. Lobiyal, "Improvement over dv-hop localization algorithm for wireless sensor networks," *International Journal of Electronics and Communication Engineering*, vol. 7, no. 4, pp. 390–400, 2013.
- [27] S. Messous, H. Liouane, and N. Liouane, "Improvement of dv-hop localization algorithm for randomly deployed wireless sensor networks," *Telecommunication Systems*, vol. 73, no. 1, pp. 75–86, 2020.
- [28] V. Kanwar and A. Kumar, "Dv-hop-based range-free localization algorithm for wireless sensor network using runner-root optimization," *The Journal of Supercomputing*, pp. 1–18, 2020.
- [29] S. Kumar and D. Lobiyal, "An advanced dv-hop localization algorithm for wireless sensor networks," *Wireless personal communications*, vol. 71, no. 2, pp. 1365–1385, 2013.
- [30] V. Kanwar and A. Kumar, "Multiobjective optimization-based dv-hop localization using nsga-ii algorithm for wireless sensor networks," *International Journal of Communication Systems*, vol. 33, no. 11, p. e4431, 2020.
- [31] T. K. Mohanta and D. K. Das, "Class topper optimization based improved localization algorithm in wireless sensor network," *Wireless Personal Communications*, pp. 1–20, 2021.
- [32] S. Dhargupta, M. Ghosh, S. Mirjalili, and R. Sarkar, "Selective opposition based grey wolf optimization," *Expert Systems with Applications*, vol. 151, p. 113389, 2020.
- [33] J. Vesterstrom and R. Thomsen, "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)*, vol. 2. IEEE, 2004, pp. 1980–1987.
- [34] D. Niculescu and B. Nath, "Dv based positioning in ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1-4, pp. 267–280, 2003.