Received November 6, 2018, accepted November 26, 2018, date of publication December 4, 2018,
date of current version December 31, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2884837

Optimized Hop Angle Relativity for DV-Hop Localization in Wireless Sensor Networks

SONGYUT PHOEMPHON[1], CHAKCHAI SO-IN[1], (Senior Member, IEEE), AND NUTTHANON LEELATHAKUL[2]

[1]Applied Network Technology Laboratory, Department of Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen 40002, Thailand
[2]Faculty of Informatics, Burapha University, Chonburi 20131, Thailand

Corresponding author: Chakchai So-In (chakso@kku.ac.th)

This work was supported in part by the Thailand Research Fund through the Royal Golden Jubilee Ph.D. Program under Grant PHD/0214/2558 and in part by Khon Kaen University.

ABSTRACT Smart multifunctional sensors integrated with wireless connectivity (also known as wireless sensor networks or WSNs) play an important role in the Internet of Things (IoT). Several challenges associated with WSNs have been researched and energy consumption represents the main limitation. Another major challenge is localization because a sensor or node should be self-contained and organized and have a low cost of integration. The range-free approach is promising due to its simplicity. Notably, it does not require additional logics and needs only key parameters, such as the number of hops and node locations. Distance-vector-hop-based localization (DV-Hop) is a pioneering range-free approach, and the corresponding localization approximation method does not require areas to be covered by nodes with known positions (also called known nodes or anchor nodes). However, the precision of this approach relies on several factors, including the node density and the method of determining the relation between the distance and the number of hops between two anchor nodes (i.e., hop size). Thus, this research enhances DV-Hop by: 1) reducing the approximation coverage to a specific area, thereby requiring fewer anchor nodes; 2) further decreasing the area using a bounding box; and 3) adopting particle swarm optimization (PSO) by integrating the number of hops and anchor nodes into the fitness function to improve the approximation precision. To evaluate the efficiency of the proposed scheme, the simulation results are compared with those of five recently proposed DV-Hop localization methods: iDV-Hop, DV-maxHop, Selective 3-Anchor DV-Hop, PSODV-Hop, and GA-PSODV-Hop.

INDEX TERMS Distance-vector-hop-based localization, hop angle relativity, localization, particle swarm optimization, wireless sensor networks.

I. INTRODUCTION

Recently, applications of microcontroller chips with integrated wireless transmission logics have become prevalent because they enable rapid access and control over wireless networks. A variety of such devices can coordinately transmit sensory data to cloud infrastructures, thereby providing the so-called big data for data analysts and decision makers. In addition, the ability to control devices that are integrated into the Internet of Things (IoT) enables increased applicability to real-world problems, such as gas pipeline leak detection, environmental monitoring, and elderly patient motion detection [1].

Sensor nodes (SNs), aka, sensors or nodes, are important devices that are deployed for monitoring sensory data, such as temperature, humidity, light, and/or pressure data. SNs can be wirelessly connected to form a large network, namely, a wireless sensor network (WSN) [1]. WSN applications have various limitations, and they face several important issues, e.g., quality of service (QoS) [2], data aggregation, energy-aware computation [3], and real-time communication issues [2], [3]. In practice, SN locations are extremely important; thus, they have attracted the attention of numerous researchers. The global positioning system (GPS) strategy might represent a solution; however, it has several limitations with respect to key WSN design requirements, such as 1) the low accuracy of SN positions, especially at locations where GPS signals are absent; 2) high power consumption; and 3) high costs due to additional hardware logics [4].

The methods for estimating SN locations have been presented in numerous studies [5], and they can be categorized into range-based and range-free techniques based on the required hardware specifications of active devices [6]. Localization strategies in range-based techniques attempt to estimate the distances or angles between SNs, such as the angle of arrival (AoA), received signal strength indicator (RSSI), time of arrival (ToA), and time difference of arrival (TDoA). Therefore, these approaches require high-precision hardware that is costly and consumes considerable energy. Range-free techniques estimate the location of a node by determining the distances between the (unknown) node and other known nodes (i.e., anchor nodes) by counting the minimum number of hops between the nodes, and they require no specific equipment, consume less power, and cost less than range-based methods. Therefore, range-free techniques have gained popularity due to their practicality for WSNs. Examples of such localization techniques are Distance Vector Hop (DV-Hop) [7], Centroid [8]–[11], and Approximate Point In Triangulation (APIT) methods [12].

In this paper, we focus on increasing the efficiency of DV-Hop because DV-Hop yields more accurate SN locations than the centroid or APIT methods, even when the number of anchor nodes provides insufficient coverage for the entire area. Compared with other learning techniques, such as support vector regression (SVR) [13] and kernel ridge regression [14], DV-Hop has the following major advantage: its localization procedure is simple and suitable for WSN deployment.

The accuracy of DV-Hop depends on the node density, the number of anchor nodes, and the network topology [7], all of which directly affect the estimation of the distance between the unknown node and the anchor node. Furthermore, the approximate distance from an anchor node with a low minimum hop count tends to be more accurate than that from an anchor node with a large minimum hop count, which suggests that the accuracy of the estimated distance between the unknown node and the anchor node is also related to the minimum hop count.

In recent research, various DV-Hop improvements have been proposed [15]–[18], particularly regarding the hop size of anchor nodes (i.e., the average ratio of the distances to the minimum hop counts); however, these techniques do not consider the relationship between the minimum number of hops and the angle between the SNs, which affects the accuracy of the estimated distance between the unknown node and the anchor node.

After distance estimation, the final estimated location can be determined, traditionally by multitrilateration [7]; however, recently, bio-inspired algorithms have been applied to increase accuracy, especially those without the common intersection of triangulated circles [15]–[18], including traditional algorithms such as the genetic algorithm (GA) [19], the ant colony algorithm [20], the bacterial foraging algorithm (BFA) [21], the artificial bee colony [22], and the firefly algorithm [23], and more recent algorithms. For example, Hamidouche *et al.* [24] surveyed algorithms for WSNs from the perspectives of clustering, coverage, and routing, some of which, including GA [25], DE [26], [27], and particle swarm optimization (PSO), were used in the context of WSN localization [28] and, in particular, with the application of DV-Hop. Note that although GA provides high accuracy but is slow to converge [25]–[28], PSO achieves promising results with fast convergence and low complexity; therefore, it is suitable for WSNs [25]–[28].

Traditional PSO has some limitations, but several improved methods have been proposed, such as bare-bones PSO (BBPSO), chaotic PSO (CPSO), and fuzzy PSO (FPSO) [29], to enhance the convergence speed; these derivations can be adopted in the proposed work because a PSO-based approach is only used to determine the particle that minimizes the absolute distance error between the anchor node and the unknown node and between the anchor node and the particle (see also the fitness function in Section IV.D.2).

Therefore, in this paper, we propose a method for identifying the areas of unknown nodes (called local areas) and acquiring the anchor nodes in each local area. The information is processed to determine the locations of unknown nodes without considering all anchor nodes. Then, based on the relationship between the angles and minimum hop counts among SNs, we improve the minimum hop counts, each of which is relevant for the unknown node and the corresponding local anchor nodes. Finally, the proposed method, which is named optimized hop angle relativity for DV-Hop localization (OHAR-DV-Hop), replaces the multitrilateration algorithm of DV-Hop with PSO and applies the bounding box method to limit the approximation area, thereby reducing the number of PSO run cycles required to determine the position that is closest to the actual position. In summary, this research consists of the following key contributions.

- We introduce a technique for improving the minimum hop count using the relationship between the angles and the minimum hop count among the SNs to estimate the distances between the anchor nodes and the unknown node.
- We apply the concept of a bounding box to limit the area of the unknown node using the improved distance.
- We design a PSO fitness function to estimate the locations of the unknown nodes by considering the minimum hop count and the associated anchor nodes.

The paper is organized as follows. Section II describes related works (i.e., WSN localization based on DV-Hop algorithms). Section III solves a problem using the DV-Hop algorithm and the proposed network model. Section IV provides details on the proposed scheme, namely, OHAR-DV-Hop, including the definitions of local anchor nodes, local area specification, hop approximation, and distance estimation for identifying the boundary of the local area. Section V presents the performance evaluation. Finally, Section VI provides the conclusions and suggests potential future work.

II. RELATED WORK

Although a variety of localization techniques [5], [6] are available for WSNs, this section specifically discusses the range-free localization method and focuses on connectivity-based multihop localization, in which the location of an unknown node is determined based on the distances from corresponding anchor nodes to the unknown node. We concentrate on the hop-by-hop connections to relate the minimum hop count to such distances.

In 2001, Niculescu and Nath [7] proposed the DV-Hop algorithm, which consists of the following three steps.

- **1st step:** Each anchor node (i.e., a node at a known location) broadcasts a signal to convey its position information with a hop number that starts at 0. Once another SN receives the signal, it records the smallest hop number from each anchor node and increases the hop number by 1 before further flooding the signal. This process continues until the paths (with the minimum number of hops) to all the anchor nodes are discovered.
- **2nd step:** The average distance per hop (hop size) is determined for each anchor node (node p), as formulated in Eq. (1).

$$Hopsize_p = \frac{\left\| z_p - z_q \right\|}{\sum_{q=1,p\neq q}^{m} h_{p,q}}, \tag{1}$$

where $z_p = (x_p, y_p)$ and $z_q = (x_q, y_q)$ are the coordinate pairs of anchor nodes p and q, respectively, and $h_{p,q}$ is the minimum hop count between the two anchor nodes. After calculating the hop size, an anchor node broadcasts this hop size throughout the network. The unknown node stores only the hop size of the closest anchor node (i.e., that with the smallest minimum hop count).

- **3rd step:** The distance between the unknown node and the anchor node ($d_{p,U}$) is determined using the hop size and the minimum hop count, which were obtained in the second step, as expressed in Eq. (2).

$$d_{p,U} = Hopsize_p \times h_{p,U}, \tag{2}$$

where $h_{p,U}$ denotes the minimum hop count between the unknown node (U) and the anchor node (p).

Eq. (3) shows the relationships between the Cartesian distances from the anchor nodes (1 to m) to the unknown node and the distances from Eq. (2).

$$\begin{cases} (x_U - x_1)^2 + (y_U - y_1)^2 = d_1^2 \\ (x_U - x_2)^2 + (y_U - y_2)^2 = d_2^2 \\ \vdots \\ (x_U - x_m)^2 + (y_U - y_m)^2 = d_m^2 \end{cases} \tag{3}$$

Eq. (3) can be written as Eq. (4) using the multitrilateration method to approximate the location of the unknown node.

$$X = \left(A^T A\right)^{-1} A^T Z, \tag{4}$$

where

$$A = -2 \times \begin{bmatrix} (x_1 - x_m) & (y_1 - y_m) \\ (x_2 - x_m) & (y_2 - y_m) \\ \vdots & \vdots \\ (x_{m-1} - x_m) & (y_{m-1} - y_m) \end{bmatrix}, \tag{5}$$

$$X = \begin{bmatrix} x_U \\ y_U \end{bmatrix}, \tag{6}$$

$$Z = \begin{bmatrix} d_1^2 - d_m^2 - x_1^2 + x_m^2 - y_1^2 + y_m^2 \\ d_2^2 - d_m^2 - x_2^2 + x_m^2 - y_2^2 + y_m^2 \\ \vdots \\ d_{m-1}^2 - d_m^2 - x_{m-1}^2 + x_m^2 - y_{m-1}^2 + y_m^2 \end{bmatrix} \tag{7}$$

Since the first DV-Hop algorithm was proposed, attempts to reduce its estimation error have been made. For example, S. Singh *et al.* [15] improved the DV-Hop location estimation method by modifying how the hop size is determined in the second step. Specifically, they calculated the average hop size by dividing the sum of all anchor node hop sizes by the number of anchor nodes. In step 3, the least-squares method was used instead of the multitrilateration method, thereby increasing the location estimation accuracy beyond that of the traditional DV-Hop approach.

Several DV-Hop derivatives have attempted to improve precision. For example, Gui *et al.* [30] proposed the selection criterion for the 3 anchor nodes with the minimum location error for suitable use in multitrilateration; however, no common intersection of the triangulated circles can lead to low precision.

Note that subsequent research can be divided into 2 categories: improvements to the methods of determining the hop size and increases in the accuracy of the third step. For instance, F. Shahzad *et al.* [31] introduced threshold maxHop based on the simulation configurations. Then, the anchor nodes at which the hop count (for unknown nodes) is below the threshold is used to estimate the location using a Pareto optimal solution without considering the hop size.

Chen and Zhang [16] improved the hop size calculation in the second step by using the least-squares error method. In the third step, PSO was applied in place of the multitrilateration method, and it yielded a more precise location estimation.

G. Sharma and A. Kumar [32] enhanced the hop size calculation by dividing the error between the actual distance and the estimated distance by the minimum hop count between the two anchor nodes. Then, they increased the hop size by the quotient. In the third step, a teaching-learning-based optimization (TLBO) method was applied to localize the unknown node instead of using the multitrilateration method. M. Mehrabi *et al.* [33] improved the DV-Hop position estimation by adopting the shuffled frog leaping algorithm (SFLA) to determine the hop size in step 2 and obtain the lowest error between the distances based on the hop size and the actual distance between two anchor nodes. In addition, they used a hybrid genetic algorithm-PSO (GA-PSO) approach instead of the multitrilateration method in the third step. Here, in each round, after the PSO process, GA operators are applied to the coordinates of the particle to replace the lowest fitness values and accelerate convergence.

In addition, S. Kumar and D. K. Lobiyal [34] improved the hop size calculation by considering the improvement term of each anchor node, which is determined by considering the communication range and the traditional DV-Hop hop size. Then, all anchor node hop sizes are averaged. Next, the average hop size is used to estimate the distances from the anchor nodes to the unknown nodes. Finally, a quadratic program (QP) was proposed to estimate the position of an unknown node. Peng and Li [17] further improved step 3 of the DV-Hop position estimation by applying a GA instead of the multitrilateration algorithm to obtain a more precise location estimation. Later, Sivakumar and Venkatesan [18] compared three position estimation algorithms, namely, the bat optimization algorithm (BOA), modified cuckoo search (MCS), and firefly optimization algorithm (FOA) trilateration, for use in the third step in place of the multitrilateration algorithm. The study showed that the FOA provides the highest localization accuracy.

S. Tomic and I. Mezei [35] proposed three improved versions of DV-Hop: iDV-Hop1, iDV-Hop2, and Quad DV-Hop. iDV-Hop1 relies on 1) the location of the unknown node, which is obtained from DV-Hop, and 2) the position of the anchor node (of which the minimum hop count to the unknown node is the smallest) to form a triangle. The vertices of the triangle are based on the position of the anchor node, the two intersection points of the anchor node communication range and the distance from the unknown node to the anchor node. Then, the triangle centroid is used for localization. IDV-Hop2 differs from iDV-Hop1 in that it creates the triangle by using the unknown node position instead of the anchor node position. In Quad DV-Hop, the authors modified the third step in DV-Hop by adopting a QP instead of a multitrilateration algorithm.

Although evolutionary computation has been used (instead of the multitrilateration algorithm) to improve the localization process, the error of the estimated distance between the unknown node and the anchor node is still high because most methods estimate the hop size by averaging the hop sizes of all anchor nodes, which results in a high error rate. Additionally, the minimum hop counts between the unknown node and the anchor node do not match the actual values because of the difference in SN density between the anchor nodes.

III. NETWORK MODEL, PROBLEM AND MOTIVATION

A. NETWORK MODEL

Let $S = \{S1, \ldots, Sn|n\epsilon N\}$ be the set of all sensor nodes, where N is the number of SNs; $A = \{A1, \ldots, Am|m\epsilon N\}$ is the set of anchor nodes, where m (out of N) is the number of anchor nodes; and $U = \{Um+1, Um+2, \ldots, Un|n\epsilon N\}$ is the set of the unknown nodes. Note that $A \cup U = S$. In a 2-dimensional network, the position of the i^{th} anchor node is (x_i, y_i), where $i = 1, 2, 3, \ldots, m$. Our objective is to estimate the j^{th} sensor node location, which is unknown and denoted by (x_j, y_j), where $j = m+1, m+2, m+3, \ldots, n$.

In this research, we rely on the following assumptions:

- Unknown nodes (SNs at unknown locations) are randomly placed within a 2D experimental area;
- Anchor nodes (SNs whose locations are known) form a grid topology that spans the 2D experimental area [36]–[38]; and
- All SNs have the same transmission range [16], [33].

Assumption 1: When DV-Hop is considered, the number of hops in the shortest path between a pair of SNs (from *Sp* to *Sq*) is the minimum hop count [30], [31].

We apply Assumption 1 to estimate the hop count between SNs (*Sp* and *Sq*). Note that the notations used throughout this paper are shown in Table 1.

TABLE 1. List of notations.

Notation	Definition
S	Set of all sensor nodes
A	Set of anchor nodes
U	Unknown node
x_{Sp}, y_{Sp}	Location of sensor node *Sp*
cr	Communication range
Local area	Areas where the experimental area is equally divided based on the anchor node positions (see Fig. 1)
Af	Local anchor node for which the shortest path to the unknown node has the smallest minimum hop count, e.g., *A1* in local area 1 (see Fig. 1b)
Ai	Another local anchor node (not *Af*) that covers the local area of the unknown node, e.g., *A2*, *A4*, and *A5* in local area 1 (see Fig. 1b)
Ak	Another anchor node, which is distinct from *Af* and *Ai*, e.g., *A3*, *A6*, *A7*, *A8*, and *A9* in local area 1 (see Fig. 1b)
NL	Number of *Ai*; *NL* = 3 in Fig. 1b
θ_k	Angle of *Ak* ($\theta_k = \angle AkAfAi$) calculated from the law of cosines
$f_1(x), f_2(x)$	Function for the adjusted number of hops between *Ai* and *U*, where *x* denotes any interested anchor node, e.g., *Ai*, *Ak*, and *Af*; only $0 \leq \theta_k \leq \frac{\pi}{2}$ will apply for f_1, and only $\frac{3\pi}{4} \leq \theta_k \leq \pi$ will apply for f_2
$f_3(x), f_4(x)$	Function for the adjusted number of hops between *Ai* and *U*, where *x* denotes any anchor node of interest, e.g., *Ai*, *Ai′*, and *Ak* for f_3 or *Ai* and *Ak* for f_4.
$g(y)$	Function used to determine whether an unknown location is near the center or edge of an area, where *y* denotes (*Ai*, *Ai′*), which is a pair of diagonally opposite anchor nodes in the same local area
$q_1(x), q_2(x)$	Function used to adjust the distance between *Ai* and *U*, where *x* is the distance between *Ai* and *U*; only one hop will apply for q_1, and two hops will apply for q_2
$h_{Sp,Sq}$	Number of hops between *Sp* and *Sq*
$h^{optimized1}{}_{Ai,U}$, $h^{optimized2}{}_{Ai,U}$	Adjusted number of hops in Stages 1 and 2 between *Ai* and *U*
$h^{optimized}{}_{Ai,U}$	Adjusted number of hops between *Ai* and *U*, i.e., an average over $h^{optimized1}{}_{Ai,U}$ and $h^{optimized2}{}_{Ai,U}$
$d_{Sp,Sq}$	Distance between *Sp* and *Sq*
$F(x_k(t))$	PSO fitness function of the k^{th} particle in round *t*

Fig. 1 shows an example of the placement of the unknown and anchor nodes and the connections between SNs within the experimental area. Note that Fig. 1b also shows a possible path between the two anchor nodes (*Af* and *Ai*; *Af* and *Ak*), e.g., from *A5* to *A1* ($h_{A1,A5} = 4$) and from *A9* to *A1* ($h_{A1,A9} = 8$).

B. PROBLEM AND MOTIVATION

The traditional DV-Hop algorithm may provide location estimations with high errors because it incorporates all anchor nodes into the calculation without considering the hop sizes of the anchor nodes, i.e., the distance to hop count ratio

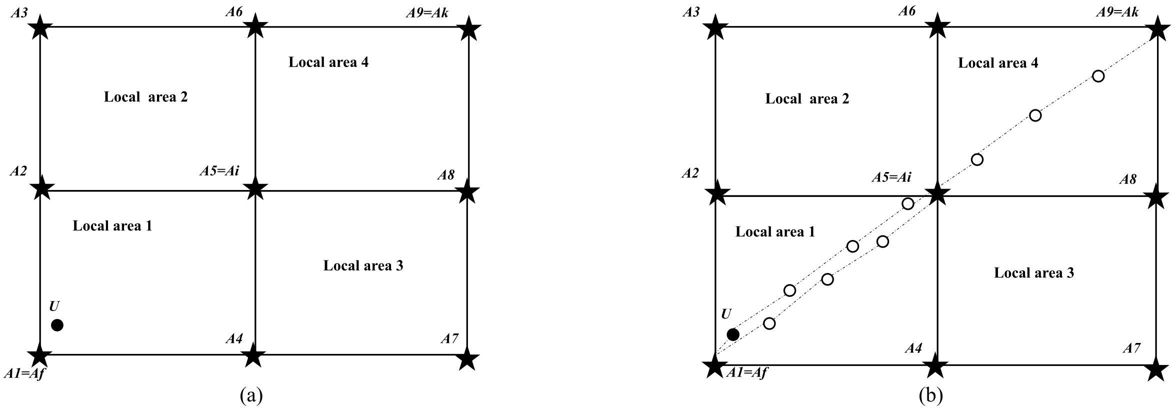


FIGURE 1. Network model: anchor node = solid star, unknown node = circle, unknown node (location to be estimated) = solid circle, dashed line = possible path between sensor nodes. (a) Unknown node (*U*) and anchor nodes (*Ai*) in each local area. (b) Possible paths between the anchor nodes (*A1* to *A5*; *A1* to *A9*) and the unknown node (*U*).

(see Eq. (1)). Figs. 2a and 2b show two cases in which the positions of unknown nodes are different (U and U'), as follows.

- **Case I:** $h_{Ai,U} + h_{Af,U} = h_{Af,Ai}$ and $h_{Ak,U} + h_{Af,U} = h_{Af,Ak}$ or $h_{Ai,U} + h_{Af,U} > h_{Af,Ai}$ and $h_{Ak,U} + h_{Af,U} > h_{Af,Ak}$, i.e., the shortest path is toward the unknown node (U), (see Figs. 2c and 2d, where $h_{A5,U} = 3$, $h_{A1,U} = 1$, $h_{A1,A5} = 4$, $h_{A9,U} = 7$, and $h_{A1,A9} = 8$), and the minimum hop count from Ai to Af (here $= 4$) is equal to the minimum hop count from that node toward U (i.e., U is within the path; Ai to $U = 3$ hops and U to $Af = 1$ hop) and similar to the minimum hop count from Ak to Af toward $U (= 8)$ or Ak to $U = 7$ hops and U to $Af = 1$ hop. In this case, the traditional DV-Hop method provides high accuracy.
- **Case II:** $h_{Ai,U} + h_{Af,U} > h_{Af,Ai}$ and $h_{Ak,U} + h_{Af,U} = h_{Af,Ak}$ or $h_{Ai,U} + h_{Af,U} = h_{Af,Ai}$ and $h_{Ak,U} + h_{Af,U} > h_{Af,Ak}$, (see Figs. 2e and 2f, where $h_{A5,U} = 4$, $h_{A1,U} = 1$, $h_{A1,A5} = 4$, $h_{A9,U} = 7$, and $h_{A1,A9} = 8$). In this case, U is not in the shortest path between the two anchor nodes. For example, the minimum hop count from Ai to $U = 4$, which is the same as that from Ai to Af. Therefore, the estimated location of the unknown node will be within a circle with center Ai, i.e., within $h_{Af,Ai} (= 4)$. In the traditional DV-Hop method, this hop count is not fine-grained.

Thus, the actual hop count should be small (less than 4). The count can be lowered with help from another anchor node, such as Ak, where $h_{Ak,U} + h_{Af,U} = h_{Af,Ak}$. This relation implies that the localization based on Ak estimates a location outside the circle with center Ak, i.e., between $h_{Ak,U}$ and $h_{Af,Ak}$ (=7 and 8). Thus, the proper minimum hop count should lie within the circle of Ai but outside the circle of Ak.

Instead of using all anchor nodes, we propose a method of selecting appropriate anchor nodes (Ai). Specifically, we focus on a grid of anchor nodes and select the anchor nodes that share the same local area as the unknown node. Other anchor nodes (Ak) can be chosen to improve the minimum hop count between Ai and U relative to the hop count between the reference anchor node (Af) and the unknown node (U) based on the relationship between the minimum hop count and the angles Ai and Ak. The improved minimum hop count is used to estimate the distance between Ai and U. Finally, the unknown node position is approximated based on PSO and the bounding box method, which limits the search area to determine the unknown node position.

IV. RELATIVE HOP ANGLE CONNECTIVITY-BASED LOCALIZATION

Fig. 3 shows the overall process flow of this research, which consists of five major components.

- Local area and anchor node identification: this step is used to determine local areas and local anchor nodes where an unknown node is encircled such that the number of hops between the anchor node and another anchor node and between the anchor node and the unknown node is minimized.
- Number of hops estimation: this step is used to optimize the number of hops between a particular anchor node and an unknown node using the hop count of anchor nodes with corresponding angles.
- Distance estimation: using the optimized hop count, the distance (between the anchor node and the unknown node) is estimated using the hope size (i.e., the anchor-to-anchor distance divided by the hop count).
- Bounding box: the derived distance is used to limit the unknown node boundary.
- Unknown node localization: PSO is applied to search for the location of the unknown node given the fitness function.

Theorem 1: The minimum hop count between an unknown node (U) and an anchor sensor node or anchor node (Ai) depends on the minimum hop count between the unknown node and another anchor node (Ak). If Ai and Ak are on the same side when they are compared to a reference anchor node (Af), then the node with the smallest minimum hop count

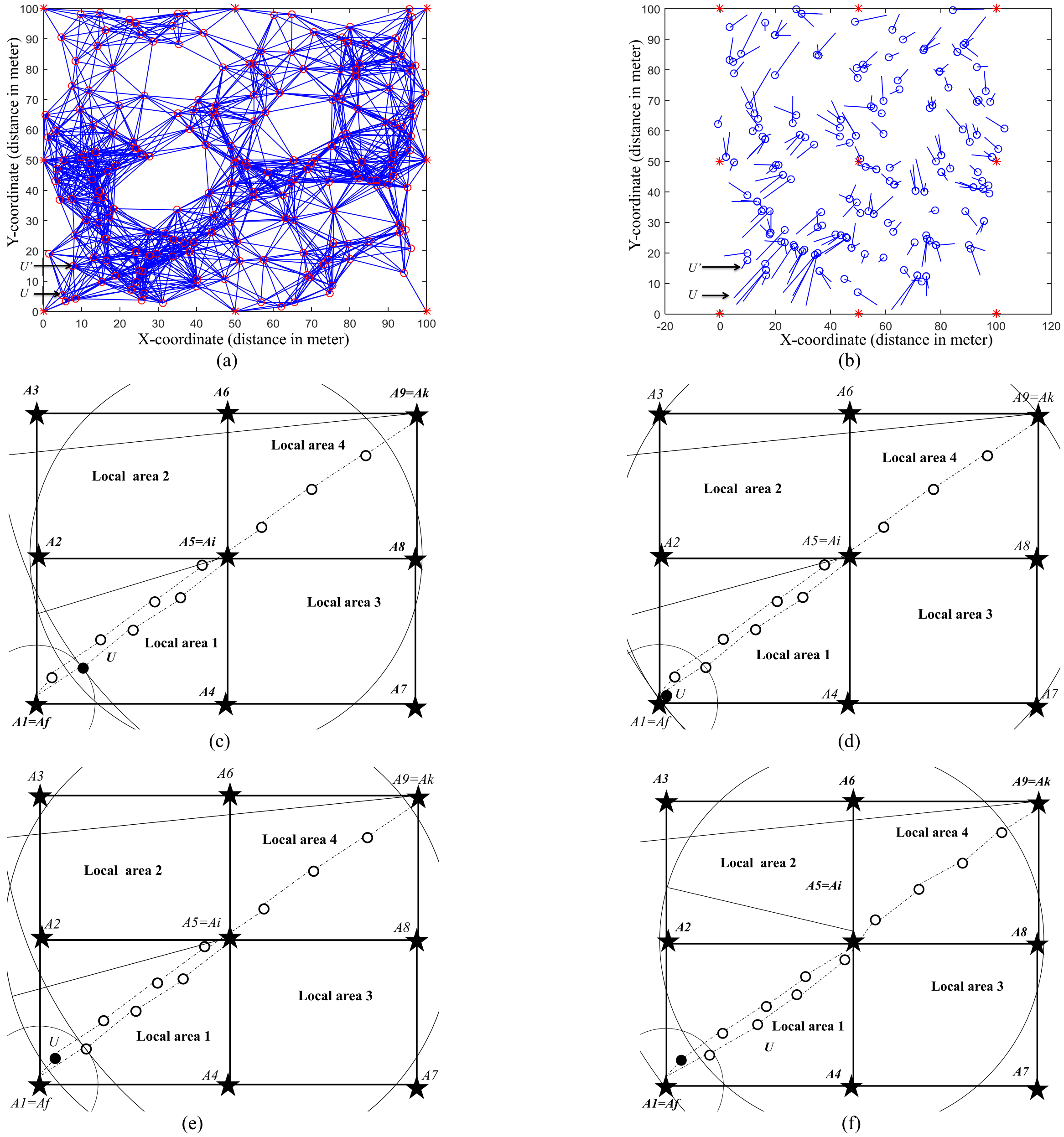


FIGURE 2. Examples of problems in DV-Hop. (a) Node deployment: Case I and Case II. (b) Location error: Case I and Case II. (c) Case I: $h_{Ai,U} + h_{Af,U} = h_{Af,Ai}$ and $h_{Ak,U} + h_{Af,U} = h_{Af,Ak}$. (d) Case I: $h_{Ai,U} + h_{Af,U} > h_{Af,Ai}$ and $h_{Ak,U} + h_{Af,U} > h_{Af,Ak}$. (e) Case II: $h_{Ai,U} + h_{Af,U} > h_{Af,Ai}$ and $h_{Ak,U} + h_{Af,U} = h_{Af,Ak}$. (f) Case II: $h_{Ai,U} + h_{Af,U} = h_{Af,Ai}$ and $h_{Ak,U} + h_{Af,U} > h_{Af,Ak}$.

for the unknown node ($h_{Af,U} = 1$) is selected to determine the proper Ak, which has a direct impact on the optimized $h_{Ai,U}$ ($h^{optimized}_{Ai,U}$).

Proof: The value of $h_{Ai,U}$ depends on $h_{Ak,U}$ if $h_{Af,U} = 1$, as shown in Fig. 4.

In the first case,

1) the unknown node is on the shortest path from *Ai* to *Af* and the path from *Ak* to *Af*, and
2) $h_{Af,U} = 1$ and $h_{Af,U} + h_{Ai,U} = h_{Af,Ai}$, $h_{Ak,U} + h_{Af,U}$ is equal to $h_{Af,Ak}$ (as shown in Figs. 4a).

In the second case,

1) the unknown node is not on the shortest path from *Ai* to *Af* but on the path from *Ak* to *Af*, and
2) $h_{Af,U} = 1$ and $h_{Af,U} + h_{Ai,U} > h_{Af,Ai}$, $h_{Ak,U} + h_{Af,U}$ is equal to $h_{Af,Ak}$ (as shown in Fig. 4b).

In the third case,

1) the unknown node is on the shortest path from *Ai* to *Af* but not on the path from *Ak* to *Af*, and
2) $h_{Af,U} = 1$ and $h_{Af,U} + h_{Ai,U} = h_{Af,Ai}$, $h_{Ak,U} + h_{Af,U}$ is greater than $h_{Af,Ak}$ (as shown in Fig. 4c).

In the fourth case,

1) the unknown node is not on the shortest path from *Ai* to *Af* or the path from *Ak* to *Af*, and
2) $h_{Af,U} = 1$ and $h_{Af,U} + h_{Ai,U} > h_{Af,Ai}$, $h_{Ak,U} + h_{Af,U}$ is greater than $h_{Af,Ak}$ (as shown in Fig. 4d).

Proposition 1: Let θ_k in Theorem 1 range from $0 \leq \theta_k \leq \frac{\pi}{2}$.

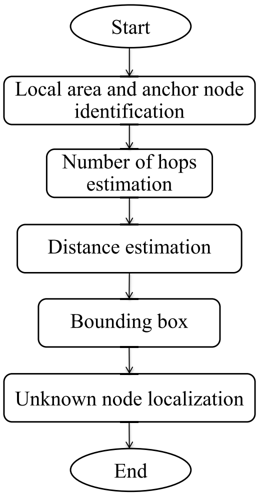

Start
Local area and anchor node identification
Number of hops estimation
Distance estimation
Bounding box
Unknown node localization
End

FIGURE 3. **Overall process flow.**

Proof: Suppose that $\frac{\pi}{2} < \theta_k < \frac{3\pi}{4}$ as shown in Fig. 5. If the unknown node is on the shortest path from *Ai* to *Af*, then it cannot be on the shortest path from *Ak* to *Af* because 1) $h_{Af,U} = 1$ and 2) $h_{Af,U} + h_{Ai,U} = h_{Af,Ai}$, which implies that $h_{Ak,U} + h_{Af,U} > h_{Af,Ak}$; additionally, the unknown node will not always be on the shortest path from *Ai* to *Af*.

The method proposed in this study uses *Ak* with an angle θ_k ($0 \leq \theta_k \leq \frac{\pi}{2}$) because an unknown node with a wider angle would be on the shortest path from *Ai* to *Af* but not on the path from *Ak* to *Af*,as shown in Fig. 5. Therefore, the remaining anchor nodes do not need to be used to adjust $h_{Ai,U}$, and a decrease in the accuracy can be avoided.

Theorem 2: The minimum hop count between a pair that consists of an unknown node (U) and an anchor node (*Ai*) depends on the minimum hop count between the unknown node and another anchor node (e.g., *Ak*) on the opposite side of Ai when their positions are relative to the referenced anchor node *Af*.

Proof: As shown in Fig. 6, if $h_{Af,U} = 1$, the value of $h_{Ai,U}$ depends on that of $h_{Ak,U}$.

- In the case where U is on the shortest path from Ai to *Af*, if $h_{Af,U} = 1$ and $h_{Af,U} + h_{Ai,U} = h_{Af,Ai}$, then $h_{Ak,Af} + h_{Af,U} = h_{U,Ak}$, as shown in Fig. 6a.
- In the case where U is on the shortest path from Ai to *Af*, if $h_{Af,U} = 1$ and $h_{Af,U} + h_{Ai,U} = h_{Af,Ai}$, then $h_{Ak,Af} + h_{Af,U} > h_{U,Ak}$ (as shown in Fig. 6b).
- In the case where U is not on the shortest path from Ai to *Af*,

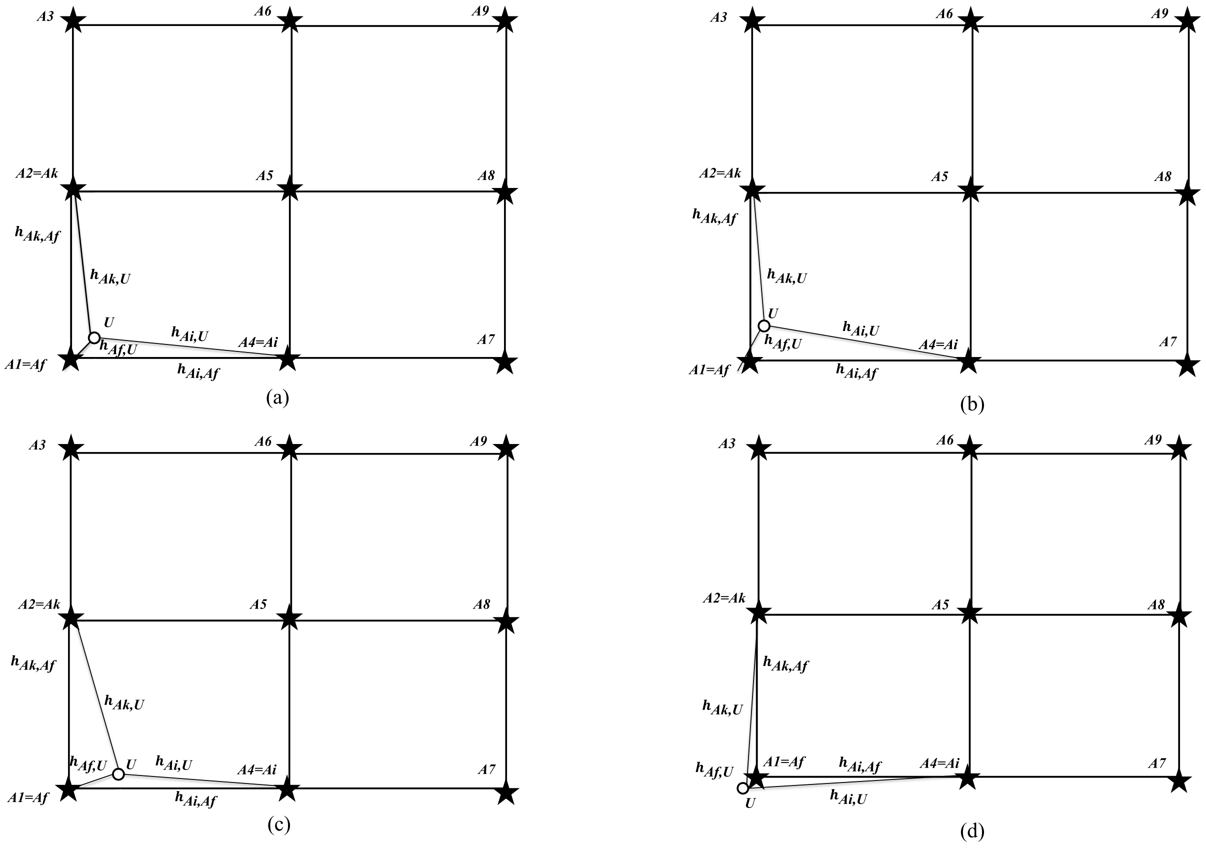


FIGURE 4. **Relationships between $h_{Ai,U}$ and $h_{Ak,U}$ when *Ak* and *Ai* are on the same side. (a) If $h_{Af,U} + h_{Ai,U} = h_{Af,Ai}$, then $h_{Ak,U} + h_{Af,U} = h_{Af,Ak}$. (b) If $h_{Af,U} + h_{Ai,U} > h_{Af,Ai}$, then $h_{Ak,U} + h_{Af,U} = h_{Af,Ak}$. (c) If $h_{Af,U} + h_{Ai,U} = h_{Af,Ai}$, then $h_{Ak,U} + h_{Af,U} > h_{Af,Ak}$. (d) If $h_{Af,U} + h_{Ai,U} > h_{Af,Ai}$, then $h_{Ak,U} + h_{Af,U} > h_{Af,Ak}$.**

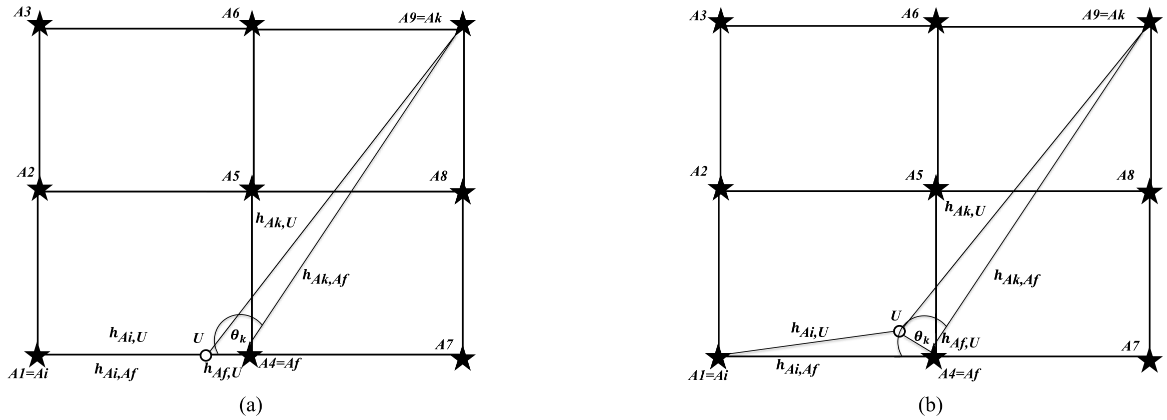


FIGURE 5. If $h_{Af,U} + h_{Ai,U} = h_{Af,Ai}$ and $\frac{3\pi}{4} > \theta_k > \frac{\pi}{2}$, then $h_{Ak,U} + h_{Af,U} > h_{Af,Ak}$. (a) $\frac{3\pi}{4} > \theta_k > \frac{\pi}{2}$ and $h_{Ak,U} > h_{Ak,Af}$. (b) $\frac{3\pi}{4} > \theta_k > \frac{\pi}{2}$ and $h_{Ak,U} = h_{Ak,Af}$.

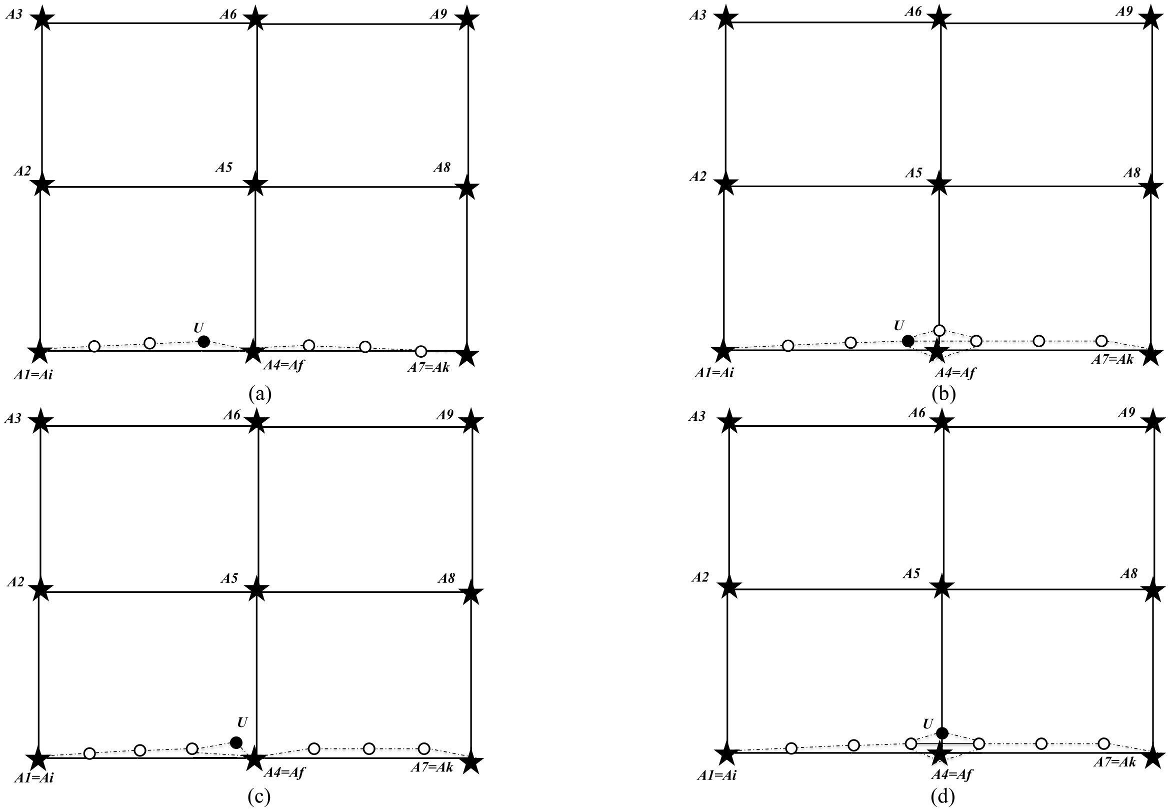


FIGURE 6. Relationships between $h_{Ai,U}$ and $h_{Ak,U}$ when *Ak* is opposite *Ai*. (a) If $h_{Af,U} + h_{Ai,U} = h_{Af,Ai}$, then $h_{Ak,Af} + h_{Af,U} = h_{U,Ak}$. (b) If $h_{Af,U} + h_{Ai,U} = h_{Af,Ai}$, then $h_{Ak,Af} + h_{Af,U} > h_{U,Ak}$. (c) If $h_{Af,U} + h_{Ai,U} > h_{Af,Ai}$, then $h_{Ak,Af} + h_{Af,U} = h_{U,Ak}$. (d) If $h_{Af,U} + h_{Ai,U} > h_{Af,Ai}$, then $h_{Ak,Af} + h_{Af,U} > h_{U,Ak}$.

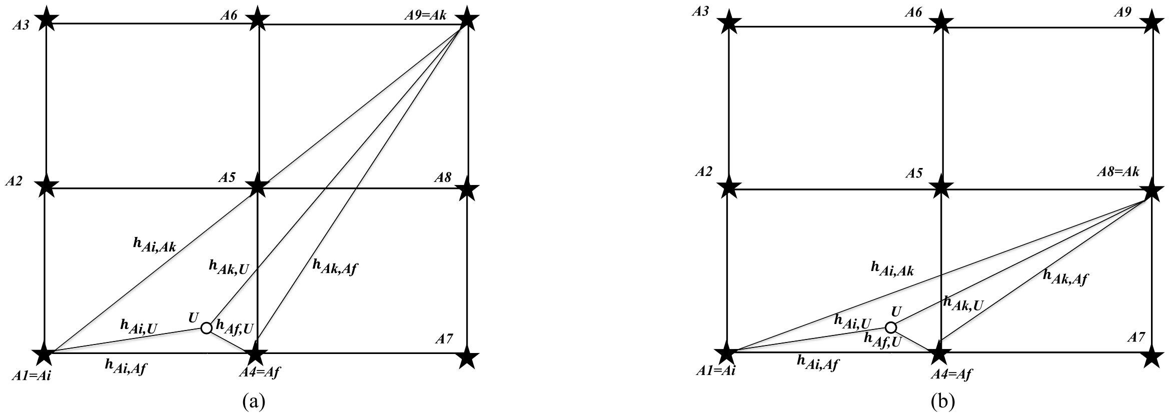
if $h_{Af,U} = 1$ and $h_{Af,U} + h_{Ai,U} > h_{Af,Ai}$, then $h_{Ak,Af} + h_{Af,U} = h_{U,Ak}$ (as shown in Fig. 6c).

- In the case where U is not on the shortest path from Ai to Af
  if $h_{Af,U} = 1$ and $h_{Af,U} + h_{Ai,U} > h_{Af,Ai}$, then $h_{Ak,Af} + h_{Af,U} > h_{U,Ak}$ (as shown in Fig. 6d) because the position of U is very close to that of Af.

Proposition 2: Let θ_k in Theorem 2 range from $\frac{3\pi}{4} \leq \theta_k \leq \pi$.

Proof: Suppose that $\frac{\pi}{2} < \theta_k < \frac{3\pi}{4}$ and the unknown node is on the shortest path from Ai to Af, as shown in Fig. 7a. Then, the unknown node cannot be on the shortest path from Ai to Ak because $(h_{Ak,U} + h_{Ai,U}) > h_{Ai,Ak}$ and $h_{Af,U} > 1$.

Therefore, in this paper, we specify the angle of Ak (θ_k) to avoid evaluating the remaining anchor nodes and improve the number of hops between U and Ai, which normally causes additional errors, especially in the case of $\frac{\pi}{2} < \theta_k < \frac{3\pi}{4}$. Furthermore, in the case of $\frac{3\pi}{4} \leq \theta_k \leq \pi$ in Fig. 7b,



(a)

(b)

FIGURE 7. **Selecting the appropriate *Ak* to improve the number of hops between *U* and *Ai*. (a) $\frac{\pi}{2} < \theta_k < \frac{3\pi}{4}$. (b) $\frac{3\pi}{4} \leq \theta_k \leq \pi$ (complying with Theorem 2).**

the unknown node might be on the shortest path from Ai to Ak. Therefore, we can use Theorem 2 to determine the relation between the minimum hop count and the angle between SNs.

Note that from Propositions 1 and 2, we do not consider Akvalues from $\frac{\pi}{2} < \theta_k < \frac{3\pi}{4}$ because they do not affect the minimum hop count between Ai and U because U cannot be a member node that creates the shortest path from Ak to Af. Therefore, we cannot compare the minimum hop count between $h_{Ak,Af}$ and $h_{Ak,U}$ for Theorem 1 and between $h_{Ai,Ak}$ and $h_{Ai,U}$ to $h_{Ak,U}$ for Theorem 2, although the difference between these hop counts will be used to adjust $h_{Ai,U}$.

A. LOCAL AREA AND LOCAL ANCHOR NODE IDENTIFICATION

Choosing anchor nodes to estimate the positions of unknown nodes affects the estimation accuracy. After DV-Hop determines the shortest path, if the minimum hop count is small, then the approximate distance between an unknown node and each anchor node is less erroneous. As a result, we propose an anchor node selection scheme that chooses anchor nodes for which the shortest path to the unknown node has the smallest minimum hop count to identify the local area and the local anchor nodes. In this paper, we categorize the positions of the unknown nodes using two schemes:

1) THE UNKNOWN NODE IS CLOSE TO AN ANCHOR NODE AT AN AREA CORNER

In this scheme, we can promptly determine the local area that contains the unknown node. For the example in Fig. 8, the unknown node (U) is closest to $A1$, which is the anchor node in local area 1; thus, U is in the same area (this also applies for U close to $A3$, $A7$, and $A9$).

2) THE UNKNOWN NODE IS FAR FROM THE ANCHOR NODES AT AN AREA CORNER

In this scheme, we determine the local area that contains the unknown node by considering the difference between the minimum hop count from each local anchor node to the local anchor node with the smallest minimum hop count (Af, which is $A4$ in Fig. 9) and the minimum hop count from the unknown node to each local anchor node next to Af, as derived in Eq. (8).

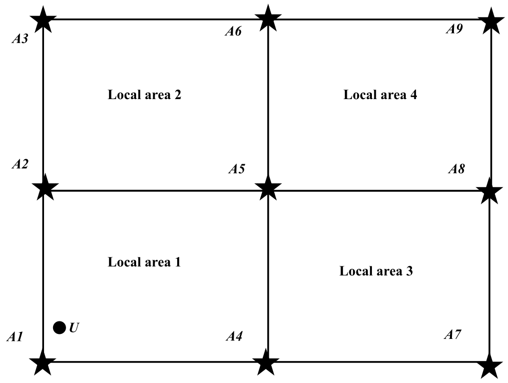


FIGURE 8. **The unknown node is close to an anchor node at an area corner.**

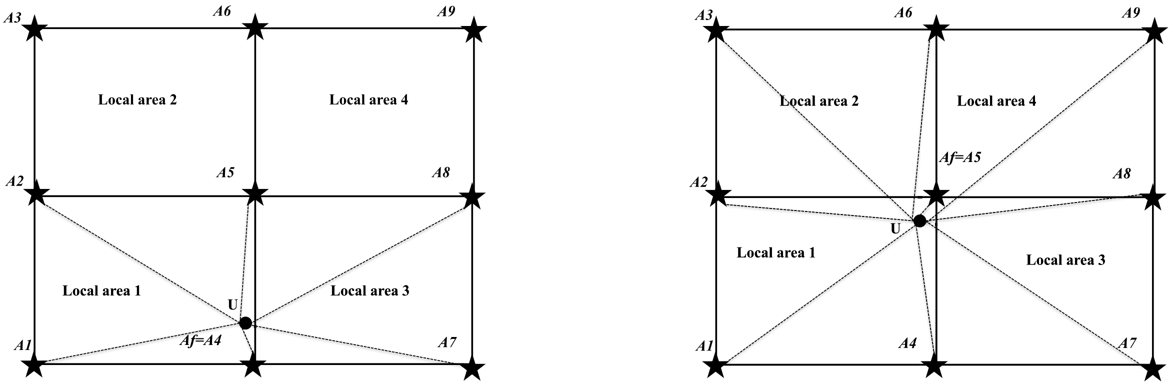
$$dif = \sum_{i=1}^{NL} |(h_{U,Ai} + h_{U,Af}) - h_{Af,Ai}| \tag{8}$$

Note that the unknown node is specified in the local area that yields the smallest dif value (in Eq. (8)). For example, the unknown node in Fig. 9a is in local area 1 because the corresponding dif value is the smallest (=1), whereas the dif value in local area 3 is 2. In Fig. 9b, $A5$ is the local node in all local areas, and the dif value of local area 1 is the smallest ($dif = 0$). Therefore, the unknown node is specified in local area 1 (the dif values of local areas 2, 3, and 4 are 2, 2, and 3, respectively).

In the case when the dif values of two local values are equal (which occurs when the unknown node is on a horizontal or vertical line that connects 2 anchor nodes), the unknown node should be in the local area with the lowest sum of the minimum hop counts between the unknown node and the associated anchor nodes (Ai).

B. HOP APPROXIMATION

After determining the appropriate local anchor nodes for the unknown node from the previous subsection, we identify the



(a) (b)

FIGURE 9. An unknown node close to an anchor node associated with more than one area. (a) The unknown node is close to *A4* in both local areas 1 and 3. (b) The unknown node is close to *A5* in more than two local areas.

local area of the unknown node together with the positions of all local anchor nodes. When we calculate the hop size (which is the distance between the two local anchor nodes divided by the hop count) based on DV-Hop, we do not consider the other anchor nodes because they might cause the hop size value to deviate further from the actual value.

Fig. 10 shows an example of calculating the hop size between *A1* and *A2* considering the position of *U* (the closest node), and it considers only the minimum hop counts between *A1* and *A2* and not those between *A1* and *A3*, *A1* and *A4*, *A2* and *A4*, *A3* and *A4*, and *A2* and *A3*.

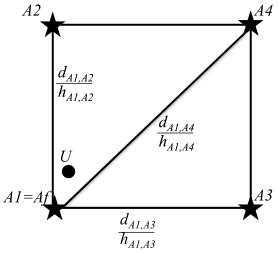

A2
A4
$\frac{d_{A1,A2}}{h_{A1,A2}}$
$\frac{d_{A1,A4}}{h_{A1,A4}}$
U
A1=Af
A3
$\frac{d_{A1,A3}}{h_{A1,A3}}$

FIGURE 10. Hop sizes between two local anchor nodes.

In this paper, we divide the methods of estimating the minimum hop counts between 1) the unknown node (U) and 2) the local anchor node with the lowest minimum hop count (i.e., *Af* or *A1* in Fig. 10) into 2 cases: U and *Af* are separated by only one hop, and U and *Af* are separated by more than one hop. Note that the estimated minimum hop counts are subsequently used to approximate the distances between the unknown node and the local anchor nodes.

1) THE UNKNOWN NODE IS ONLY ONE HOP AWAY FROM THE LOCAL ANCHOR NODE WITH THE SMALLEST MINIMUM HOP COUNT

Let *Af* be the local anchor node with the smallest minimum hop count; thus, $h_{Af,U} = 1$. Let *Ai* denote the local anchor node with a minimum hop count ($h_{Ai,U}$) that needs to be updated. Let *Ak* be the anchor node that is used to update the minimum hop counts ($h_{Ak,U}$) of $h_{Ai,U}$. According to Theorem 1, given $0 \leq \theta_k \leq \frac{\pi}{2}$, we estimate $h_{Ai,U}$ using Eq. (9), shown at the bottom of this page, and α is a variable used to adjust the minimum hop count. Note that based on Theorem 2, given $\frac{3\pi}{4} \leq \theta_k \leq \pi$, we estimate $h_{Ai,U}$ using Eq. (10), shown at the bottom of this page.

Because 1) the optimal value of the minimum hop count of *Ai* ($h^{optimized}_{Ai,U}$) should be in the range of $h_{Ai,U} \pm$ a1/2 -unit hop count and 2) $|h_{Ai,U} - h^{optimized}_{Ai,U}|$ is less than or equal to $|(h_{Ai,U} +$ 1-unit hop count$) - h^{optimized}_{Ai,U}|$ but greater than $|(h_{Ai,U} -$ 1-unit hop count$) - h^{optimized}_{Ai,U}|$, we set the value of α in Eqs. 9, 10, 12 and 13 to a 1/2 -unit hop count. Consequently, regardless of whether the number of anchor

$$f_1(x) = \begin{cases} h_{Ai,U},\ (h_{U,Ai} + h_{U,Af}) == h_{Af,Ai} \text{ and } (h_{U,Ak} + h_{U,Af}) == h_{Af,Ak} \\ h_{Ai,U} + \alpha,\ (h_{U,Ai} + h_{U,Af}) == h_{Af,Ai} \text{ and } (h_{U,Ak} + h_{U,Af}) - h_{Af,Ak} == 1 \\ h_{Ai,U} - \alpha,\ (h_{U,Ai} + h_{U,Af}) - h_{Af,Ai} == 1 \text{ and } (h_{U,Ak} + h_{U,Af}) == h_{Af,Ak} \\ h_{Ai,U},\ (h_{U,Ai} + h_{U,Af}) - h_{Af,Ai} == 1 \text{ and } (h_{U,Ak} + h_{U,Af}) - h_{Af,Ak} == 1 \end{cases} \tag{9}$$

$$f_2(x) = \begin{cases} h_{Ai,U},\ (h_{U,Ai} + h_{U,Af}) == h_{Af,Ai} \text{ and } (h_{Ak,Af} + h_{U,Af}) == h_{U,Ak} \\ h_{Ai,U} + \alpha,\ (h_{U,Ai} + h_{U,Af}) == h_{Af,Ai} \text{ and } (h_{Ak,Af} + h_{U,Af}) - h_{U,Ak} == 1 \\ h_{Ai,U} - \alpha,\ (h_{U,Ai} + h_{U,Af}) - h_{Af,Ai} == 1 \text{ and } (h_{Ak,Af} + h_{U,Af}) == h_{U,Ak} \\ h_{Ai,U},\ (h_{U,Ai} + h_{U,Af}) - h_{Af,Ai} == 1 \text{ and } (h_{Ak,Af} + h_{U,Af}) - h_{U,Ak} == 1 \end{cases} \tag{10}$$

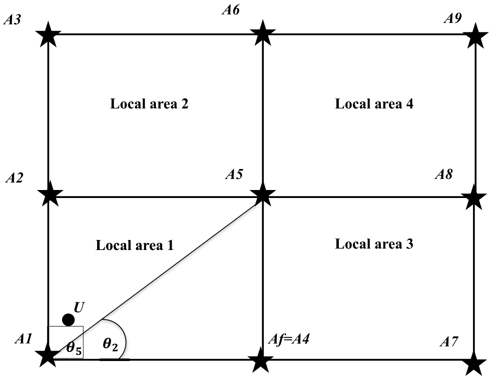


FIGURE 11. Example of how to choose *Ak* to improve the minimum hop count of *Ai* for the unknown node.

nodes (Ak) increases or decreases, $h_{Ai,U}^{optimized}$ will be in the range of $h_{Ai,U}\pm$ a 1/2 -unit hop count.

To determine Ak, two stages are applied to improve the minimum hop count of Ai. In Stage 1, the anchor nodes used for the improvement are in the same local area as Ai. In Stage 2, the anchor nodes are outside the local area.

- **Stage 1:** The anchor nodes used for the improvement are in the same local area as Ai in Eq. (9), and $0 \leq \theta_k \leq \frac{\pi}{2}$, as shown in Fig. 11. In this topology, $Ai = A4$ in local area 1 and $Af = A1$. Therefore, the usable Ak could be $A2$ or $A5$ (θ_k could be θ_2 and θ_5.) After obtaining Ak, we can calculate the improved minimum hop count of Ai. For the case in which Ak involves 2 nodes, we calculate the improvement to the minimum hop count by averaging all improved Ai minimum hop counts from all probable Ak nodes.
- **Stage 2:** The anchor nodes used for the improvement are outside the local area. After obtaining Ak, we calculate the improvement to the minimum hop count of Ai via a method similar to that in Stage 1. For the case in which Ak can be more than one node (= 4 nodes in Fig. 11), we calculate the improved Ai minimum hop count by averaging all improved Ai minimum hop counts from all probable Ak nodes.
    - If Ak is on the same side as Ai in Eq. (9), then $Ak = \{A3, A6, A7, A8, A9\}$ can be used in Eq. (9) because $0 \leq \theta_k \leq \frac{\pi}{2}$.

Similarly, consider $Ai = A2$. In Stage 1, the usable Ak could be $\{A4, A5\}$, and in Stage 2, the usable Ak could be $\{A3, A6, A7, A8, A9\}$ based on Eq. (9).

After performing Ak selection in the two stages, the minimum hop count is improved. Fig. 12 presents an example of adjusting the hop count from Ai to U ($h_{Ai,U}^{optimized}$) and shows that there are 4 possible locations of the unknown node that are close to local anchor nodes, namely, Af: $\{A1, A5, A2, A4\}$. Compared with the former two nodes ($A1$ and $A5$), the calculations in the cases of $A2$ and $A4$ are similar because they are both in two local areas.

In Fig. 12a, the minimum hop count between U and $A1$ is the smallest; thus, $Af = A1$. $A1$ is in only one local area, and the set of local anchor nodes (Ai) is $\{A2, A4, A5\}$.

For the case when $Ai = A2$, we improve the minimum hop count ($h_{A2,U}^{optimized}$) as follows.

- **Stage 1:** Solve $f_1(A4)$ and $f_1(A5)$ because the angles of $A4$ and $A5$ satisfy the condition $0 \leq \theta_k \leq \frac{\pi}{2}$; thus, $Ak = \{A4, A5\}$. The improved minimum hop count from $A2$ to U is $h_{A2,U}^{optimized1} = \frac{\sum_{k=4,5} f_1(Ak)}{|Ak|}$ according to Eq. (9).
- **Stage 2:** Solve $f_1(A3)$, $f_1(A6)$, $f_1(A7)$, $f_1(A8)$, and $f_1(A9)$. The angles of $A3$, $A6$, $A7$, $A8$, and $A9$ satisfy the condition $0 \leq \theta_k \leq \frac{\pi}{2}$; thus, $Ak = \{A3, A6, A7, A8, A9\}$. All nodes in Ak are on the same side of Ai; therefore, only Eq. (9) is applied. Then, the improved minimum hop count from $A2$ to the unknown node is the average of all the results: $h_{A2,U}^{optimized2} = \frac{\sum_{k=3,6,7,8,9} f_1(Ak)}{|Ak|}$.

Next, we average the improved minimum hop counts from both Stages 1 and 2 as $h_{A2,U}^{optimized} = \frac{\sum_{j=1}^{2} h_{A2,U}^{optimizedj}}{2}$.

Similar to the case above, when $Ai = \{A4, A5\}$, we improve the minimum hop count as follows.

- For $Ai = A4$, we obtain $h_{A4,U}^{optimized1} = \frac{\sum_{k=2,5} f_1(Ak)}{|Ak|}$ in Stage 1 and $h_{A4,U}^{optimized2} = \frac{\sum_{k=3,6,7,8,9} f_1(Ak)}{|Ak|}$ in Stage 2.
- For $Ai = A5$, we obtain $h_{A5,U}^{optimized1} = \frac{\sum_{k=2,4} f_1(Ak)}{|Ak|}$ in Stage 1 and $h_{A5,U}^{optimized2} = \frac{\sum_{k=3,6,7,8,9} f_1(Ak)}{|Ak|}$ in Stage 2.

Then, the improved minimum hop counts are averaged as follows.

- $h_{A4,U}^{optimized} = \frac{\sum_{j=1}^{2} h_{A4,U}^{optimizedj}}{2}$
- $h_{A5,U}^{optimized} = \frac{\sum_{j=1}^{2} h_{A5,U}^{optimizedj}}{2}$

In Fig. 12b, the minimum hop count between U and $A5$ is the smallest; thus, $Af = A5$. $A5$ can be considered in all four local areas, and the set of local anchor nodes (Ai) is $\{A1, A2, A4\}$. The steps in this improvement are similar to those in the case of Fig. 12a, except that Eq. (10) is also involved in this case.

- For $Ai = A1$, we obtain $h_{A1,U}^{optimized1} = \frac{\sum_{k=2,4} f_1(Ak)}{|Ak|}$ in Stage 1 and by applying Eq. (9) to the case in which $0 \leq \theta_k \leq \frac{\pi}{2}$ and $h_{A1,U}^{optimized2} = \frac{\sum_{k=3,7} f_1(Ak) + \sum_{k=6,8,9} f_2(Ak)}{|Ak|}$ in Stage 2 by applying Eq. (10) to the case in which $\frac{3\pi}{4} \leq \theta_k \leq \pi$.
- Similarly, for $Ai = A2$, we obtain $h_{A2,U}^{optimized1} = \frac{\sum_{k=1,4} f_1(Ak)}{|Ak|}$ in Stage 1 and $h_{A2,U}^{optimized2} = \frac{\sum_{k=3,6} f_1(Ak) + \sum_{k=7,8,9} f_2(Ak)}{|Ak|}$ in Stage 2.
- Similarly, for $Ai = A4$, we obtain $h_{A4,U}^{optimized1} = \frac{\sum_{k=1,2} f_1(Ak)}{|Ak|}$ in Stage 1 and $h_{A4,U}^{optimized2} = \frac{\sum_{k=7,8} f_1(Ak) + \sum_{k=3,6,9} f_2(Ak)}{|Ak|}$ in Stage 2.

Then, we average the improved minimum hop counts from both Stage 1 and Stage 2 as follows.

- $h_{A1,U}^{optimized} = \frac{\sum_{j=1}^{2} h_{A1,U}^{optimizedj}}{2}$
- $h_{A2,U}^{optimized} = \frac{\sum_{j=1}^{2} h_{A2,U}^{optimizedj}}{2}$
- $h_{A4,U}^{optimized} = \frac{\sum_{j=1}^{2} h_{A4,U}^{optimizedj}}{2}$

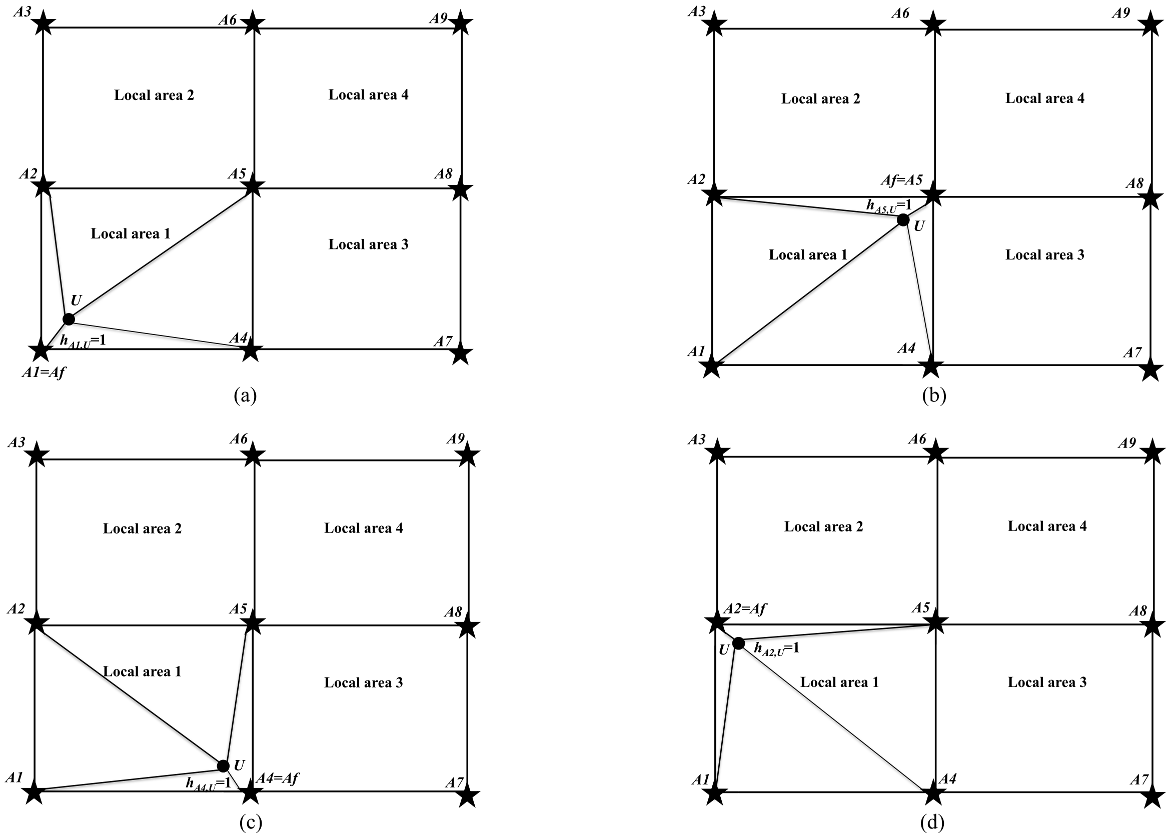


FIGURE 12. Possible locations of the unknown node. (a) The unknown node is near the corner (close to *A1*). (b) The unknown node is near the anchor in the middle of the area (close to *A5*). (c) The unknown node is near the edge (close to *A4*). (d) The unknown node is near the edge (close to *A2*).

In Fig. 12c, the minimum hop count between U and $A4$ is the smallest; thus, $Af = A4$. $A4$ is located in two local areas. The set of local anchor nodes (Ai) is $\{A1, A2, A5\}$. The steps for improvement are similar to those in the case of Fig. 12b.

- For $Ai = A1$, we obtain $h_{A1,U}^{optimized1} = \frac{\sum_{k=2,5} f_1(Ak)}{|Ak|}$ in Stage 1 by applying Eq. (9) for $0 \leq \theta_k \leq \frac{\pi}{2}$ and $h_{A1,U}^{optimized2} = \frac{\sum_{k=3,6} f_1(Ak)+\sum_{k=7,8} f_2(Ak)}{|Ak|}$ in Stage 2 by applying Eq. (10) for $\frac{3\pi}{4} \leq \theta_k \leq \pi$.
- Similarly, for $Ai = A2$, we obtain $h_{A2,U}^{optimized1} = \frac{\sum_{k=1,5} f_1(Ak)}{|Ak|}$ in Stage 1 and

$h_{A2,U}^{optimized2} = \frac{\sum_{k=3,6,8,9} f_1(Ak)+\sum_{k=7} f_2(Ak)}{|Ak|}$ in Stage 2.

- For $Ai = A5$, we obtain

$h_{A5,U}^{optimized1} = \frac{\sum_{k=1,2} f_1(Ak)}{|Ak|}$ in Stage 1 and
$h_{A5,U}^{optimized2} = \frac{\sum_{k=3,6,7,8,9} f_1(Ak)}{|Ak|}$ in Stage 2.

Then, we average the improved minimum hop counts from both Stage 1 and Stage 2 as follows.

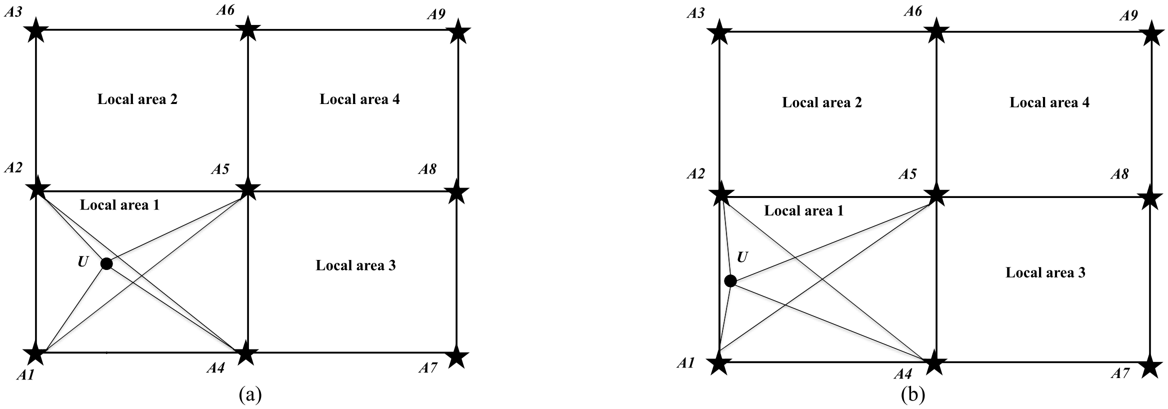
- $h_{A1,U}^{optimized} = \frac{\sum_{j=1}^{2} h_{A1,U}^{optimizedj}}{2}$
- $h_{A2,U}^{optimized} = \frac{\sum_{j=1}^{2} h_{A2,U}^{optimizedj}}{2}$
- $h_{A5,U}^{optimized} = \frac{\sum_{j=1}^{2} h_{A5,U}^{optimizedj}}{2}$

Again, the calculation of $Af = A2$ is similar to that of $Af = A4$ but different from that of $Ai = \{A1, A4, A5\}$ (see also Fig. 12d).

2) THE UNKNOWN NODE IS MORE THAN ONE HOP AWAY FROM THE LOCAL ANCHOR NODE WITH THE SMALLEST MINIMUM HOP COUNT

When there are no local anchor nodes that are one hop away from the unknown node (U), the minimum hop count improvement can be classified based on two scenarios: U is in the inner area of the local area (hereafter called the inner local area) or U is near the edge of the local area. The localization can be performed by comparing the minimum hop counts between the local anchor nodes that are opposite each other along the corresponding diagonal, as shown in Eq. (11).

Scenario I: the unknown node (U) is located near the center of the inner local area. For a local area, we apply Eq. (11) to both pairs of diagonally opposite local anchor nodes (Ai_1, $Ai'{}_1$) and (Ai_2, $Ai'{}_2$). If ($g(Ai_1, Ai'{}_1)$, $g(Ai_2, Ai'{}_2)$) is equal to (0, 0), (0, 1), or (1, 0), the unknown node (U) is located near the center of the inner local area. For example, $A1$ and $A5$ (and $A2$ and $A4$) are diagonally opposite in local area 1 in Fig. 13a. Because $(g(A1, A5), g(A2, A4)) = (1, 0)$, U is located near the center of the inner local area.



(a) (b)

FIGURE 13. **The unknown node is more than one hop away from the local anchor node with the smallest minimum hop count. (a) The unknown node is located near the center of the inner local area. (b) The unknown node is near the edge of the local area of the anchor node.**

Scenario II: the unknown node (U) is located near the edge of the local area. If a($g(Ai_1, Ai'_1)$, $g(Ai_2, Ai'_2)$) is equal to (1, 1), and U is near the edge of the local area. In Fig. 13b, ($g(A1, A5)$ and $g(A2, A4)$) = (1, 1), which implies that U is near the edge of the local area.

$$g(y) = \begin{cases} 1, & (h_{U,Ai} + h_{U,Ai'}) - h_{Ai,Ai'} > 0 \; and \; h_{U,Ai} \neq h_{U,Ai'} \\ 0, & \left(h_{U,Ai} + h_{U,Ai'}\right) - h_{Ai,Ai'} = 0 \; or \; h_{U,Ai} = h_{U,Ai'} \end{cases} \tag{11}$$

Note that Eqs. (9) and (10) are not suitable for the case in which an unknown node and a local anchor node are more than one hop away. Consequently, we adapt Theorems 1 and 2 to improve the minimum hop count between a local anchor node and an unknown node using Eqs. (12) and (13) for $\frac{3\pi}{4} \leq \theta_{AkAi'Ai} \leq \pi$.

$$f_3(x) = \begin{cases} h_{Ai,U}, \, h_{Ai',Ak} + h_{U,Ai'} == h_{U,Ak} \\ h_{Ai,U} + \alpha, \, (h_{Ai',Ak} + h_{U,Ai'}) - h_{U,Ak} == 1 \\ h_{Ai,U} - \alpha, \, h_{U,Ak} - (h_{Ai',Ak} + h_{U,Ai'}) == 1 \\ h_{Ai,U} + 2\alpha, \, (h_{Ai',Ak} + h_{U,Ai'}) - h_{U,Ak} > 1 \\ h_{Ai,U} - 2\alpha, \, h_{U,Ak} - (h_{Ai',Ak} + h_{U,Ai'}) > 1 \end{cases} \tag{12}$$

$$f_4(x) = \begin{cases} h_{Ai,U}, \, \left(h_{Ak,U} + h_{Ai,U}\right) == h_{Ak,Ai} \\ h_{Ai,U} - \alpha, \, \left(h_{Ak,U} + h_{Ai,U}\right) - h_{Ak,Ai} == 1 \\ h_{Ai,U} - 2\alpha, \, \left(h_{Ak,U} + h_{Ai,U}\right) - h_{Ak,Ai} > 1 \end{cases} \tag{13}$$

Ak is determined first. Fig. 14 shows an example of selecting Ak to improve $h_{Ai,U}$ and $h_{Ai',U}$. Specifically, if (Ai, Ai') is ($A1, A5$), then Ak is {$A6, A8, A9$}. If (Ai, Ai') is ($A2, A4$), then Ak is {$A3, A7$} based on the condition of $\frac{3\pi}{4} \leq \theta_{AkAi'Ai} \leq \pi$.

After obtaining Ak, we then estimate the minimum hop counts from Ai to U ($h^{optimized}_{Ai,U}$). If Ak is a set of more than one node, we estimate each minimum hop count by averaging all the corresponding minimum hop counts that are obtained from all nodes in the set Ak.

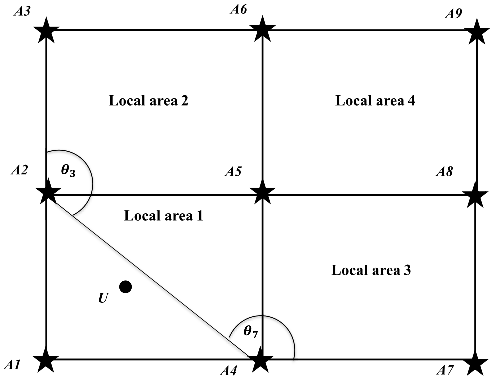


FIGURE 14. **Anchor node (*Ak*) selection for improving the minimum hop counts from *Ai* and *Ai′* to the unknown node.**

Furthermore, the minimum hop count from Ai' to the unknown node ($h^{optimized}_{Ai',U}$) can be calculated in the same manner as that for Ai. However, Ai and Ai' are opposite each other along the corresponding diagonal. Eq. (14) uses $h_{Ai',U}$, $h^{optimized}_{Ai,U}$, and $h_{Ai,U}$ to estimate $h^{optimized}_{Ai',U}$ instead of Eqs. (12) and (13).

$$h^{optimized}_{Ai',U} = h_{Ai',U} + (|h^{optimized}_{Ai,U} - h_{Ai,U}|) \tag{14}$$

Fig. 15 shows an example of determining the minimum hop count and selecting pairs of anchor nodes in both scenarios. In the first scenario, U is located near the center of the inner local area, and in the second, U is near the edge of the local area.

- In Fig. 15a, the unknown node is located near the center of the inner local area. We improve the minimum hop counts between each pair (Ai, Ai') in a similar manner to that for ($A1, A5$) and ($A2, A4$) as follows.
    - ○ For a diagonally opposite pair ($A1$, $A5$), given $Ai = A1$ and $Ai' = A5$, we improve the minimum hop count between $A1$ and U ($h_{A1,U}$) as follows.

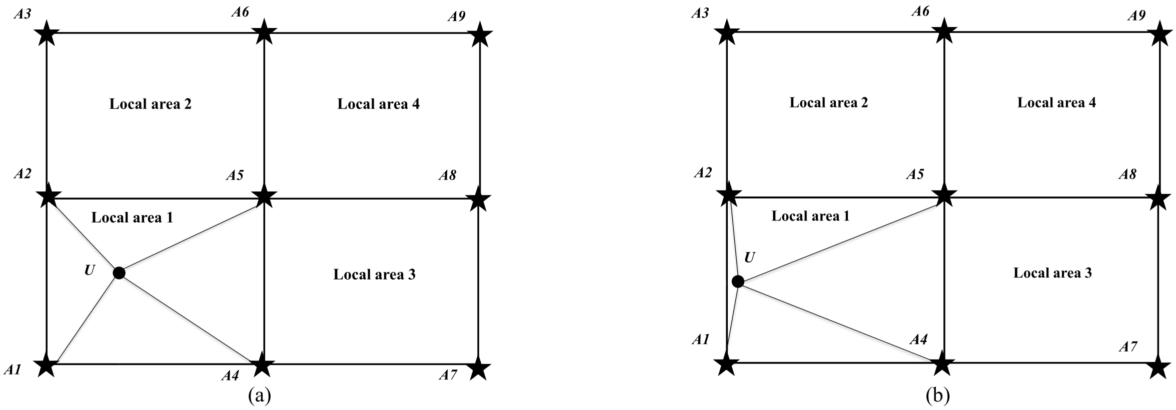


FIGURE 15. **Local anchor node pair selection for improving the minimum hop count based on the position of *U*. (a) Selection of two pairs of diagonally opposite local anchor nodes. (b) Selection of a pair of local anchor nodes on the same edge (*A1, A2*) and two pairs of diagonally opposite local anchor nodes.**

- $f_3(A6)$, $f_3(A8)$, and $f_3(A9)$ are determined using Eq. (12). Based on the condition that $\frac{3\pi}{4} \leq \theta_{AkAi'Ai} \leq \pi$, $Ak = \{A6, A8, A9\}$.
- $f_4(A6)$, $f_4(A8)$, and $f_4(A9)$ are determined using Eq. (13). Based on the condition that $\frac{3\pi}{4} \leq \theta_{AkAi'Ai} \leq \pi$, $Ak = \{A6, A8, A9\}$.

Therefore, the improved minimum hop count of $A1$ is the average of the above calculations: $h^{optimized}_{A1,U} = \frac{\sum_{k=6,8,9}(f_3(Ak)+f_4(Ak))}{2|Ak|}$. Additionally, we improve the minimum hop count of $A5$ with Eq. (14), i.e., $h^{optimized}_{A5,U} = h_{A5,U} + (|h^{optimized}_{A1,U} - h_{A1,U}|)$.

- For the pair $(A2, A4)$, given $Ai = A2$ and $Ai' = A4$, we improve the minimum hop count from $A2$ to U $(h_{A2,U})$ as follows.
  - $f_3(A3)$ and $f_3(A7)$ are determined using Eq. (12). Based on the condition that $\frac{3\pi}{4} \leq \theta_{AkAi'Ai} \leq \pi$, $Ak = \{A3, A7\}$.
  - $f_4(A3)$ and $f_4(A7)$ are determined using Eq. (13). Based on the condition that $\frac{3\pi}{4} \leq \theta_{AkAi'Ai} \leq \pi$, $Ak = \{A3, A7\}$.

Therefore, the improved minimum hop count of $A2$ is the average of the above calculations: $h^{optimized}_{A2,U} = \frac{\sum_{k=3,7}(f_3(Ak)+f_4(Ak))}{2|Ak|}$. Additionally, we improve the minimum hop count of $A4$ with Eq. (14), i.e., $h^{optimized}_{A4,U} = h_{A4,U} + (|h^{optimized}_{A2,U} - h_{A2,U}|)$.

- In Fig. 15b, the unknown node is near the edge of the local area. In this case, we improve the minimum hop counts between the pairs $(A5, A1)$ and $(A4, A2)$. However, an additional pair is located on the border, namely, $(A1, A2)$.
  - For the pair $(A5, A1)$, given $Ai = A5$ and $Ai' = A1$, we improve $h_{A5,U}$ as follows.
    - $f_3(A6)$, $f_3(A8)$, and $f_3(A9)$ are determined using Eq. (12). Based on the condition that $\frac{3\pi}{4} \leq \theta_{AkAi'Ai} \leq \pi$, $Ak = \{A6, A8, A9\}$.
    - $f_4(A6)$, $f_4(A8)$, and $f_4(A9)$ are determined using Eq. (13).

Therefore, the improved minimum hop count of $A5$ is the average of the above calculations: $h^{optimized}_{A5,U} = \frac{\sum_{k=6,8,9}(f_3(Ak)+f_4(Ak))}{2|Ak|}$.

- For the pair $(A4, A2)$, given $Ai = A4$ and $Ai' = A2$, we improve $h_{A4,U}$ as follows.
  - $f_3(A3)$ and $f_3(A7)$ are determined using Eq. (12). Based on the condition that $\frac{3\pi}{4} \leq \theta_{AkAi'Ai} \leq \pi$, $Ak = \{A3, A7\}$.
  - $f_4(A3)$ and $f_4(A7)$ are determined using Eq. (13).

Therefore, the improved minimum hop count of $A4$ is the average of the above calculations: $h^{optimized}_{A4,U} = \frac{\sum_{k=3,7}(f_3(Ak)+f_4(Ak))}{2|Ak|}$.

- For the pair $(A1, A2)$, given $Ai = A1$ and $Ai' = A2$, where Ai and Ai' are not diagonal, we improve the minimum hop count from $A1$ to U $(h_{A1,U})$ as follows.
  - $f_3(A3)$ and $f_3(A6)$ are determined using Eq. (12). Based on the condition that $\frac{3\pi}{4} \leq \theta_{AkAi'Ai} \leq \pi$, $Ak = \{A3, A6\}$.
  - $f_4(A3)$ and $f_4(A6)$ are determined using Eq. (13).

Therefore, the improved minimum hop count of $A1$ is the average of the above calculations: $h^{optimized}_{A1,U} = \frac{\sum_{k=3,6}(f_3(Ak)+f_4(Ak))}{2|Ak|}$. Additionally, we improve the minimum hop count of $A2$ with Eq. (14), i.e., $h^{optimized}_{A2,U} = h_{A2,U} + (|h^{optimized}_{A1,U} - h_{A1,U}|)$.

C. DISTANCE DETERMINATION

In Section IV.A, we specify the local areas and local anchor nodes. In Section IV.B, we determine the anchor nodes used to improve the minimum hop count. In this section, we estimate the distance between each local anchor node and the unknown node.

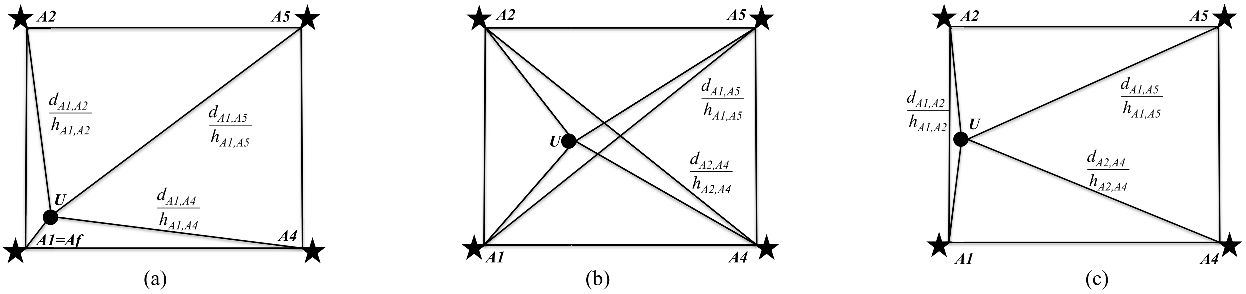

A2
A5
$\frac{d_{A1,A2}}{h_{A1,A2}}$
$\frac{d_{A1,A5}}{h_{A1,A5}}$
U
$\frac{d_{A1,A4}}{h_{A1,A4}}$
A1=Af
A4
(a)
A2
A5
$\frac{d_{A1,A5}}{h_{A1,A5}}$
U
$\frac{d_{A2,A4}}{h_{A2,A4}}$
A1
A4
(b)
A2
A5
$\frac{d_{A1,A2}}{h_{A1,A2}}$
U
$\frac{d_{A1,A5}}{h_{A1,A5}}$
$\frac{d_{A2,A4}}{h_{A2,A4}}$
A1
A4
(c)

FIGURE 16. Estimating the distance between each local node and the unknown node. (a) *U* is only one hop away from *Af*. (b) *U* is located near the center of the inner local area (*U* and *Af* are separated by more than one hop). (c) *U* is near the edge of the local area (*U* and *Af* are separated by more than one hop).

The improved minimum hop count between each local anchor node and the unknown node has been obtained. Here, we do not consider the average hop size (as in traditional DV-Hop). Instead, we define the hop size as the distance (between two local anchor nodes) divided by the corresponding (improved) minimum hop count, i.e., $hopsize_{Ai,Af} = \frac{d_{Ai,Af}}{h_{Ai,Af}}$. Then, the hop size is multiplied by the improved minimum hop count of each local anchor node to estimate the distance from the local anchor node to the unknown node ($d_{Ai,U}$).

1) THE UNKNOWN NODE AND THE LOCAL ANCHOR NODE WITH THE SMALLEST MINIMUM HOP COUNT ARE SEPARATED BY ONLY ONE HOP (SEE IV.B.1)

Fig. 16a illustrates a scenario in which the unknown node and anchor node are separated by only one hop. We estimate the following distances by multiplying the hop size between each $Ai = \{A2, A4, A5\}$ and $Af = A1$ by the corresponding improved minimum hop count.

- First, we approximate the distance from Ai to U: $d_{Ai,U} = hopsize_{Ai,Af} \times h^{optimized}_{Ai,U}$, for $i = 2, 4$, and 5.
- We then approximate the distance from Af to the unknown node ($d_{A1,U}$). Because $d_{A1,U}$ has the largest difference in distances from each anchor Ai to $A1$ and from each Ai to the unknown node, $d_{A1,U} = max(|d_{A1,A2} - d_{A2,U}|, |d_{A1,A4} - d_{A4,U}|, |d_{A1,A5} - d_{A5,U}|)$. Note that $d_{A1,U}$ covers the position of the unknown node.

2) THE UNKNOWN NODE AND THE LOCAL ANCHOR NODE WITH THE SMALLEST MINIMUM HOP COUNT ARE SEPARATED BY MORE THAN ONE HOP (SEE IV.B.2)

We estimate the distances by multiplying the hop size between two diagonally opposite anchor nodes by the corresponding improved minimum hop count. In such situations, there are two possible scenarios.

Scenario I: the unknown node (U) is located near the center of the inner local area, as shown in Fig. 16b.

Each of the following distances is equal to the product of 1) the hop size between two diagonally opposite anchor nodes $(Ai, Ai') = (A1, A5)$ or $(A2, A4)$ and 2) the improved corresponding minimum hop count $hopsize_{Ai,Ai'} = \frac{d_{Ai,Ai'}}{h_{Ai,U}+h_{Ai',U}}$.

- Approximate distance from Ai to U: $d_{Ai,U} = hopsize_{Ai,Ai'} \times h^{optimized}_{Ai,U}$
- Approximate distance from Ai' to U: $d_{Ai',U} = hopsize_{Ai,Ai'} \times h^{optimized}_{Ai',U}$
  where $i = 1, i' = 5$; and $i = 2, i' = 4$.

Scenario II: the unknown node (U) is near the edge of the local area, as shown in Fig. 16c.

Each of the following distances is equal to the product of 1) the hop size between the two local anchor nodes around the edge $(Ai, Ai') = (A1, A2)$ and the two local anchor nodes on the diagonal, namely, $(Ai, Ai') = (A1, A5)$ or $(A2, A4)$, and 2) the improved corresponding minimum hop count $hopsize_{Ai,Ai'} = \frac{d_{Ai,Ai'}}{h_{Ai,Ai'}}$.

- Approximate distance from Ai to U: $d_{Ai,U} = hopsize_{Ai,A2} \times h^{optimized}_{Ai,U}$, for $i = 1$ and 4
- Approximate distance from Ai to U: $d_{Ai,U} = hopsize_{A1,Ai} \times h^{optimized}_{Ai,U}$, for $i = 2$ and 5

Note that in the scenarios above (See C.1 and C.2), the distance is determined; however, the distance between each local anchor node and the unknown node may be overestimated, especially when it exceeds the communication range (cr).

- For the case in which the minimum hop count between the local anchor node and the unknown node is 1, the estimated distance must be in the range of $0 \leq d_{Ai,U} \leq cr$. Therefore, we must adjust the distance according to Eq. (15) so that it is different from $g(x)$ and $f(x)$. This distance correction is denoted as $q_1(x)$, where x denotes the distance between the anchor and unknown nodes.
- For the case in which the minimum hop count between the local anchor node and the unknown node is 2, the estimated distance must be in the range of $cr < d_{Ai,U} \leq 2 \times cr$. Therefore, we must adjust the distance according to Eq. (16). Note that ω is a variable indicating the upper bound of a random number with a maximum value of cr (see also Section V).

$$q_1(x) = \begin{cases} d_{Ai,U}, & 0 \leq d_{Ai,U} \leq cr \\ cr, & cr < d_{Ai,U} \end{cases} \tag{15}$$

$$q_2(x) = \begin{cases} cr + rand(0,\omega), & d_{Ai,U} < cr \\ d_{Ai,U}, & cr < d_{Ai,U} \leq 2 \times cr \\ 2 \times cr, & 2 \times cr < d_{Ai,U} \end{cases} \quad (16)$$

D. BOUNDING THE AREA FOR LOCALIZING UNKNOWN NODES

In Section IV.C, we estimate the distances between each local anchor node and the unknown node. In this section, we estimate the positions of unknown nodes in 2 steps: 1) bounding the area of the unknown node and 2) approximating the unknown node.

1) DEFINE THE BOUNDARY OF THE UNKNOWN NODE

After obtaining the distance from each local anchor node to the unknown node, we apply the bounding box [39] technique to reduce the size of the approximation area by limiting the lower and upper bounds. The rectangular area is calculated by $d_{Ai,U}$ and the local anchor node position (x_{Ai},y_{Ai}) as follows.

- Eq. (17) can be used to determine the boundaries of the bounding box (B_i) of a local anchor node (Ai) by calculating the difference in the local anchor node position (x_{Ai}, y_{Ai}) and the distance from the local anchor node to the unknown node ($d_{Ai,U}$).

$$B_i : \left[x_{Ai} - d_{Ai,U}, x_{Ai} + d_{Ai,U}\right] \times [y_{Ai} - d_{Ai,U}, y_{Ai} + d_{Ai,U}], \quad (17)$$

where $1 \leq i \leq m$. In this case, m is the number of local anchor nodes (in this paper, $m = 4$).

- Eq. (18) is used to determine the intersection area (R_j) of all bounding boxes.

$$R_j = \bigcap_{i=1}^{m} B_i \quad (18)$$

$$R_j = \bigcap_{i=1}^{m} \left[\max_{\forall i}(x_{Ai} - d_{Ai,U}), \max_{\forall i}(y_{Ai} - d_{Ai,U})\right] \times \left[\min_{\forall i}(x_{Ai} + d_{Ai,U}), \min_{\forall i}(y_{Ai} + d_{Ai,U})\right] \quad (19)$$

For simplicity, we rewrite Eq. (18) as Eq. (19), which specifies 1) the smallest R_j on the x-axis by calculating the maximum value of x_{Ai}– $d_{Ai,U}$ and 2) the largest R_j on the x-axis by calculating the minimum value of $x_{Ai}+d_{Ai,U}$. Similarly, we can apply Eq. (19) for R_j on the y-axis by calculating the maximum value of y_{Ai}– $d_{Ai,U}$ and the minimum value of $y_{Ai}+d_{Ai,U}$, as in the example in Fig. 17. The bounding box of the unknown node (R_j) is determined based on the min-max method in Eq. (19). Specifically, we use the positions of three local anchor nodes ($A1$, $A2$, and $A3$) and the distances obtained earlier ($d_{A1,U}$, $d_{A2,U}$, and $d_{A3,U}$).

2) UNKNOWN NODE LOCALIZATION BASED ON PSO

The bounding box method, which was discussed earlier, helps reduce the size of the approximation area, thereby reducing the number of PSO run cycles required to determine the unknown node position when computational resources are scarce [33].

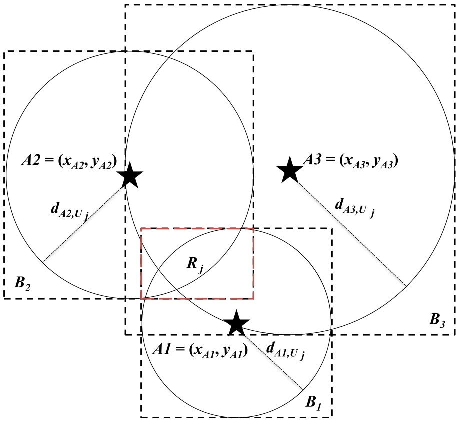


FIGURE 17. **Min-max method.**

PSO is a heuristic method that adopts social structures to search for the most appropriate solution to a specific problem. Applying PSO to calculate the sum of the distances between the unknown node and the local anchor nodes could yield more accurate results than those produced by the multitrilateration method and least-squares method [15].

Algorithm 1 shows the application of PSO for localization. First, we create random particles at the initial position (X). Then, we specify the velocity (v) and position (x, y) and determine the best position of the particle in the current round (pb). Subsequently, we determine the optimal positions of all particles (gb) from the lowest fitness value of all particles (pb). We repeat this process by specifying a new velocity and position using Eqs. (20) and (21).

Algorithm 1 PSO

```
for $i \in S$
    Randomly select position $X_i$ and velocity $v_i$ of particle $i$
    $pb_i = X_i$
end for
Update $gb = \min(F(pb))$
repeat
    for $i \in S$
        Update velocity of particle $v_i$
        Update position of particle $X_i$
        $fitnessSol = F(X_i)$
        $pb_i = X_i$ when, $fitnessSol < F(pb_i)$
        $gb = X_i$ when, $fitnessSol < F(gb)$
    end
until stopping condition
```

We must calculate the fitness function of particle X before moving it to a new location. If this new position yields a fitness value that is less than that of the previous position, we update the particle's position. The process is repeated in the same manner for all particles (gb) until the solution is obtained, the process reaches the specified maximum number of cycles, or the entire search area in the bounding box is searched.

$$v_k(t+1) = w(t)\,v_k(t) + c_1 r_1 \left(pb_k - x_k(t)\right) + c_2 r_2 (gb - x_k(t)), \tag{20}$$

$$x_k(t+1) = x_k(t) + v_k(t+1), \tag{21}$$

where $v_k(t)$ is the velocity of the k^{th} particle in the t^{th} cycle, $x_k(t)$ represents the (x, y) coordinates of the k^{th} particle in the t^{th} cycle, pb_k is the position of the best fitness value discovered by the k^{th} particle, gb is the best position that is discovered by the swarm, c_1 and c_2 are the acceleration factors, r_1 and r_2 are random numbers in the range of [0, 1], and t is the current number of iterations.

The inertial weight (w) is a particle parameter that is used to control the effect of run speed of recent iterations. The initial value of w must be large enough so that the particle position does not remain at a local minimum. Moreover, if the initial value is too small, then the convergence process might be too slow. Therefore, we should set the value of w to be initially large and decrease it in later iterations. The value of w linearly decreases as the number of iterations increases, as expressed in Eq. (22).

$$w(t) = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} \times t, \tag{22}$$

where w_{max} is the termination inertial weight, w_{min} is the initial inertial weight, and $iter_{max}$ is the maximum number of iterations.

In this paper, we improve the PSO fitness function to be more suitable for position estimation, as shown in Eq. (23). In particular, we include this factor in the form of β_i in the fitness function, which is similar to the impact of the minimum hop counts (between U and Ak) proposed by Chen and Zhang [16]. Here, β_i reflects the fact that the minimum hop counts between U and Ak affect the localization accuracy. Our intensive evaluation indicated that high minimum hop counts correspond to low localization errors.

In addition, another factor, the number of anchor nodes (Ak), is considered in the form of γ_i. The simulations suggest that a greater number of anchor nodes corresponds to higher location accuracy. From these two factors in a weighted linear relationship (μ), we derive the fitness function, such that the value is low for a high minimum hop count and/or a large number of anchor nodes. Additionally, this function minimizes the difference between the distance from each particle to the anchor nodes and the estimated distance from the anchor nodes to the unknown node.

$$F(x_k(t)) = \sum_{i=1}^{m} \left(\mu\beta_i + (1-\mu)\gamma_i\right) \left|\left(\|x_k(t) - (x_{Ai}, y_{Ai})\| - d_{Ai,U}\right)\right|, \tag{23}$$

where x_{Ai}, y_{Ai} are the coordinates of the i^{th} local anchor node, $d_{Ai,U}$ is the estimated distance between the i^{th} local anchor node and the unknown node (U), μ is an adjustment (weight) factor for the two parameters (β and γ) in the range of 0 to 1, β_i is the inverse of the minimum hop count between the i^{th} local anchor node and $U(\frac{1}{h_{Ai,U}})$ and γ_i is the inverse of the size of the other anchor nodes (Ak) for the improvement of Ai ($\frac{1}{|Ak|}$).

V. PERFORMANCE EVALUATION

In this section, we assess the performance of the proposed method (OHAR-DV-Hop) and compare it with that of other DV-Hop-based algorithms, such as iDV-Hop [15], DV-maxHop [31], Selective 3-Anchor DV-Hop [30], PSODV-Hop [16], and GA-PSODV-Hop [33].

A. SIMULATION CONFIGURATIONS

The evaluation framework is a standard configuration of the Windows 7 (64-bits) operating system running on an Intel(R) Core (TM) Quad Q8400 2.66 GHz CPU with 2048 × 2 MB of DDR-SDRAM memory and a 320 GB, 7200 RPM hard disk. For comparison, we simulate various network configurations in MATLAB with the standard library.

We adopt a logarithmic attenuation model [40] as the method of signal propagation over the distance according to the equation below.

$$P_l(d) = P_l(d_0) + 10\eta \log\left(\frac{d}{d_0}\right) + X_\sigma, \tag{24}$$

where $P_l(d)$ denotes the path loss over the distance d, d_0 is the reference distance (=1 meter), X_σ is a Gaussian random distribution function with 0 mean, and η is the path loss exponent (=4)

Two network scenarios are used to represent both small (100 ×100 m^2) and large (300 × 300 m^2) scales [15], [16], [33]. For comparison with [15], [16], [30], [31], [33], first, the dimension of the experimental area is 100 × 100 m^2, and the number of unknown nodes is varied at 150, 200, and 250 to evaluate the extent to which the sensor node density affects the accuracy. The positions of the unknown nodes are randomly determined according to a uniform distribution [30]. The positions of the anchor nodes form grids of 3 × 3, 4 × 4, and 5 × 5 nodes, and each grid is used to determine whether the number of anchor nodes influences the estimation accuracy.

In addition, after running 20 experiments, we learned that if the communication range is set to 18 meters, each sensor node has at least one connection [19], [22]. Furthermore, we increased the communication range to 22 meters [16], which is the highest radius for which the centroid technique can be applied.

Second, the area is based on a factor of 3 in the first scenario, i.e., 300 × 300 m^2 for various numbers of unknown nodes, including 150, 200, and 250. Similar to the first scenario, the positions of the unknown nodes are randomly determined according to a uniform distribution [30]. The positions of the anchor nodes form a grid with dimensions

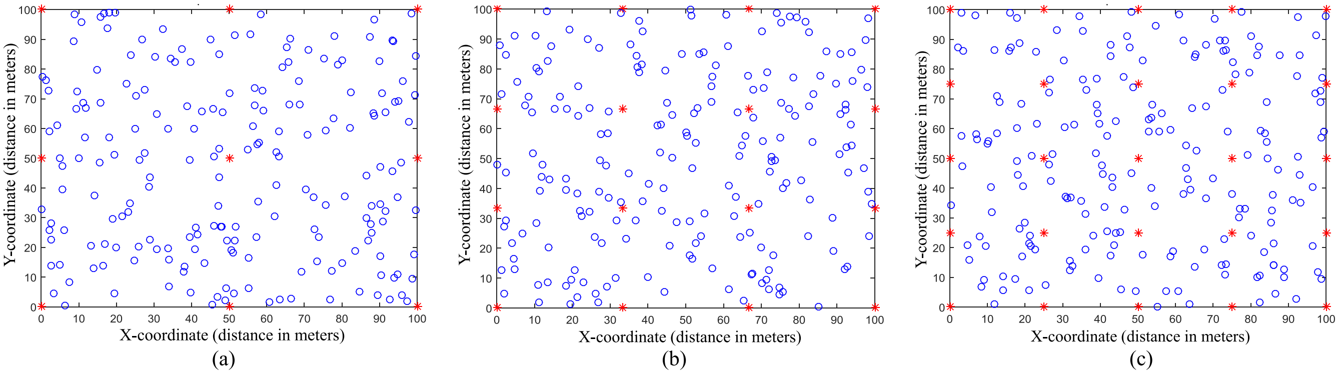


FIGURE 18. Placement of anchor nodes. (a) 9 anchor nodes. (b) 16 anchor nodes. (c) 25 anchor nodes.

of multiples of 3, including a communication range over 18 × 3 and 22 ×3 meters.

An experiment is performed to determine the effect of ω ranging from $0.1cr$ to $0.9cr$ (see Eq. (16)) to determine an optimal value for the six cases (small and large scale). Note that we observed an increasing trend for the error; thus, for conciseness, we show the error only up to $0.4cr$. These values are listed in Table 2. Fig. 18 shows examples of how the SNs are placed in the network for performance comparison purposes. In terms of PSO parameters, we follow the recommendation from the performance analysis report [16], [41]–[44] to determine the proper values (see Table 3).

TABLE 2. Simulation parameters [15], [16], [33].

Parameter	Symbol	Small scale	Large scale
Area of SNs	$L \times L$	100×100 m^2	300×300 m^2
Number of SNs	SN	N	
Number of anchors	m	9, 16, and 25 nodes	
Number of unknowns	N-m	150, 200, and 250 nodes	
Communication range	cr	18 and 22 meters	54 and 66 meters

TABLE 3. PSO parameters [16], [41]–[44].

Parameter	Value
c_1, c_2	2.00, 2.00 [44]
w_{min}, w_{max}	0.4, 0.9 [42]
$iter_{max}$	20
Particle (X)	20

We adopt the mean location error (MLE) [15], [16], [30], [31], [33] with a 95% confidence interval as the metric to measure the performance based on other related research. As shown in Eq. (25), the MLE value is the sum of the estimated location error divided by the number of unknown nodes. The lower the MLE value is, the higher the accuracy.

$$MLE = \frac{\left\| z_j^{actual} - z_j^{est} \right\|}{N - m}, \tag{25}$$

TABLE 4. Nine anchor nodes (small scale).

SNs	9 anchor nodes							
	cr = 18 m				cr = 22 m			
	0.1 cr	*0.2 cr*	*0.3 cr*	*0.4 cr*	*0.1 cr*	*0.2 cr*	*0.3 cr*	*0.4 cr*
150	3.79	3.88	4.11	4.52	3.97	4.04	4.23	4.56
200	3.21	3.21	3.21	3.22	3.60	3.61	3.61	3.61
250	3.06	3.06	3.07	3.07	3.25	3.25	3.25	3.25
Avg.	3.35	3.38	3.46	3.60	3.61	3.63	3.69	3.81

where $z_j^{actual} = (x_j^{actual}, y_j^{actual})$ denotes the actual coordinate pair of the j^{th} unknown node, $z_j^{est} = (x_j^{est}, y_j^{est})$ denotes the estimated location of the j^{th} unknown node, N denotes the number of SNs, and m denotes the number of anchor nodes.

B. SIMULATION RESULTS

Tables 4 to 11 show the MLEs for six different cases (including both small- and large-scale networks) to determine ω ranging from $0.1cr$ to $0.4cr$, with $0.1cr$ as the step size. In general, low ω values result in the best location estimation in all cases. For example, for a small-scale network with 9 anchor nodes, the location estimates with communication ranges of 18 and 22 m are 3.35 and 3.61 for $0.1cr$, 3.38 and 3.63 for $0.2cr$, 3.46 and 3.69 for $0.3cr$, and 3.60 and 3.81 for $0.4cr$, respectively. Note that the MLEs of the other two cases display the same trend, although the MLE is lower on average; for example, the values are 2.45 and 2.89 and 2.03 and 2.37 for 18- and 22-m ranges, respectively, at $0.1cr$.

With more anchor nodes, the MLEs also decrease; for example, at $0.1cr$, the MLEs are 3.35, 2.45, and 2.03, for 9, 16, and 25 nodes, respectively. Similarly, for a large-scale network, the trend holds but with higher values; for example, at $0.1cr$, for 9, 16, and 25 nodes with 54 and 66 m as the range, the MLEs are 9.99 and 10.74, 7.30 and 9.17, and 6.06 and 7.07, respectively. Additionally, as the number of anchor nodes increases, the error decreases, e.g., from 9.99 to 6.06 for 9 to 25 nodes.

Considering the first scenario (a small-scale network), Fig. 19 shows the MLE values of all algorithms when the

TABLE 5. Sixteen anchor nodes (small scale).

SNs	16 anchor nodes							
	cr = 18 m				*cr* = 22 m			
	0.1 cr	*0.2 cr*	*0.3 cr*	*0.4 cr*	*0.1 cr*	*0.2 cr*	*0.3 cr*	*0.4 cr*
150	2.73	2.81	2.97	3.30	3.47	3.64	4.01	4.57
200	2.35	2.41	2.50	2.72	2.74	2.81	2.93	3.21
250	2.27	2.29	2.34	2.47	2.47	2.48	2.48	2.48
Avg.	2.45	2.50	2.60	2.83	2.89	2.98	3.14	3.42

TABLE 6. Twenty-five anchor nodes (small scale).

SNs	25 anchor nodes							
	cr = 18 m				*cr* = 22 m			
	0.1 cr	*0.2 cr*	*0.3 cr*	*0.4 cr*	*0.1 cr*	*0.2 cr*	*0.3 cr*	*0.4 cr*
150	2.49	2.60	2.63	2.80	2.50	2.54	2.83	3.30
200	1.94	2.01	2.15	2.36	2.36	2.36	2.36	2.46
250	1.67	1.70	1.76	2.26	2.26	2.26	2.26	1.95
Avg.	2.03	2.10	2.18	2.47	2.37	2.39	2.48	2.57

TABLE 7. Nine anchor nodes (large scale).

	cr = 54 m				*cr* = 66 m			
	0.1 cr	*0.2 cr*	*0.3 cr*	*0.4 cr*	*0.1 cr*	*0.2 cr*	*0.3 cr*	*0.4 cr*
150	11.33	12.03	12.82	14.06	11.87	12.52	13.19	14.18
200	9.57	9.79	9.98	10.07	10.72	11.01	11.22	11.29
250	9.08	9.48	9.60	9.67	9.65	10.07	10.17	10.23
Avg.	9.99	10.43	10.80	11.26	10.74	11.20	11.52	11.90

TABLE 8. Sixteen anchor nodes (large scale).

SNs	16 anchor nodes							
	cr = 54 m				*cr* = 66 m			
	0.1 cr	*0.2 cr*	*0.3 cr*	*0.4 cr*	*0.1 cr*	*0.2 cr*	*0.3 cr*	*0.4 cr*
150	8.16	8.71	9.26	10.26	10.37	11.28	12.51	14.21
200	7.00	7.35	7.77	8.51	8.16	8.57	9.11	10.04
250	6.74	7.09	7.32	7.78	7.33	7.68	7.76	7.81
Avg.	7.30	7.71	8.11	8.85	8.62	9.17	9.79	10.68

TABLE 9. Twenty-five anchor nodes (large scale).

SNs	25 anchor nodes							
	cr = 54 m				*cr* = 66 m			
	0.1 cr	*0.2 cr*	*0.3 cr*	*0.4 cr*	*0.1 cr*	*0.2 cr*	*0.3 cr*	*0.4 cr*
150	7.44	8.06	8.20	8.76	7.47	7.87	8.82	10.32
200	5.78	6.13	6.68	7.38	7.03	7.19	7.33	7.69
250	4.95	5.27	5.50	6.14	6.71	7.00	7.07	7.11
Avg.	6.06	6.48	6.80	7.42	7.07	7.35	7.74	8.37

number of SNs is 150. When the number of anchor nodes increases from 9 to 25 and the communication range is 18 meters, the MLE value decreases, as illustrated in Fig. 19a.

The MLE values of iDV-Hop, DV-maxHop, Selective 3-Anchor DV-Hop, PSODV-Hop, GA-PSODV-Hop, and OHAR-DV-Hop decrease from 7.58 to 7.29, 7.27 to 6.85, 6.52 to 5.17, 5.54 to 4.26, 5.15 to 3.74, and 3.51 to 2.13, respectively. The results confirm that as the number of anchor nodes increases, the minimum hop counts between the unknown node and the anchor nodes generally decrease, thereby providing more accurate hop sizes.

Furthermore, OHAR-DV-Hop outperforms the other algorithms because the OHAR-DV-Hop algorithm can determine more accurate minimum hop counts between the unknown node and the corresponding anchor nodes (i.e., the local anchor nodes that surround the unknown node).

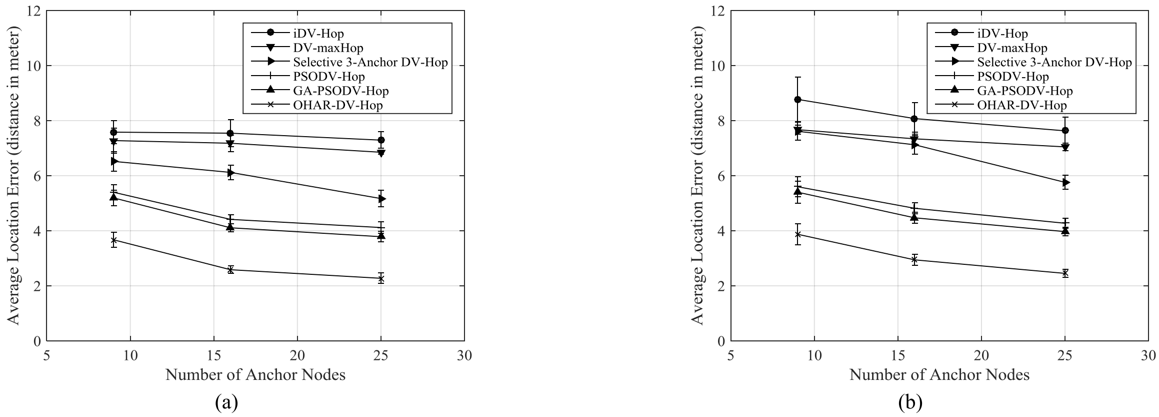
The errors of iDV-Hop are high. Improvements were obtained by replacing the multitrilateration algorithm with the least-squares method; however, the least-squares method cannot estimate the optimal position because the error between the distance from each particle to the anchor nodes and the estimated distance from the anchor nodes to the unknown node ($\|x_k(t) - (x_{Ai}, y_{Ai})\| - d_{Ai,U}$) is not normally distributed. Therefore, an optimization method, such as PSO, was introduced to minimize the errors that were obtained from the least-squares method.

DV-maxHop limits the number of anchor nodes within a specific range, so the nodes with high hop counts were not considered, nor was the hop size adjustment implemented, thereby decreasing the estimation precision. Similar to DV-maxHop, Selective 3-Anchor DV-Hop does not consider a hop size adjustment or the use of traditional multitrilateration, which can result in low accuracy.

PSODV-Hop, for example, improves the fitness function to obtain the lowest location estimation errors. However, the use of PSO alone has a limitation when determining a proper hop size. GA-PSODV-Hop improves the hop size calculation by applying the SFLA prior to position estimation based on a hybrid GA-PSO that provides high estimation accuracy. However, the error of the estimated hop size is still high when all anchor nodes are considered in the calculation.

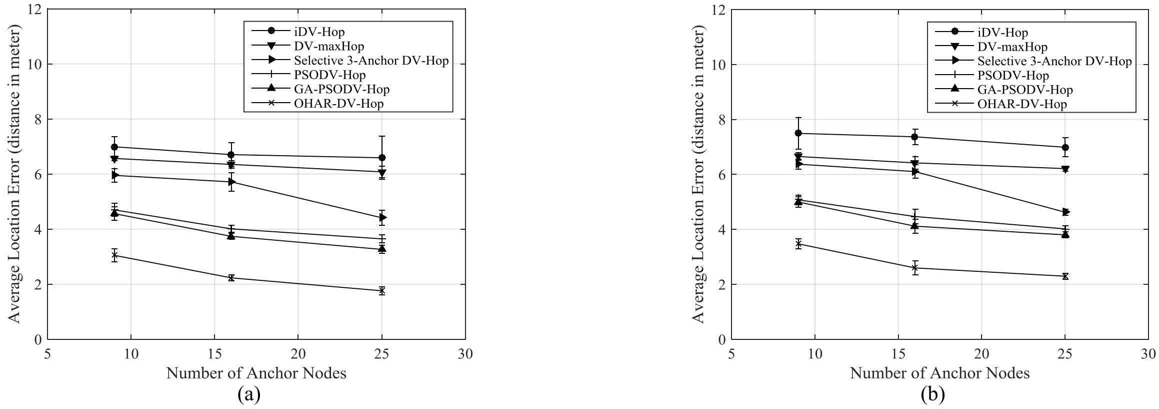
In Fig. 19b, when the communication range is increased to 22, the overall MLE values increase because the hop size also increases, thereby resulting in a decrease in the accuracy of the estimated distance between the anchor node and the unknown node. For example, when the number of anchor nodes is 25, the MLE increases from 7.29 to 7.63, 6.85 to 7.05, 5.17 to 5.76, 4.26 to 4.41, 3.74 to 3.97, and 2.13 to 2.31 for iDV-Hop, DV-maxHop, Selective 3-Anchor DV-Hop, PSODV-Hop, GA-PSODV-Hop, and OHAR-DV-Hop, respectively. OHAR-DV-Hop outperforms the other methods and yields the lowest MLE values because of the improved hop size computation process. As the range increases, the hop size error tends to increase, particularly with the averaging approach; however, OHAR-DV-Hop considers only the hop size between two connected anchor nodes associated with the unknown node; thus, regardless of the range, OHAR-DV-Hop still yields high accuracy.

Similar to Fig. 19, Fig. 20 shows the MLE values when the number of SNs is 200. Overall, the trend is similar to that in Fig. 19. The algorithms can be arranged in descending order according to the MLE as follows: iDV-Hop,



(a) (b)

FIGURE 19. MLEs for 150 sensor nodes (small scale). (a) Communication range = 18 meters. (b) Communication range = 22 meters.



(a) (b)

FIGURE 20. MLEs for 200 sensor nodes (small scale). (a) Communication range = 18 meters. (b) Communication range = 22 meters.

DV-maxHop, Selective 3-Anchor DV-Hop, PSODV-Hop, GA-PSODV-Hop, and OHAR-DV-Hop. In the case of the 18-meter communication range (Fig. 20a), the proposed OHAR-DV-Hop method provides the lowest error (from 2.87 to 1.55).

For iDV-Hop, DV-maxHop, Selective 3-Anchor DV-Hop, PSODV-Hop, and GA-PSODV-Hop, the errors decrease from 6.99 to 6.59, 6.65 to 6.08, 5.95 to 4.41, 4.91 to 3.86, and 4.53 to 3.24, respectively. Fig. 20b shows the results for the case of the 22-meter communication range. The trends are similar to those for the 18-meter case despite the higher MLEs. For example, when the number of anchor nodes is 25, the MLEs increase from 6.59 to 6.99, 6.08 to 6.21, 4.41 to 4.62, 3.86 to 4.23, 3.26 to 3.80, and 1.55 to 2.10 for iDV-Hop, DV-maxHop, Selective 3-Anchor DV-Hop, PSODV-Hop, GA-PSODV-Hop, and OHAR-DV-Hop, respectively.

Similar to Figs. 19 and 20, Fig. 21 shows the MLE values for the case of 250 SNs. As shown in Fig. 21a, when the number of anchor nodes increases, the MLEs of iDV-Hop, DV-maxHop, Selective 3-Anchor DV-Hop, PSODV-Hop, GA-PSODV-Hop, and OHAR-DV-Hop decrease from 6.93 to 6.13, 6.31 to 5.41, 5.87 to 4.27, 4.78 to 3.67, 4.43 to 3.06, and 2.65 to 1.28, respectively.

The performance of OHAR-DV-Hop is outstanding because increasing the number of anchor nodes results in more anchor nodes of interest being used to improve the minimum hop count, which leads to more precise location estimations. In addition, when the communication range is extended, the overall MLE values increase; for example, when the number of anchor nodes is 25, the MLEs increase from 6.13 to 6.96, 5.41 to 5.87, 4.27 to 4.42, 3.67 to 4.13, 3.06 to 1.98, and 1.28 to 2.32 for iDV-Hop, DV-maxHop, Selective 3-Anchor DV-Hop, PSODV-Hop, GA-PSODV-Hop, and OHAR-DV-Hop, respectively.

The results in Figs. 19 to 21 show that as the number of sensor nodes increases (from 150 to 250), the MLE values tend to decrease due to the high sensor node density, which leads to lower minimum hop counts between the SNs, thereby enabling more accurate hop size calculations.

For the second scenario (a large-scale network), Fig. 22 shows the MLE values of all algorithms with 150 SNs. When the number of anchor nodes increases from 9 to 25 and the communication range is 54 meters, the MLE value decreases, as shown in Fig. 22a.

The MLE values of iDV-Hop, DV-maxHop, Selective 3-Anchor DV-Hop, PSODV-Hop, GA-PSODV-Hop, and

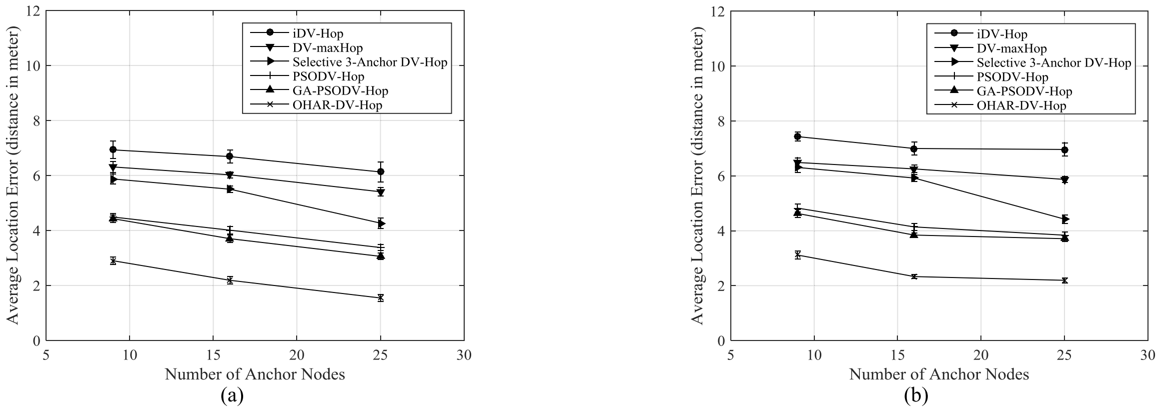


FIGURE 21. MLEs for 250 sensor nodes (small scale). (a) Communication range = 18 meters. (b) Communication range = 22 meters.

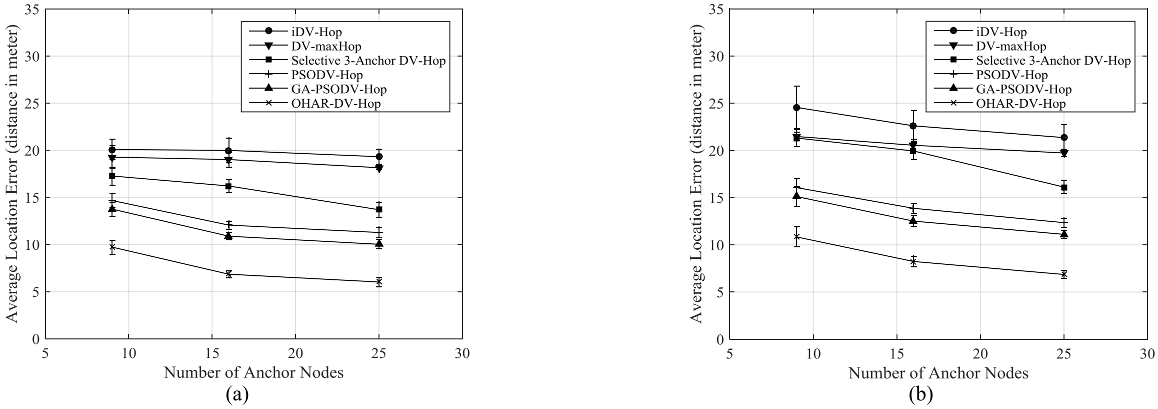


FIGURE 22. MLEs for 150 sensor nodes (large scale). (a) Communication range = 18×3 meters. (b) Communication range = 22×3 meters.

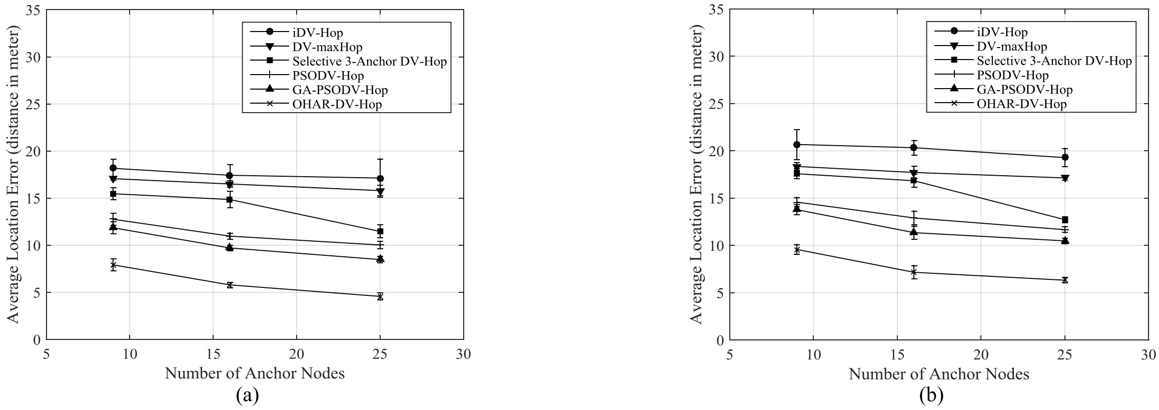


FIGURE 23. MLEs for 200 sensor nodes (large scale). (a) Communication range = 18×3 meters. (b) Communication range = 22×3 meters.

OHAR-DV-Hop decrease from 20.09 to 19.32, 19.28 to 18.15, 17.27 to 13.70, 14.67 to 11.28, 13.76 to 10.03, and 9.73 to 6.02, respectively. Here, OHAR-DV-Hop is superior, i.e., it yields the lowest MLE values.

When the communication range is increased to 66, the overall MLE values increase because the hop size also increases, thereby resulting in lower accuracy for the estimated distance between the anchor node and the unknown node. For example, Fig. 22b shows that when the number of anchor nodes is 25, the MLE increases from 19.32 to 21.36, 18.15 to 19.73, 13.70 to 16.12, 11.28 to 12.35, 10.03 to 11.11, and 6.02 to 6.86 for iDV-Hop, DV-maxHop,

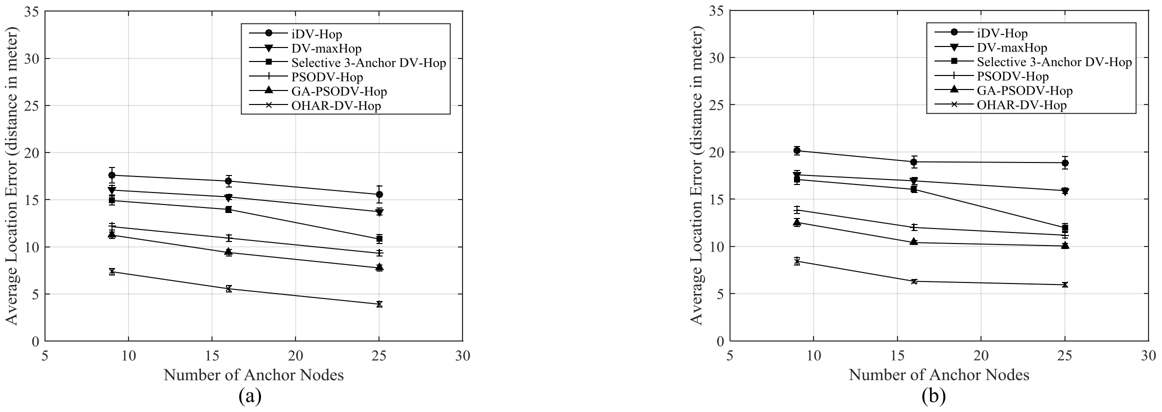


FIGURE 24. MLEs for 250 sensor nodes (large scale). (a) Communication range = 18×3 meters. (b) Communication range = 22×3 meters.

Selective 3-Anchor DV-Hop, PSODV-Hop, GA-PSODV-Hop, and OHAR-DV-Hop, respectively.

Similar to Fig. 22, Fig. 23 shows the MLE values with 200 SNs. Overall, the trend follows that in Fig. 22. The methods, in terms of descending MLE values, can be ranked as follows: iDV-Hop, DV-maxHop, Selective 3-Anchor DV-Hop, PSODV-Hop, GA-PSODV-Hop, and OHAR-DV-Hop. In the case of the 54-meter communication range (Fig. 23a), OHAR-DV-Hop yields the lowest error (from 7.92 to 4.58). For iDV-Hop, DV-maxHop, Selective 3-Anchor DV-Hop, PSODV-Hop, and GA-PSODV-Hop, the errors decrease from 18.17 to 17.13, 17.07 to 15.80, 15.47 to 11.47, 12.76 to 10.03, and 11.86 to 8.49, respectively.

For a 66-meter communication range, Fig. 23b shows that the trends are similar to those in the 54-meter case despite the higher MLEs. For example, with 25 anchor nodes, the MLEs increase from 17.13 to 19.29, 15.80 to 17.14, 11.47 to 12.73, 10.03 to 11.67, 8.49 to 10.49, and 4.58 to 6.33 for iDV-Hop, DV-maxHop, Selective 3-Anchor DV-Hop, PSODV-Hop, GA-PSODV-Hop, and OHAR-DV-Hop, respectively.

Similar to Figs. 22 and 23, Fig. 24 shows the MLE values with 250 SNs. Fig. 24a shows that when the number of anchor nodes increases, the MLEs of iDV-Hop, DV-maxHop, Selective 3-Anchor DV-Hop, PSODV-Hop, GA-PSODV-Hop, and OHAR-DV-Hop decrease from 17.60 to 15.57, 16.02 to 13.74, 14.92 to 10.84, 12.14 to 9.33, 11.25 to 7.77, and 7.36 to 3.92, respectively. The performance of OHAR-DV-Hop is outstanding. When the communication range is extended, the overall MLE values increase. For example, with 25 anchor nodes, the MLEs increase from 15.57 to 18.87, 13.74 to 15.91, 10.84 to 11.99, 9.33 to 11.20, 7.77 to 10.05, and 3.92 to 5.95 for iDV-Hop, DV-maxHop, Selective 3-Anchor DV-Hop, PSODV-Hop, GA-PSODV-Hop, and OHAR-DV-Hop, respectively.

Figs. 22 to 24 also show that as the number of SNs increases (from 150 to 250), the MLE values tend to decrease for the same reasons as observed for the small-scale network; however, for the large-scale network, the average MLEs tend to be higher because the hop size of the large network is larger (i.e., the distance between the two anchor nodes is longer) than that of the small-scale network, and therefore, the estimation error tends to be higher.

VI. CONCLUSIONS AND FUTURE WORK

One of the major problems in WSN networks is imprecise SN localization. In this study, we propose a range-free WSN position estimation approach to improve the DV-Hop technique. We suggest a strategy for calculating and using the minimum hop counts between SNs to approximate the positions of unknown nodes. The calculation of the minimum hop counts is enhanced by selecting appropriate local anchor nodes and using the angles between SNs to increase the accuracy of the approximate distances between the local anchor nodes and the unknown node. We reduce the size of the area around the unknown node (by creating a bounding box) and apply the PSO algorithm in conjunction with the enhanced distance to determine the locations of the unknown nodes.

In this research, we compare the performance of the proposed method with that of other DV-Hop-derived methods, namely, iDV-Hop, PSODV-Hop, Selective 3-Anchor DV-Hop, DV-maxHop, and GA-PSODV-Hop, using a configuration in which the locations of the anchor nodes form a grid. We vary the number of SNs and the communication range in both small- and large-scale scenarios. The experimental results show that OHAR-DV-Hop yields the best performance, followed by GA-PSODV-Hop, PSODV-Hop, Selective 3-Anchor DV-Hop, DV-maxHop, and iDV-Hop, which yield errors of 36.45%, 42.78%, 54.22%, 59.72%, and 63.49%, respectively.

Although our approach outperforms the others in terms of localization, additional investigations on the assumptions and constraints, such as QoS-aware mechanisms and data aggregation techniques, must be conducted. Moreover, other hybrid schemes and optimization techniques based on soft computing must be intensively studied in future work. In addition, various factors and configurations, such as the network density and diversity, network topology, network dimensions, and signal propagation models, may also affect the location approximation performance; therefore, comprehensive simulations and analyses should be performed. Finally, practical considerations regarding network protocols should also be investigated in future research.

In future investigations, we will focus on noise and signal fading effects during localization in a real experimental framework. We will aim to develop a WSN framework for investigating the factors that might affect the efficacy of the proposed method, in addition to corresponding analyses and simulations.

REFERENCES

[1] A. Kamilaris and A. Pitsillides, "Mobile phone computing and the Internet of Things: A survey," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 885–898, Dec. 2016.

[2] S. Ehsan and B. Hamdaoui, "A survey on energy-efficient routing techniques with QoS assurances for wireless multimedia sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 2, pp. 265–278, 2nd Quart., 2012.

[3] N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados, "Energy-efficient routing protocols in wireless sensor networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 551–591, 2nd Quart., 2013.

[4] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and Practice*. Vienna, Austria: Springer-Verlag, 1997.

[5] T. J. S. Chowdhury, C. Elkin, V. Devabhaktuni, D. B. Rawat, and J. Oluoch, "Advances on localization techniques for wireless sensor networks: A survey," *Comput. Netw.*, vol. 110, pp. 284–305, Dec. 2016.

[6] G. Han, H. Xu, T. Q. Duong, J. Jiang, and T. Hara, "Localization algorithms of wireless sensor networks: A survey," *Telecommun. Syst.*, vol. 52, no. 4, pp. 2419–2436, 2013.

[7] D. Niculescu and B. Nath, "DV based positioning in Ad Hoc networks," *Telecommun. Syst.*, vol. 22, nos. 1–4, pp. 267–280, 2003.

[8] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Pers. Commun.*, vol. 7, no. 5, pp. 28–34, Oct. 2000.

[9] S. Phoemphon, C. So-In, and T. G. Nguyen, "An enhanced wireless sensor network localization scheme for radio irregularity models using hybrid fuzzy deep extreme learning machines," *Wireless Netw.*, vol. 24, no. 3, pp. 799–819, 2018.

[10] S. Phoemphon, C. So-In, and N. Leelathakul, "Fuzzy weighted centroid localization with virtual node approximation in wireless sensor networks," *IEEE Internet Things J.*, doi: 10.1109/JIOT.2018.2811741.

[11] S. Phoemphon, C. So-In, and D. Niyato, "A hybrid model using fuzzy logic and an extreme learning machine with vector particle swarm optimization for wireless sensor network localization," *Appl. Soft Comput.*, vol. 65, pp. 101–120, Apr. 2018.

[12] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proc. Int. Conf. Mobile Comput. Netw.*, 2003, pp. 81–95.

[13] J. Lee, W. Chung, and E. Kim, "A new kernelized approach to wireless sensor network localization," *Inf. Sci.*, vol. 243, pp. 20–38, Sep. 2013.

[14] X. Yan, A. Song, Z. Yang, and W. Yang, "An improved multihop-based localization algorithm for wireless sensor network using learning approach," *Comput., Elect. Eng.*, vol. 48, pp. 247–257, Nov. 2015.

[15] S. P. Singh and S. C. Sharma, "Implementation of a PSO based improved localization algorithm for wireless sensor networks," *IETE J. Res.*, pp. 1–13, Feb. 2018.

[16] X. Chen and B. Zhang, "Improved DV-hop node localization algorithm in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 8, no. 8, pp. 1–7, 2012.

[17] 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.

[18] S. Sivakumar and R. Venkatesan, "Meta-heuristic approaches for minimizing error in localization of wireless sensor networks," *Appl. Soft Comput.*, vol. 36, pp. 506–518, Nov. 2015.

[19] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.

[20] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.

[21] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Syst. Mag.*, vol. 22, no. 3, pp. 52–67, Mar. 2002.

[22] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Eng. Fac., Dept. Comput. Eng., Erciyes Univ., Kayseri, Turkey, Tech. Rep. TR06, 2005.

[23] X. S. Yang, "Firefly algorithm," in *Nature-Inspired Metaheuristic Algorithms*. Bristol, U.K.: Luniver Press, 2010, pp. 81–96. [Online]. Available: https://dl.acm.org/citation.cfm?id=1628847

[24] R. Hamidouche, Z. Aliouat, A. M. Gueroui, A. A. A. Ari, and L. Louail, "Classical and bio-inspired mobility in sensor networks for IoT applications," *J. Netw. Comput. Appl.*, vol. 121, no. 1, pp. 70–88, 2018.

[25] C.-S. Shieh, V.-O. Sai, Y.-C. Lin, T.-F. Lee, T.-T. Nguyen, and Q.-D. Le, "Improved node localization for WSN using heuristic optimization approaches," in *Proc. 18th Int. Conf. Netw. Netw. Appl.*, 2016, pp. 95–98.

[26] 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," *Appl. Soft Comput.*, vol. 68, pp. 39–52, Jul. 2018.

[27] N. Pholdee, S. Bureerat, and A. R. Yıldız, "Hybrid real-code population-based incremental learning and differential evolution for many-objective optimisation of an automotive floor-frame," *Int. J. Vehicle Des.*, vol. 73, nos. 1–3, pp. 20–53, 2017.

[28] R. V. Kulkarni and G. K. Venayagamoorthy, "Particle swarm optimization in wireless-sensor networks: A brief survey," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 2, pp. 262–267, Mar. 2011.

[29] Y. Zhang, S. Wang, and G. Ji, "A comprehensive survey on particle swarm optimization algorithm and its applications," *Math. Problems Eng.*, vol. 2015, Feb. 2015, Art. no. 931256.

[30] L. Gui, T. Va, A. Wei, and R. Dalce, "Improvement of range-free localization technology by a novel DV-hop protocol in wireless sensor networks," *Ad Hoc Netw.*, vol. 24, pp. 55–73, Jan. 2015.

[31] F. Shahzad, T. R. Sheltami, and E. M. Shakshuki, "DV-maxHop: A fast and accurate range-free localization algorithm for anisotropic wireless networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 9, pp. 2494–2505, Sep. 2017.

[32] G. Sharma and A. Kumar, "Improved DV-Hop localization algorithm using teaching learning based optimization for wireless sensor networks," *Telecommun. Syst.*, vol. 67, no. 2, pp. 163–178, 2017.

[33] 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.

[34] S. Kumar and D. K. Lobiyal, "Novel DV-Hop localization algorithm for wireless sensor networks," *Telecommun. Syst.*, vol. 64, no. 3, pp. 509–524, 2017.

[35] S. Tomic and I. Mezei, "Improvements of DV-Hop localization algorithm for wireless sensor networks," *Telecommun. Syst.*, vol. 61, no. 1, pp. 93–106, 2016.

[36] S. Zaidi, A. El Assaf, S. Affes, and N. Kandil, "Accurate range-free localization in multi-hop wireless sensor networks," *IEEE Trans. Commun.*, vol. 64, no. 9, pp. 3886–3900, Sep. 2016.

[37] A. Al-Qaisi, A. I. Alhasanat, A. Mesleh, B. S. Sharif, C. C. Tsimenidis, and J. A. Neasham, "Quantized lower bounds on grid-based localization algorithm for wireless sensor networks," *Ann. Telecommun.*, vol. 71, pp. 239–249, Jun. 2016.

[38] K. Xu, G. Takahara, and H. Hassanein, "On the robustness of grid-based deployment in wireless sensor networks," in *Proc. 6th Int. Conf. Wireless Commun. Mobile Comput.*, 2006, pp. 1183–1188.

[39] P. Aiping, G. Xiaosong, C. Wei, and L. Haibin, "A distributed localization scheme for wireless sensor networks based on bounding box algorithm," in *Proc. 9th Int. Conf. Electron. Meas. Instrum.*, Beijing, China, 2009, pp. 984–988.

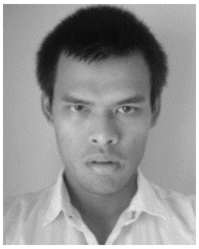
[40] S. Gu, Y. Yue, C. Maple, and C. Wu, "Fuzzy logic based localisation in wireless sensor networks for disaster environments," in *Proc. 18th Int. Conf. Autom. Comput.*, 2012, pp. 1–5.

[41] Y. Liu, X.-F. Tian, and Z.-H. Zhan, "Research on inertia weight control approaches in particle swarm optimization," *J. Nanjing Univ.*, vol. 47, no. 4, pp. 364–371, 2011.

[42] G. I. Evers, "An automatic regrouping mechanism to deal with stagnation in particle swarm optimization," M.S. thesis, Univ. Texas–Pan American, Edinburg, TX, USA, 2009.

[43] J. Wang, ‘‘Particle swarm optimization with adaptive parameter control and opposition,’’ *J. Comput. Inf. Syst.*, vol. 7, no. 12, pp. 4463–4470, 2011.

[44] Z.-H. Zhan, J. Xiao, J. Zhang, and W.-N. Chen, ‘‘Adaptive control of acceleration coefficients for particle swarm optimization based on clustering analysis,’’ in *Proc. IEEE Cong. Evol. Comput.*, Sep. 2007, pp. 3276–3282.



SONGYUT PHOEMPHON received the B.S. and M.S. degrees (Hons.) from the Department of Computer Science, Faculty of Science, Khon Kaen University, Thailand, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree. He was also an Intern at the National Electronics and Computer Technology Center, Thailand. His research interests include mobile computing, mobile and wireless sensor networks, mobile ad hoc networks, machine learning and intelligent systems, and computer network and distributed systems.

CHAKCHAI SO-IN (SM’14) he is currently a Professor with the Department of Computer Science, Khon Kaen University, Thailand. received the Ph.D. degree in computer engineering from Washington University in St. Louis, USA, in 2010. He was an Intern at CNAP/NTU, Cisco Systems, WiMAX Forum, and Bell Labs (Alcatel Lucent). His research interests include mobile computing, wireless and sensor networks, and computer networking. He has served as (Guest) Editor for *SpringerPlus*, *Mobile Information Systems*, the *IET Intelligent Transport Systems*, ECTI-CIT, and *PeerJ*, and a Committee Member for many conferences/journals, such as Globecom, ICC, VTC, WCNC, ICNP, the IEEE Transactions Wireless Communications, the IEEE Transactions on Computers, the IEEE Transactions on Vehicular Technology, the IEEE Transactions on Mobile Computing, the IEEE Transactions on Industrial Informatics, the *IEEE Communications Magazine*, and the *IEEE Communications Letters*. He has published over 85 papers in journals, such as the IEEE JSAC, *Communications Magazine, Wireless Communications Magazine, Android Application Development*, and *Computer Networks and Security Labs*, and 10 books. He is a Senior Member of the ACM.

NUTTHANON LEELATHAKUL received the B.Eng. degree (Hons.) in electrical engineering from Kasetsart University, Bangkok, Thailand, in 1999, the M.Eng. degree in electrical and computer engineering from Cornell University in 2004, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University in 2010. He has been a Faculty Member with the Faculty of Informatics, Burapha University, Chon Buri, Thailand, since 2010. His current research interests include computer networking, security in IoT, deep learning, and data analytics.