JOURNAL OF ENGINEERING RESEARCH AND TECHNOLOGY, VOLUME 13, ISSUE 1, MARCH 2026

Received on (29-04-2025) Accepted on (03-01-2026)

CGDV-Hop: An Optimized DV-Hop Algorithm with Anchor Selection for Enhancing Sensor Node Localization in Wireless Sensor Networks

Amina Y. AlSallut, Ruba A. Salamah, Maali A. ElHaj, Jehad M. Aldahdooh., and Aiman A. AbuSamra

https://doi.org/10.33976/JERT.13.1/2026/1

Abstract— In Wireless Sensor Networks (WSNs), many sensor nodes are deployed to gather information about their surroundings. The accuracy of this information is dependent on determining the exact location where it was collected. Localization is widely employed in WSNs to find the coordinates of target nodes using anchor nodes as reference points. In this paper, we propose the Closeness Greedy DV-Hop (CGDV-Hop) algorithm, which combines three different techniques, namely, Closeness Centrality for selecting the optimal anchors, Greedy Best First Search for optimizing the Hop count between unknown nodes and anchors, and a DV-Hop for determining the sensor node locations. Simulation results demonstrate the superiority of the CGDV-Hop compared to traditional DV-Hop and its current improvements primarily in terms of Average Localization Error (ALE) and Localization Error Reduction Ratio (LERR), while maintaining a favorable accuracy-complexity tradeoff.

Index Terms— Wireless Sensor Networks, WSN, Localization, DV-Hop, Anchor selection, Optimization.

I INTRODUCTION

WSNs utilize large numbers of sensor nodes with either static or dynamic mobility. The localization of a wireless node is crucial for a wide range of applications, such as vehicle tracking and military surveillance. Traditional localization techniques in WSNs are generally divided into two main categories: range-based and range-free, based on the type of measurement data used [1]. A range-based localization algorithm needs additional power and time consumption, which contradicts the limited resources of sensor nodes. Due to these limitations, increasing attention is being paid to Range-free algorithms.

One of the most widely utilized range-free algorithms is DV-Hop [2]. The simplicity, scalability and the less need for hardware resources encouraged the algorithm to be used more commonly and decreased the cost of the localization algorithm. The fundamental idea behind DV-Hop is to multiply the average hop distance in WSNs by the number of hops that each anchor node has in order to determine the distances between known and unknown nodes [3]. After that, multilateration, triangulation, and trilateration are used to retrieve the position data of unknown nodes [4]. Several efforts have been made to improve the performance of DV-HOP algorithms and reduce the localization error. Some DV-HOP variants can be found in the literature review section.

As DV-Hop is based on the distance and number of hops between nodes, it has been demonstrated that it faces challenges to locate nodes with a high accuracy, compared to the range-based algorithms. Also, as the number of hops between the unknown node and the anchor node increases, DV-HOP is prone to increase the localization error [5]. To overcome these shortcomings, many improved algorithms based on DV-Hop were proposed by researchers [6-15]. They enhanced the accuracy with the cost of increasing computation and hence power consumption, especially for the algorithms that use meta-heuristic optimization techniques as a subsequent step in the localization process. As a result, there is a significant research need in developing a localization method that simultaneously increases accuracy and keeps computational complexity low. In order to address this gap, this study proposes CGDV-Hop, which improves localization accuracy by optimizing hop-count and intelligently selecting anchors without adding costly computational optimization steps. CGDV-Hop is supposed to decrease the complexity and hence, the power consumption at the source-limited sensor nodes.

In the first phase of CGDV-Hop, anchor nodes' locations are determined such that the nearest anchors to the majority of nodes (best anchors) are selected and used in the subsequent steps. In the second phase, an optimization AI algorithm (Greedy Best-First algorithm) will be used in order to find the best path between the nodes and the anchors to optimize the average number of hops. In the optimization step, the prior knowledge from the first phase of selecting anchors can be used to speed up the process of finding the shortest path. This is due to the use of the number of hops obtained from the best located anchors instead of the randomly located ones.

In the third phase, the process of calculating locations proceeds as usual in the traditional DV-Hop algorithm. The use of the optimal number of hops in this phase will increase the accuracy and decrease the localization error.

The main contributions of this research are as follows:

- Integrating three techniques to localize a wireless sensor node, explained as follows:
  - Anchors' Preselection using Closeness Centrality Algorithm to select the best located anchors.
  - Intelligent Best path algorithm (Greedy Best-First Search) to find the best (shortest) path between each unknown node and anchors to calculate the optimal hop count.
  - DV-Hop algorithm, with the use of the optimal hop count obtained from the previous step, to locate the sensor node.
- Performance comparisons measusing accuracy (*ALE*, *LERR*) and qualitative complexity among traditional DV-Hop, DV-Hop with Anchors' preselection, recent optimaization-based DV-Hop variants and CGDV-Hop algorithms are conducted.

The remainder of this paper is organized as follows: Section II provides the related work and section III presents background study. Section IV details the methodology, the proposed CGDV-Hop algorithm. the simulation environment, and performance metrics. Section V presents a discussion of the results. Finally, Section VI concludes the paper and discusses future research directions.

II RELATED WORK

The applicability of the traditional DV-Hop is restricted to certain settings despite having the advantages of being inexpensive, simple to use, and very practicable. Researchers have proposed a variety of improved DV-Hop algorithms to increase the node localization accuracy.

There are a number of applications that have improved the algorithm by modifying one or more of its steps. The first step of most DV-HOP algorithm variants is similar to the first phase of the traditional one. In this phase, all unknown nodes receive the minimum hop counts from each anchor [5]. An Improved DV-Hop Localization Algorithm (IDVLA) has been proposed in [6]. In this algorithm, instead of using the average hop distance for each anchor node individually, the average hop count received from all of the anchors is used. To determine the location of each unknown node, the authors used a 2-D Hyperbolic algorithm rather than using the least square method as in the traditional DV-HOP.

Some other studies focused on enhancing the computational cost of DV-Hop. [7] introduced Weighted Centroid Localization Algorithm (WCL) that utilizes a weight assigned to each anchor node. It is inversely proportional to the distance between the node and the anchor. The weighted approach reduces computational complexity while maintaining accuracy that is comparable to that of the DV-HOP. An Improved Weighted Centroid Algorithm is proposed in [8]. They selected a subset of anchor nodes whose hop counts are the smallest among all other anchors. Also, an Improved DV-Hop Localization (IDV-Hop) algorithm is conducted by [9] adding a correction step to the process of computing the average hop counts. This step slightly increased the computational complexity. Despite the complexity improvement of these methods, the localization accuracy remains relatively modest.

On the other hand, some studies have utilized optimization techniques in order to optimize the solutions of either the final step of DV-Hop, where the location of a wireless node is estimated or the first and the second steps, where the anchors are selected and the number of hops is propagated. [10] used a genetic algorithm as one of the meta-heuristic algorithms following the DV-HOP position calculation process. This algorithm is used to optimize the calculated location of an unknown node and minimize the localization error. Unfortunately, it has the drawback of increased computational complexity due to the use of genetic algorithms. In [11], a strategy which selects the optimal static anchor node's locations is proposed. This strategy computes the distance between anchor nodes and all the network nodes and then uses the k nearest anchors to the majority of nodes. This method is applicable to stationary anchors and sensor nodes.

In [12], the study aimed at enhancing the standard DV-positioning capabilities so that it could more precisely pinpoint where the sensors were located in the detecting region. The weighted minimum means square error (MMSE) [13] criterion was used to replace the unbiased estimation approach while determining the ideal anchor nodes iteratively. The proposed DV-Hop provides higher positioning performance, according to simulation findings. A multi-objective optimization-based enhancement of the DV-Hop localization algorithm using a Differential Evolution with Multiple Neighborhoods (DEMN) strategy is proposed by [14]. By integrating DEMN into the position estimation stage of DV-Hop, the proposed approach refines the estimated coordinates of unknown nodes through iterative optimization. Simulation results demonstrate improved accuracy compared to traditional DV-Hop, at the expense of increased computational complexity due to the use of evolutionary optimization.

Recent works have concentrated on improving DV-Hop localization precision by incorporating meta-heuristic optimization methods and advanced distance calculation methodologies. Yıldız [15] introduced a parameter-free Fire Optimizer to enhance average hop distance estimation, resulting in scalable and precise localization without the need for explicit tuning of parameters. Kaur et al. [16] presented PSLDV-Hop, which integrates particle swarm optimization with a refinement technique to reduce cumulative hop error rates. Hadir et al. [17] utilized PSO to enhance distance correction in DV-Hop for IoT and WSN environments, resulting in increased localization accuracy though experiencing additional computation complexity.

Other studies have investigated different bio-inspired optimizers, like the Remora Optimization Algorithm used in DVHOP-ROA by Rabhi et al. [18] and the VVC-HCO technique presented by Ali et al. [19], both of which concentrate on post-estimation optimization to lower localization error. Furthermore, Wang et al. [20] suggested a multi-hop localization technique that incorporates hierarchical anchor extension and path tortuosity correction to increase the reliability of distance estimations.

Also, [21], [22] proposed a strategy where each anchor node selects an optimal subset of other anchors to improve accuracy. This subset is then used to recalculate the average hop size, which, along with the subset information, is broadcasted to nearby unknown nodes. This may overwhelm the network and increase the burden on the anchor nodes. Applying human conception optimization to the DV-Hop algorithm with time-varying acceleration coefficients is another noteworthy enhancement [23]. By adding a parameter to modify the hop sizes of anchor nodes, this technique improves position estimations without requiring extra hardware.

The effect of the number of anchor nodes and the communication radius has been studied in many researches. In [24], the impact of the communication radius and the number of anchor nodes on positioning precision is analyzed. The results of the simulations demonstrated clearly that the inaccuracies of location and the average positioning error are minimized as the number of anchor nodes and communication radius increase. [25] developed a DV-Hop based on Average Hop Distance and Estimated Coordinates. This method was dependent on two factors: location error with varying communication radius and number of anchor nodes. The simulation of this method showed that the performance of the improved DV-Hop algorithm in WSNs is better compared with the traditional algorithm.

Although these methods greatly improve the accuracy of localization, they often depend on iterative optimization procedures which increase processing overhead. On the other hand, by optimizing anchor selection and hop-count estimation prior to position calculation, the proposed CGDV-Hop approach enhances localization efficiency while avoiding complex meta-heuristic optimization yet preserving competitive accuracy.

III Background

The theoretical background of this research depends on three essential algorithms: Closeness Centrality Algorithm, Greedy Best-First search algorithm and DV-Hop algorithm. The following subsections present an overview of these algorithms.

A Closeness Centrality Algorithm

Finding the location of the anchors through manual deployment or a GPS device is the fundamental idea behind the anchor node localization approach. The goal of selecting the anchor node location is to choose the location of the anchor more effectively. The locations of the anchor nodes should be optimal in the sense of being distributed over the network area in order to increase the node's localization coverage and accuracy [11].

By measuring the average distance between each node and every other node in the network, the closeness centrality technique finds the key nodes. The concept behind it is that the node that is closest to the network's center must also be the closest to the other nodes. The closer a node is to other nodes, the higher its proximity centrality value.

The average shortest distance between a node v_i and all other nodes in a network of n connected nodes is determined by equation (1).

$$d_i = \frac{1}{n-1}\sum_{i\neq j}^{n} d_{ij} \qquad (1)$$

The proximity centrality of the node v_i, or the reciprocal of d_i, is defined as shown in (2):

$$CC(i) = \frac{1}{d_i} = \frac{n-1}{\sum_{i\neq j} d_{ij}} \qquad (2)$$

The resulting closeness centrality values are sorted in descending order, with the first anchor node being the node with the highest closeness centrality value. Next, the sorted closeness centrality values are iterated through at regular intervals. The quick sort technique [26] is utilized to determine the optimal interval, and the chosen node is then used as an anchor node.

In order to determine the locations of the remaining nodes in the topology, the localization approach (e.g. DV-Hop) is chosen in the final step.

B DV-Hop

The procedure of the traditional DV-Hop algorithm is illustrated in Figure 1 and summarized in the following steps:

Step 1. Minimal hop count collection:

The coordinate information and the hop count, which is initially set to *0* and incremented by *1* after each hop, are broadcasted by each anchor node. Each node updates and stores the minimal value if the obtained value is smaller than the previous one. At the end of this step, all nodes will have the smallest hop counts.

Step 2. Average hop distance Determination:

In step 1, the minimal hop value and anchor coordinates are obtained between any two anchor nodes. In the second step, the average hop distance denoted as H_i is calculated by (3):

$$H_i = \frac{\sum_{i\neq j}\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}}{\sum_{i\neq j} h_{ij}} \qquad (3)$$

Where h_{ij} is the minimum hop count between the i^{th} and j^{th} anchor nodes. All the network's nodes will then get a broadcast of all the average hop distances.

Step 3. Unknown nodes coordinates Determination:

Each unknown node k calculates the distance d_{ik} between itself and the i^{th} anchor node by multiplying the average hop distance H_i by the minimal hop count h_{ik}:

$$d_{ik} = H_i \times h_{ik} \qquad (4)$$

After obtaining three or more anchor distance data, the least square method could be used to estimate the coordinates of the unknown node. The distances between the anchor nodes (x_i , y_i) and the unknown node (x , y) are as follows:

$$(x_i - x)^2 + (y_i - y)^2 = d_i^2 \qquad (5)$$

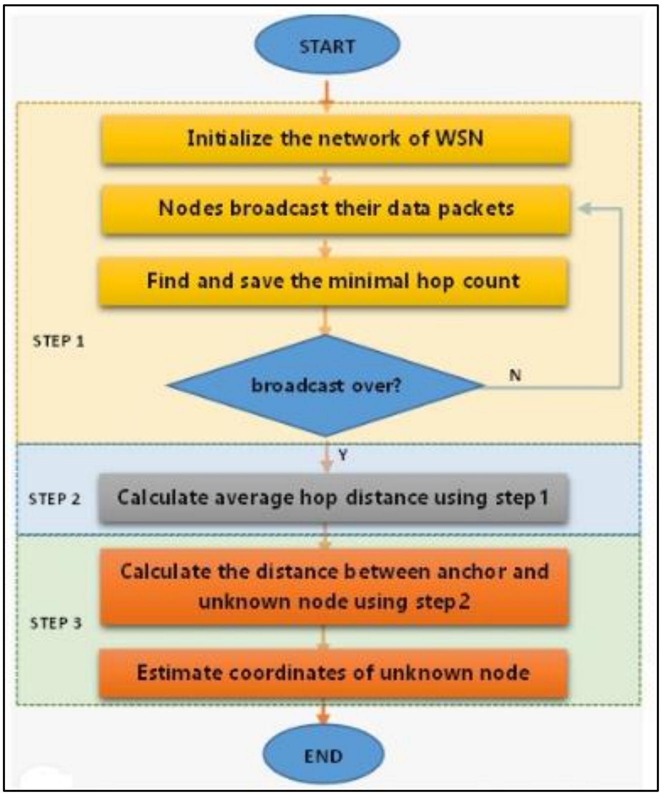

Figure1. Procedure of DV-Hop Localization Algorithm [27]

C Greedy Best-First search algorithm (GBFS)

In an attempt to reach the search goal, the path that seems best at the time is always chosen by the greedy best-first search algorithm. It combines methods for depth-first search and breadth-first search [28]. It employs search and heuristic functions. At each stage, the most promising node is selected with the aid of best-first search. This node, which is closest to the goal node, is expanded and the least cost is estimated by a heuristic function which is the estimated cost from node n to the goal. By using the priority queue, the greedy best-first algorithm is implemented [29].

Instead of identifying a certain and ideal solution, a greedy-based search will attempt to find the necessary search path as quickly as possible. If GBFS discovers the answer, it halts the search at the node that provided the answer. If not, it will figure out the heuristic value and extend the child node with the lowest heuristic value. In other words, GBFS assumes that states with lower heuristic values are located on the least expensive path to the closest goal state. Heuristic value is therefore calculated for each step and then compared to all the states created up to that point (but not expanded yet).

This compromise ensures that a potential search path is rapidly recognized, which can occasionally be very helpful. Also, this algorithm has a low complexity and may be the least among the algorithms that use prior knowledge, such as A star search algorithm [30].

IV Methodology

In the following subsections, detailed description of the proposed algorithm is introduced.

A Proposed Algorithm

The proposed localization algorithm integrates three different algorithms, each of which has a specific goal. According to the use of these algorithms, CGDV-Hop is divided into three phases as described below.

Phase I: Anchors Selection

Instead of randomly selecting a number of anchors as known located nodes, the closeness centrality algorithm is used to select the most centralized nodes, i.e., the nodes which tend to be the nearest to a group of unknown nodes. The algorithm calculates the average distance between each anchor and all the nodes in the network using (1). Then, the distances are sorted in ascending order. After that, a number of anchors are selected that are spaced by a certain interval, which is calculated by the following formula:

$$interval = ceiling(Nodes\ No./Anchors\ No.) \qquad (6)$$

This selection technique increases the coverage range of the anchors, and hence, a larger number of nodes receives both locations and number of hops broadcasted by the anchors. Moreover, the accuracy of the localization algorithm increases.

Phase II: Hop Count Optimization

In this phase, the Greedy Best First Search algorithm is used to obtain the optimal path, and hence the optimal number of hops between an unknown node and an anchor node. Instead of beginning the search with random initializations, the received hop counts from the selected anchors at phase 1 are used as prior knowledge and heuristic information to the Greedy Best First Search algorithm. This prior knowledge greatly speeds up the search process.

More specifically, the best path between each node and each anchor is found using the received hop counts as heuristics. Then, the best (the nearest optimal) three anchors for each unknown node can be determined and the average hop count is calculated using only these three anchors. The complexity of the algorithm is considerably reduced by using only three anchors instead of all anchors. The resultant optimized hop count, which is used in the DV-Hop phase, increases the accuracy of the localization algorithm.

Phase III: Localization using DV-Hop

In this phase, the optimal hop count obtained from the second phase is used as an input to the DV-Hop localization algorithm. This compensates for the need for an optimization process after the localization estimations are completed. Hence, the complexity of the algorithm is decreased compared with the algorithms that supplement the process of estimating locations with a process of optimization.

The optimal minimum hop count is used directly in step 3 of the DV-Hop algorithm described earlier into equations 4 and 5 to estimate the location of nodes. The flowchart of the CGDV-Hop algorithm is illustrated in Figure 2.

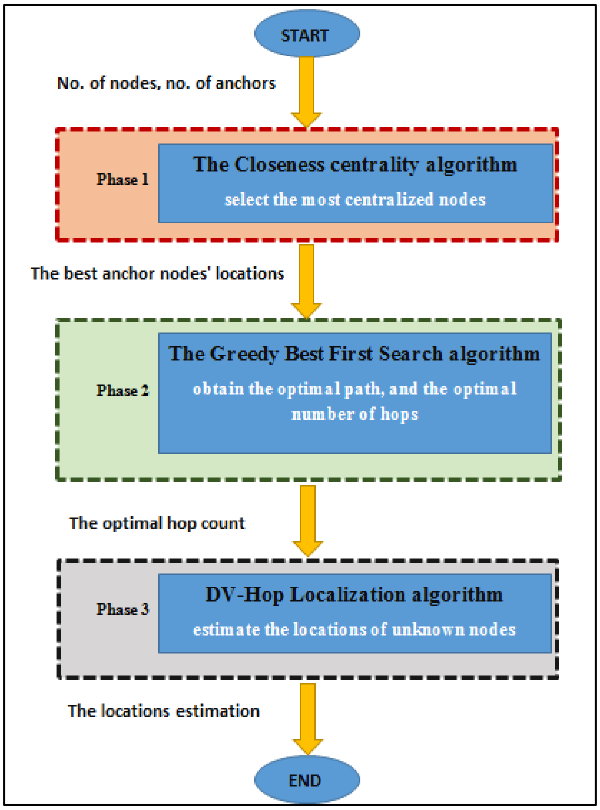

Figure2. Phases of proposed CGDV-Hop Localization Algorithm

B Simulation Environment

The CGDV-Hop algorithm was implemented in Matlab R2024b. The simulation environment parameters are shown in Table 1. To make the comparisons fair, the same parameters and the same environments have been simulated for the traditional DV-Hop, DV-Hop with preselected Anchors and the CGDV-Hop algorithms.

TABLE 1

Simulation Parameters

Parameter	Value
Network Size	100m * 100m
Total Number of Nodes	100, 200, 300
Anchors Density	10%, 20%, 30%
Node Deployment	Random
Communication Radius	50

All the source codes and results of the algorithms that have been compared were regenerated. The simulations have been repeated for different combinations of number of nodes and anchor nodes density.

C Performance Metrics

To evaluate the localization efficiency of the proposed CGDV-Hop algorithm and the reference DV-Hop-based approaches, the following performance metrics are adopted:

- **Average Localization Error (*ALE*):**

The primary accuracy metric which is calculated by first determining the Localization Error (*LE*) for each node. *LE* is defined as the Euclidean distance between a single unknown sensor node's estimated and actual positions and is calculated as follows:

$$LE_i = \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \tag{7}$$

Where, (x_i, y_i) denotes the actual coordinates of the i^{th} unknown node and $(\hat{x}_i, \hat{y}_i)$ represents the estimated coordinates obtained from the localization algorithm.
The *ALE* is then derived as the arithmetic mean of localization errors across all unknown nodes:

$$ALE = \frac{1}{N_u} \sum_{i=1}^{N_u} LE_i \tag{8}$$

Where, N_u is the number of unknown nodes in the network. Anchor nodes are excluded from this calculation since their positions are known in advance.

- **Localization Error Reduction Ratio (*LERR*):**

The localization error reduction ratio, which quantifies an algorithm's relative accuracy increase over the conventional DV-Hop approach, is defined as follows:

$$LERR = \frac{ALE_{DV-Hop} - ALE_{Method}}{ALE_{Method}} x100\% \tag{9}$$

Compared to the baseline DV-Hop algorithm, this metric represents the percentage reduction in average localization error attained by a particular approach.

- **Computational Complexity:**

In addition to metrics for accuracy (*ALE*, *LERR*), the algorithm's complexity is evaluated as well. Since execution time depends on hardware and implementation, the comparison is performed qualitatively based on the processing cost, iteration needs, and search methods (such as meta-heuristic versus graph-theoretic approaches). Accordingly, bsed on the beforementiond metrics, the complexity of the algorithms are categorized as:

- Low complexity: non-iterative DV-Hop based methods.
- Moderate complexity: enhanced DV-Hop with preprocessing or filtering.
- High complexity: optimization-based or meta-heuristic localization methods.

Also, a theoretical Big-O notation analysis is provided as a complement for qualitative analysis.

V Results and Analysis

The nodes' deployment with 100 total nodes and 10%, 20% and 30% anchor ratios are shown in Figures 3, 4 and 5, respectively. All nodes were deployed randomly, and then the anchors were selected based on the closeness centrality algorithm. Anchors' selection improves the distribution of them so that each mediates a different group of nodes and thus improves the overall coverage.

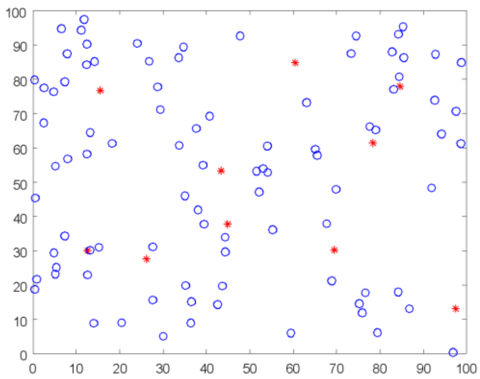

Figure 3. Nodes Deployment with 10% Anchors

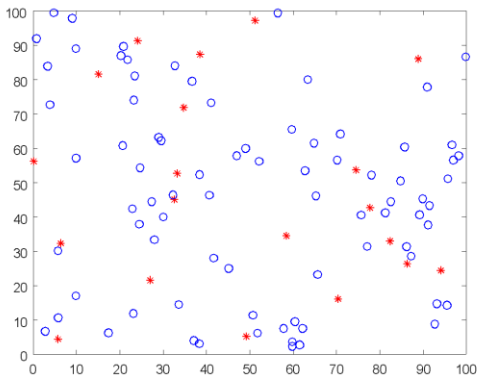

Figure 4. Nodes Deployment with 20% Anchors

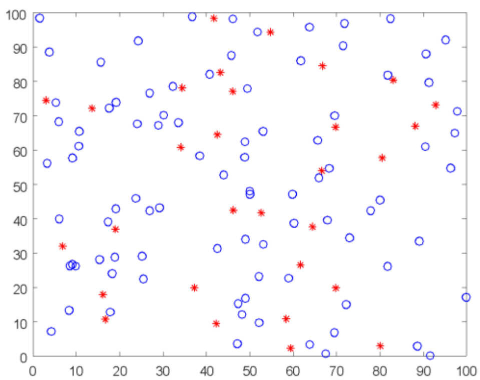

Figure 5. Nodes Deployment and 30% Anchors

The Deployments with 200 and 300 total nodes with varied anchor ratios were also generated randomly.

Results of the *ALE* for the three different algorithms varying the total number of nodes with 10%, 20% and 30% anchor ratios are shown in Table 2 and are illustrated in figures 6, 7 and 8, respectively. The average error decreases as the number of anchors increases. These results are logical as the larger number of anchors increases the probability of receiving hop counts from anchors at the same node, and hence the accuracy improves.

On the other hand, when compared to the other two approaches, CGDV-Hop consistently delivers the lowest *ALE* for all anchor densities. The improvement is especially obvious at 10% anchor density, where CGDV-Hop significantly lowers the localization error at all node densities. This pattern suggests that the suggested hop-distance correction and anchor optimization technique works particularly well in sparse anchor situations.

TABLE 2

Average Localization Error (ALE) Results

Anchor Ratio	Number of Nodes	DV-Hop	Preselection DV-Hop	CGDV-Hop
10%	**100**	22.54	20.10	**17.20**
	200	19.90	17.90	**14.90**
	300	18.90	17.20	**13.80**
20%	**100**	20.40	19.30	**15.57**
	200	18.30	16.90	**13.90**
	300	17.20	16.00	**12.90**
30%	**100**	18.86	17.67	**12.20**
	200	16.90	15.80	**11.10**
	300	15.90	14.90	**10.50**

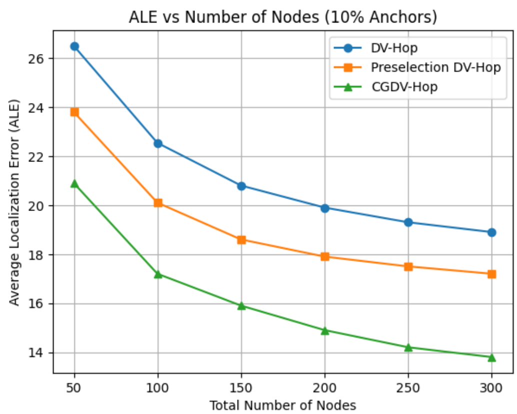

Figure 6. Average Localization Error with 10% Anchors

For all methods, a general decrease in *ALE* is shown as the number of nodes rises from 100 to 300. Higher node density enhances network connectivity and hop-count reliability, which results in more precise distance estimation, hence this trend is expected.

Nevertheless, CGDV-Hop maintains a distinct performance advantage over the conventional and preselection-based DV-Hop methods, exhibiting superior scalability despite this overall improvement.

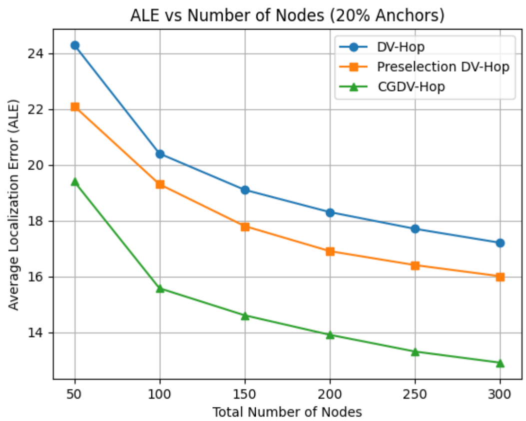

Figure 7. Average Localization Error with 20% Anchors

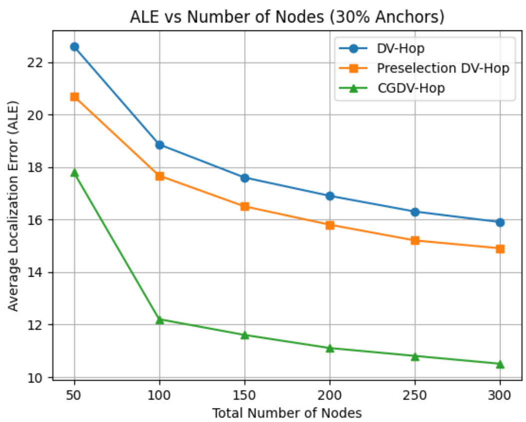

Figure 8. Average Localization Error with 30% Anchors

Table 3 displays each method's localization error reduction ratio (*LERR*) in comparison to the conventional DV-Hop algorithm in order to further quantify the relative performance gain. The suggested CGDV-Hop outperforms the Preselection DV-Hop approach, which shows a slight improvement, with an average *LERR* of roughly 27.56% across various anchor densities.

Also, CGDV-Hop shows competitive localization error reduction when compared to current optimization-based DV-Hop versions documented in the literature. The complexity comparison in Table 3 is qualitative according to the definition provided in Performance Metrics subsection. the results indicate that CGDV-Hop achieves a favorable balance between accuracy enhancement and computational overhead compared to alternative optimization-based approaches.

TABLE 3

LERR and Complexity Comparison

Algorithm	**LERR (%)**	**Computational Complexity**
DV-Hop	0	Low
Preselection DV-Hop	7.51	Low–Moderate
DEMN + Multi-objective DV-Hop [14]	~16 (Random Deployment)	High (Iterative Optimization)
Fire Hawk Optimizer DV-Hop [15]	~25 (Random Deployment)	High (about 50 iterations are needed)
CGDV-Hop	**27.56**	**Moderate**

- **Error Reduction Analysis**

The difference between the actual Euclidean distance d_{ik} and the estimated distance $\hat{d}_{ik} = H_i \times h_{ik}$ is the primary cause of the localization error in DV-Hop. In Phase I, we select anchors that minimize the average hop count $\bar{h}$ to all nodes by using Closeness Centrality. Reducing the maximum number of hops needed to reach an anchor naturally limits the propagation of distance estimation errors since the cumulative error in distance estimation is proportional to the hop count h_{ik}.

In addition, Phase II minimizes the "path tortuosity" error that is common in standard DV-Hop, which selects any shortest path—even one that is not straight—by using GBFS to determine which path closely matches the Euclidean straight line.

- **Complexity Analysis**

In terms of computational complexity, the traditional DV-Hop algorithm exhibits low complexity since it relies primarily on hop-count propagation and least-squares position estimation without additional refinement steps. A limited anchor selection stage based on hop-count information is introduced by the Preselection DV-Hop, which is non-iterative but has a small computational overhead increase.

The proposed CGDV-Hop eliminates population-based search procedures and iterative meta-heuristic optimization while incorporating structured anchor grouping and hop-distance correction techniques, which add a little computational cost. As a result, CGDV-Hop achieves significant gains in localization accuracy while maintaining a moderate processing complexity. This makes CGDV-Hop particularly suitable for wireless sensor networks with limited resources.

On the other hand, optimization-based DV-Hop variations that have been documented in the literature usually use swarm-based or iterative evolutionary methods, which result in much greater processing requirements and computational complexity. The accuracy improvement at the expense of increased cost was explicitly mentioned in the methods compared in the table above.

For the Big-O notation analysis, the three phases of

CGDV-Hop can be used to formally examine its computational complexity. For example, let N represent all nodes, A represent anchors, and E represent communication links.

- **Phase I** (Anchor Selection): Calculating closeness centrality requires computing all-pairs shortest paths. For an unweighted graph, this takes $O(N(N+E))$.
- **Phase II** (Hop Optimization): The Greedy Best-First Search (GBFS) for U unknown nodes to find optimal paths to A anchors has a complexity of $O(U.A.log\ N)$ in the average case when using a priority queue for heuristic search.
- **Phase III** (Localization): The final DV-Hop steps (average hop distance calculation and least squares estimation) have a complexity of $O(A^2 + A.U)$. Thus, the total complexity is dominated by the initial centrality calculation, $O(N^2 + N.E)$, which is significantly lower than the $O(I.P.N)$ complexity of meta-heuristic variants (where I is iterations and P is population size).

VI Conclusions and Future Work

In this paper, a hybrid algorithm (CGDV-Hop) is proposed to overcome the high average localization error of the traditional DV-Hop localization algorithm and the high complexity of its variants that use machine learning algorithms as an optimization later phase. This work integrates three algorithms as successive phases.

In phase 1, the Closeness Centrality algorithm is used to select the best located anchors aiming to choose the median of each group of nodes, and hence increasing the coverage. Phase 2 uses the Greedy Best First Search algorithm to find the optimal path and thus the optimal number of hops between an unknown node and each anchor. This optimized Hop Count is then used as an input to the next phase. The last phase uses the DV-Hop algorithm to estimate locations with the aid of the optimal Hop Count obtained from the previous phase.

Experimental results demonstrated that CGDV-Hop outperforms both the traditional DV-Hop and the DV-Hop with anchors Preselection using Closeness Centrality algorithms in terms of Average Localization Error. Moreover, CGDV-Hop gains the least Error Reduction Ratio compared to the current optimization-based DV-Hop variants. In contrast to existing DV-Hop improvements that depend on cost computational optimization or static anchor configurations, CGDV-Hop combines greedy and lightweight graph-theoretic search approaches to optimize hop counts prior to localization. This proactive optimization achieves a favorable accuracy–complexity tradeoff, distinguishing CGDV-Hop from recent DV-Hop variants.

- **Mobile-WSN Applicability**

In mobile wireless sensor networks, node mobility causes frequent topology variations that affect hop-count stability and anchor-distance estimation. The proposed CGDV-Hop can be extended by periodically recomputing anchor centrality and updating hop paths using incremental graph updates instead of full recomputation. The update interval T can be adaptively selected according to node velocity v, where higher mobility requires shorter update periods. Furthermore, distance estimate between consecutive topology updates can be stabilized by including Kalman-filter-based smoothing or mobility prediction. In dynamic environments, this allows CGDV-Hop to control computational cost while maintaining localization accuracy.

References

[1] D. Han, Y. Yu, K. C. Li, and R. F. de Mello, “Enhancing the sensor node localization algorithm based on improved DV-Hop and DE algorithms in wireless sensor networks,” Sensors, vol. 20, no. 3, 2020.

[2] Fengrong, Izzeldin, Xinni, Kamarul, and Hao, “A survey on DV-Hop localization techniques in three-dimensional wireless sensor networks,” Int. J. Online Biomed. Eng. (iJOE), vol. 16, no. 11, 2020.

[3] M. Mehrabi, H. Taheri, and P. Taghdiri, “An improved DV-hop localization algorithm based on evolutionary algorithms,” Telecommun. Syst., vol. 64, no. 4, pp. 639-647, 2017.

[4] S. Kumar and D. K. Lobiyal, “An advanced DV-hop localization algorithm for wireless sensor networks,” Wireless Pers. Commun., vol. 71, no. 2, pp. 1365-1385, 2013.

[5] G. Farjamnia, Y. Gasimov, C. Kazimov, and M. Hashemi, “A survey of DV-Hop localization methods in wireless sensor networks,” J. Commun. Eng., pp. 359-398, 2021.

[6] H. Chen, K. Sezaki, P. Deng, and H. C. So, “An improved DV-hop localization algorithm for wireless sensor networks,” Proc. IEEE Conf. Industrial Electronics and Applications, Singapore, 2008, pp. 1557-1561.

[7] B. Zhang, M. Ji, and L. Shan, “A weighted centroid localization algorithm based on DV-hop for wireless sensor network,” Proc. Int. Conf. Wireless Communications, Networking and Mobile Computing, Shanghai, China, 2012.

[8] G. Song and D. Tam, “Two novel DV-Hop localization algorithms for randomly deployed wireless sensor networks,” Int. J. Distrib. Sens. Netw., vol. 11, no. 2, 2015.

[9] W. Yu and H. Li, “An improved DV-Hop localization method in wireless sensor networks,” Proc. Comput. Sci. and Automation Engineering, 2012.

[10] B. Peng and L. Li, “An improved localization algorithm based on genetic algorithm in wireless sensor networks,” Cogn. Neurodyn., vol. 9, no. 2, pp. 249-256, 2015.

[11] W. Liu, X. Luo, G. Wei, and H. Liu, “Node localization algorithm for wireless sensor networks based on static anchor node location selection strategy,” Comput. Commun., vol. 192, 2022.

[12] T. Chen, S. Hou, L. Sun, and K. Sun, “An enhanced DV-Hop localization scheme based on weighted iteration and optimal beacon set,” Electronics, vol. 11, no. 14, 2022.

[13] X. Huang, D. Han, M. Cui, G. Lin, and X. Yin, “Three-dimensional localization algorithm based on improved A and DV-Hop algorithms in wireless sensor network,” Sensors, vol. 21, no. 13, 2021.

[14] P. Wang et al., "DV-hop localization based on distance estimation using multinode and hop loss in IoT," IEEE Internet Things J., vol. 11, 2024.

[15] D. Yıldız, "Accurate and scalable DV-Hop-based WSN localization with parameter-free Fire Hawk optimizer," Mathematics, vol. 13, 2025.

[16] B. Kaur et al., "PSLDV-Hop: A robust localization algorithm for WSN using PSO and refinement process," PeerJ Comput. Sci., vol. 11, 2025.

[17] A. Hadir et al., "Optimized DV-Hop localization algorithm using PSO for IoT and WSNs," Procedia Comput. Sci., 2025.

[18] S. Rabhi, R. I. Alkanhel, and D. S. Khafaga, "DVHOP-ROA: An improved DV-Hop range-free localization in wireless sensor networks using remora optimization algorithm," 2025.

[19] M. Ali, B. E. Abu-elmaaty, and S. A. Napoleon, "An improved DV-Hop localization technique via VVC-HCO for wireless sensor networks," Int. J. Telecommun., 2025.

[20] L. Wang, X. Liu, and D. Zou, "A multi-hop localization algorithm based on path tortuosity correction and hierarchical anchor extension for wireless sensor networks," Electronics, vol. 14, 2025.

[21] Y. Cao and J. Xu, "DV-Hop-based localization algorithm using optimum anchor nodes subsets for wireless sensor network," Ad Hoc Netw., vol. 140, 2023.

[22] V. C. S. R. Rayavarapu and A. Mahapatro, "MOANS DV-Hop: An anchor node subset-based localization algorithm for wireless sensor networks," Ad Hoc Netw., vol. 152, 2024.

[23] S. K. Panda, D. Acharya, D. K. Das et al., "An improved DV-Hop localization algorithm based on human conception optimization with time varying acceleration coefficients for wireless sensor network," Wireless Pers. Commun., 2024.

[24] J. Xu et al., "Improvement on the DV-Hop localization algorithm based on average hop distance optimization," J. Phys.: Conf. Ser., 2022.

[25] W. Gao, Y. Sun, W. Li, and W. Wang, "An improved DV-Hop algorithm based on average hop distance and estimated coordinates," Proc. Chinese Control and Decision Conf., 2017.

[26] H. Sajindra, "Quick sort algorithm," Medium, Oct. 2021. [Online]. Available: https://medium.com/@Hirushan-Tech/quick-sort-algorithm-8b465e05501a

[27] J. Chen, W. Zhang, Z. Liu, R. Wang, and S. Zhang, "CWDV-Hop: A hybrid localization algorithm with distance-weight DV-Hop and CSO for wireless sensor networks," IEEE Access, vol. 9, pp. 380-399, 2020.

[28] M. Nwadiugwu, "The depth first search and breadth first search as a graph traversal," 2016. [Online]. Available: https://arxiv.org/abs/1602.04464

[29] C. Frasinaru and M. Raschip, "Greedy best-first search for the optimal-size sorting network problem," Procedia Comput. Sci., vol. 159, pp. 586-595, 2019.

[30] S. Raghavan, "Investigation of heuristic search techniques for path planning," Int. J. Eng. Res. Technol. (IJERT), vol. 11, no. 5, 2022.

Amina Y. AlSallut: Received a Master Degree of Computer Engineering at the Islamic University of Gaza, 2012. Currently, is an enrolled student in Phd Computer Engineering program at the Islamic University of Gaza. Since 2009, she has been a lecturer at the Engineering Department of College of Science at Technology, and has become the head of the Computer Department in January 2015 to July 2016. She has many published researches regarding many fields in Computer Science, Artificial Intillegence, Wireless Sensor Networks, and Machine Learning.

Ruba A. Salamah: Holds a Master Degree of Computer Engineering at the Islamic University of Gaza, 2009. Currently, is an enrolled student in Phd Computer Engineering program at the Islamic University of Gaza. Since 2000, she has been a lecturer at the Computer Engineering Department of Islamic University of Gaza.

Maali A. El-Haj: Obtained her M.Sc. degree in Information Technology in 2018. Since 2022, she has been enrolled in a Ph.D. Computer Engineering program at the Islamic University of Gaza. She has several published research papers in the field of Information Technology and currently works as a lecturer in the same domain.

Jehad M. Aldahdooh: Received his Ph.D. degree in Computer Science from the University of Helsinki, Finland, in 2024. He earned his M.Sc. degree in Computer Science from Eötvös Loránd University, Hungary, and his B.Sc. degree in Computer Engineering from the Islamic University of Gaza. His research interests include machine learning, NLP, biomedical informatics, and data mining.

Aiman A. Abusamra: A Professor in the Computer Engineering Department at the Islamic University of Gaza. He received his Ph.D. degree from the National Technical University of Ukraine in 1996. Dr. AbuSamra was the Assistant Vice President of IT Affairs at IUG between 2011- 2014. He was a supervisor of many Master's theses in computer networks, computer security, and other topics. His research interests include blockchain technology, computer networks and mobile computing. Dr. AbuSamra is a member of the Technical Committee of the International Arab Journal of Information Technology (IAJIT). He was recognized as one of the most active reviewers.