# RESEARCH

# **Open Access**

# A high-accuracy and low-energy range-free localization algorithm for wireless sensor networks



Huanqing Cui<sup>1\*</sup>, Sen Wang<sup>1</sup> and Chuanai Zhou<sup>2</sup>

\*Correspondence: cuihg@sdust.edu.cn

<sup>1</sup> College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, People's Republic of China <sup>2</sup> College of Business, Qingdao Binhai University, Qingdao, People's Republic of China

# Abstract

As the backbone of the Internet of Things, wireless sensor networks are widely applied to perceive the physical world. Most applications need to associate perception information with a position to generate physical significance. This paper proposes WRCDV-Hop, which has quadruple improvements of the well-known DV-Hop. First, the hop count between a pair of sensor nodes is measured as a continuous value rather than a discrete value. Second, the unknown nodes calculate the average distance per hop by the weighted method. Third, each sensor node only records and relays limited but sufficient beacons of the anchors. Fourth, the unknown nodes apply the whale optimization algorithm to estimate positions. The first two improvements ensure that the distance estimation between a pair of sensor nodes is highly accurate, and the third improvement reduces the energy consumption. The last improvement makes the position estimation more precise. The simulation results show that WRCDV-Hop performs well in terms of localization accuracy and energy consumption.

**Keywords:** Wireless sensor network localization, Range-free, Whale optimization algorithm, Hop count

# 1 Introduction

Wireless sensor networks (WSNs) have become one of the main methods of perceiving the physical world and are an important infrastructure component of the Internet of Things. A WSN consists of a large number of sensor nodes that communicate with each other wirelessly. It can monitor, sense and collect the required data in an area of interest [1]. WSNs have many applications, such as mine surveys, traffic and road monitoring, military deployments, underwater networks, etc. Most of these applications need to associate the information with the location to make the information meaningful.

Existing wireless sensor localization algorithms are usually classified into range-based and range-free algorithms [2]. A range-based algorithm measures the distance between sensor nodes using the received signal strength indicator (RSSI), time of arrival (TOA), angle of arrival (AOA), or time difference of arrival (TDOA), and then it establishes nonlinear equations to be solved by optimization methods. The RSSI-based localization algorithm proposed in [3] combines bird swarm optimization and a quasiaffine evolutionary algorithm and



© The Author(s) 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http:// creativeCommons.org/licenses/by/4.0/.

optimizes the node centroid location according to RSSI. SEAL [4] was designed for underwater WSNs. It utilizes a high-speed autonomous underwater vehicle (AUV) to assist in localization, and assumes that the sensor nodes and AUVs can calculate the internode and node-AUV distances. MDS-ALM [5] forms the distances between the sensor nodes as a sparse distance matrix and utilizes an augmented Lagrange matrix (ALM) filling algorithm to construct a complete distance matrix. Finally, multidimensional scaling (MDS) is applied to implement localization. DSM-MMF (a diagonal spatial mapping-based multiple signal classification and matching pursuit fusion algorithm) [6] is proposed for indoor localization. It implements AOA and TOA estimations to localize the target by using a service antenna array with the spacing of the antenna, and does not satisfy the spatial sampling theorem. The cube-based three-dimensional location system [7] uses distributed TDOA-based sensor arrays placed with a predefined method, which was designed for WSNs supporting narrow-band signal transmissions only. The range-based algorithm is constrained by the ranging technologies. RSSI is easy and inexpensive to implement, but its accuracy depends heavily on the environmental and weather conditions [8]. TOA and TDOA are more accurate than RSSI, but they depend on time-synchronized nodes, two-way ranging, or time-difference approaches provided by extra hardware, so they are expensive and affect the network's lifespan [9, 10].

The range-free algorithm only depends on the connectivity among the sensor nodes, and is easier and cheaper to implement than the range-based algorithm, so it is widely adopted in applications. ICBGLS [10] follows the geometric constraint-based scheme and considered the diameter finding algorithm of convex polygons to decrease the average localization error as well as the average execution time. PCAL [11] is based on a neural network built by the hop counts between sensor nodes and uses principal component analysis (PCA) to reduce the dimensionality of the data. CRWSNP [12] applies convex programming to estimate the original positions, and then applies multiple iterations to improve the position estimation. ITCL-PIT [13] is an improved triangle centroid localization algorithm based on the point in the triangle criterion, which reduces the overlap region by introducing two new anchors. The anchor segmentation and projection for irregular networks (ASPI) algorithm [14] are designed for irregular WSNs. It consists of anchor segmentation border construction, convex hull identification and projection-based localization.

As a well-known range-free localization algorithm, DV-Hop (distance-vector hop) [15] has attracted much attention due to its simplicity, low-cost and easy implementation, and there have been many improvements and enhancements. WHDV-Hop [16] introduces weights to the correlation matrix of the estimated distances between sensor nodes and anchors to improve accuracy. Three improvements of DV-Hop are proposed in [17]. Hereinto, iDV-Hop1 and iDV-Hop2 choose the two nearest anchors to a sensor node and the estimated position of the original DV-Hop to form three reference points, and the sensor node takes the centroid of these three points as its final estimation. Quad DV-Hop formulates the localization problem as a bounded least square problem and utilizes quadratic programming to solve it. DV-maxHop [18] introduces a control parameter *MaxHop* in the first step of DV-Hop for anisotropic WSNs, so a sensor node ignores the beacon whose hop count is greater than *MaxHop*. For aquaculture WSNs, IMDV-hop [19] combines weighted least squares and hidden terminal coupling to enhance DV-Hop. PSODV-Hop [20] estimates the positions by a two-dimensional hyperbolic algorithm and corrects the estimated position by the particle swarm optimization (PSO) algorithm. DECHDV-Hop [21] expresses the hop counts between

sensor nodes by continuous values instead of discrete values and formulates the localization problem as an optimization problem that is solved by differential evolution (DE) algorithms. The half-measure weighted centroid-based DV-Hop algorithm [22] calculates the average hop distance by the half-measure weighted centroid and applies the least square method to estimate the sensor nodes' positions. HMCS-D [23] is a hybrid DV-Hop algorithm that includes three improvements, applying a correction factor to improve the hop counts, introducing a virtual intersecting circle to estimate the hop distance and using a modified cuckoo search to estimate the sensor node positions. IR-DV-Hop [24] uses a recursive position computation and an optimization of the average distance per hop computation of anchors by exploiting a polynomial approximation. NSGA-II-DV-Hop [25] applies the non-dominated sorting genetic algorithm II (NSGA-II) to estimate the positions. The improved DV-Hop based on hop thinning and distance correction [26] corrects the minimum hop counts by RSSI, and the average hop distance by a weighted average value of the hop distance error, and estimates the distance error. For three-dimensional WSNs, N2-3DDV-Hop [27] formulates the localization problem as a multi-objective model and solves it by NSGA-II. PMDV-Hop [28] proposes a path matching algorithm to determine the optimal anchor-anchor shortest path and uses a modified PSO algorithm to optimize the initial estimated position.

This paper proposes WRCDV-Hop (weighted, RSSI-based, and continuous hop count DV-Hop), which both can improve the localization accuracy and also reduce the energy consumption. The improvements include the following:

- 1 Continuous value expressed hop counts. DV-Hop expresses the hop count between a pair of sensor nodes as an integer, which is far from the real distance. WRCDV-Hop expresses the hop count as a continuous number according to the neighbourhood relationship and RSSI.
- 2 Weighted average hop distance. The unknown nodes utilize a weighted method to calculate the average distance per hop instead of taking the first received average distance per hop as its result.
- 3 Limited number of relaying packets. When calculating the shortest paths between sensor nodes, a sensor node only receives and relays some of the anchors' packets instead of all the packets, so the number of packets to be flooded is reduced significantly, and subsequently, the energy consumption will be greatly reduced.
- 4 Improved whale optimization algorithm-based position estimation. The sensor nodes utilize the improved whale optimization algorithm (WOA), a popular swarm intelligence optimization algorithm, to estimate their positions. This method can improve the estimation accuracy.

The rest of the paper is organized as follows. Section 2 defines the localization problem and describes DV-Hop's basic steps. Section 3 presents the details of the proposed WRCDV-Hop algorithm. Section 4 presents the software simulation results and an analysis. Finally, Sect. 5 concludes the paper.

## 2 Problem statements and DV-hop algorithm

A WSN consists of a large number of sensor nodes, and each sensor node has a unique ID. The *localization* determines the positions of the unknown nodes with the help of anchors. Assume there are N sensor nodes  $\{s_1, s_2, \ldots, s_N\}$ . The first M sensor nodes are

the *anchors*, and the latter N - M sensor nodes are the *unknown nodes*. Let  $(x_i, y_i)$  be the position of sensor node  $s_i$  (i = 1, 2, ..., N), and  $(\hat{x}_i, \hat{y}_i)$  be the estimated position of unknown node  $s_i$  (i = M + 1, M + 2, ..., N). All sensor nodes has a circular radio range with a radius R. Two sensor nodes are *neighbours* if they can communicate with each other directly, i.e., their distance is less than R. Suppose the WSN is densely deployed and connected, that is, any pair of sensor nodes can communicate with each other directly or indirectly. Given two sensor nodes  $s_i$  and  $s_j$ , their hop count  $h_{ij}$  is the number of sensor nodes they pass through when they communicate, where the hop count of a pair of neighboring sensor nodes is 1. Moreover, two neighboring sensor nodes can measure RSSI, and let RSSI<sub>ij</sub> be the RSSI value from  $s_i$  to  $s_j$ .

Radio irregularity is a common phenomenon in wireless sensor networks, and we use DOI (degree of irregularity) to denote the irregularity of the radio pattern. DOI determines the communication range for 360 different directions, and it is defined as

$$K_{\theta} = \begin{cases} 1, \theta = 0\\ K_{\theta-1} \pm r \times \text{DOI}, 0 < \theta < 360 \end{cases} \text{ where } \|K_0 - K_{359}\| \le \text{DOI} \tag{1}$$

where  $K_{\theta}$  is a coefficient to represent the difference in path loss in directions  $\theta$ , and r is a random number fitting Weibull distribution. When DOI = 0, communication range is a perfect circle. As DOI increases, the communication range becomes more and more irregular.

DV-Hop [15] consists of three phases. First, each anchor broadcasts a beacon in a flooding manner, and then each sensor node calculates the shortest paths to the other sensor nodes. Second, each anchor  $s_i$  (i = 1, 2, ..., M) calculates the average distance per hop Avg Dis<sub>i</sub> using

Avg Dis<sub>i</sub> = 
$$\frac{\sum_{j=1, j \neq i}^{M} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j=1, j \neq i}^{M} h_{ij}}$$
, (2)

and it floods  $Avg Dis_i$  to the whole network. Each unknown node calculates the distances to each anchor using the first received  $Avg Dis_i$  and the corresponding hop counts. Finally, each unknown node estimates its position.

# 3 Methodology

The original DV-Hop algorithm has two main disadvantages, a low localization accuracy and a high energy consumption. The former results from the large error in calculating the distance between the sensor nodes and the position estimation algorithm. The latter results from the two message flooding operations to calculate the shortest paths and to broadcast the average distance per hop. WRCDV-Hop aims to address both of the above-mentioned disadvantages. The basic procedure of WRCDV-Hop is the same as DV-Hop as follows:

- Calculate the continuous hop-count between the sensor nodes, see 3.1
- Calculate the shortest paths between the unknown nodes and anchors, see 3.2
- Calculate the average hop distance between the sensor nodes, see 3.3

• Calculate the positions of the unknown nodes using improved WOA, see 3.4

This section presents the details of these improvements.

## 3.1 Continuous hop count calculation

As mentioned in [21], DV-Hop estimates that the distance between a pair of neighbouring sensor nodes is always R. As shown in Fig. 1, sensor nodes  $s_i$ ,  $s_j$ , and  $s_k$  are neighbours, but the distances from  $s_i$  to  $s_j$  and  $s_i$  to  $s_k$  are different. If these two distances are estimated to be R, the estimated distances between the unknown nodes and the anchors will be extremely inaccurate, subsequently leading to a high localization error. Therefore, WRCDV-Hop also represents the hop count between a pair of sensor nodes by a continuous value as [21], and it applies RSSI to make the value more accurate. The continuous hop count makes the distance estimation more accurate than the discrete hop count, and it can lead to a high localization accuracy.

Suppose  $s_i$  and  $s_j$  are a pair of neighbouring nodes, as shown in Fig. 2. The area of the radio intersection region ranges of  $s_i$  and  $s_j$  is

$$A_{ij} = 2R^2 \arccos\left(\frac{d_{ij}}{2R}\right) - \frac{d_{ij}}{2}\sqrt{4R^2 - d_{ij}^2}$$
(3)

where  $d_{ij}$  is the distance between  $s_i$  and  $s_j$ . Let NS<sub>i</sub> and NS<sub>j</sub> be the numbers of neighbouring sensor nodes of  $s_i$  and  $s_j$ , respectively. Let NS<sup>i</sup><sub>ij</sub> and NS<sup>j</sup><sub>ij</sub> be the number of common neighbour sensor nodes nearer to  $s_i$  than  $s_j$ , and nearer to  $s_j$  than  $s_i$ , respectively. Given a common neighbour sensor node  $s_k$  of  $s_i$  and  $s_j$ , if RSSI<sub>ki</sub>  $\geq$  RSSI<sub>kj</sub>,  $s_k$  is nearer to  $s_i$  than  $s_j$ . If RSSI<sub>ki</sub>  $\leq$  RSSI<sub>kj</sub>,  $s_k$  is nearer to  $s_j$  than  $s_i$ . Therefore, we can calculate NS<sup>i</sup><sub>ij</sub> and NS<sup>j</sup><sub>ij</sub> based on RSSI values. For a densely and uniformly deployed WSN, we can use the following equations to estimate  $d_{ij}$ :

$$\frac{\text{NS}_{i_j}^{\text{NS}_{i_j}}}{\text{NS}_i} = R^2 \arccos(\frac{x_1}{2R}) - \frac{x_1}{4} \sqrt{4R^2 - x_1^2}$$

$$\frac{\text{NS}_{i_j}^{\text{NS}_{i_j}}}{\text{NS}_j} = R^2 \arccos(\frac{x_1}{2R}) - \frac{x_2}{4} \sqrt{4R^2 - x_1^2}$$
(4)



Fig. 1 Node distribution model [21]



Fig. 2 The intersection model of two sensor nodes

After calculating  $x_1$  and  $x_2$  by (4), the continuous hop count between  $s_i$  and  $s_j$  is

$$h_{ij} = \frac{x_1 + x_2}{2R}$$
(5)

After deployment, each sensor node broadcasts a message including its ID. Once a sensor node receives a message, it records the corresponding ID and RSSI into a neighbour list. Next, the neighbouring sensor nodes exchange their respective neighbour lists, and the continuous hop count can be calculated by the above method.

Algorithm 1 presents the details. All the sensor nodes run this algorithm simultaneously. In this algorithm,  $Nbr_i$  represents the neighbour list of  $s_i$ . Initially,  $Nbr_i$  is empty (Line 1), and it is updated when  $s_i$  receives its neighbour's packet (Lines 3–5). When  $s_i$ cannot receive any packet, it sends  $Nbr_i$  to its neighbours (Line 7). Lines 8–23 calculate the continuous hop counts between  $s_i$  and each of its neighbours  $s_j$ . Herein, Lines 12–21 calculate  $N_{ij}^i$  and  $N_{ij}^i$ , and Line 22 calculates  $h_{ij}$  using (4) and (5).

Algorithm 1 Hop count calculation	
<b>Require:</b> a sensor node $s_i (i = 1, 2, \dots, N)$	
<b>Ensure:</b> $h_{ij}$ for each neighbor $s_j$ of $s_i$	
1: $Nbr_i \leftarrow \emptyset$	
<ol><li>broadcast its ID to its neighbors</li></ol>	
3: while receive a packet from $s_j$ do	b until no packet is received
4: $Nbr_i \leftarrow Nbr_i \cup \{(s_j, RSSI_{ji})\}$	
5: end while	
6: $NS_i \leftarrow$ number of elements in $Nbr_i$	
7: broadcast $Nbr_i$ to its neighbors	
8: for all neighbor $s_j$ of $s_i$ do	
9: $NS_j \leftarrow$ number of elements in $Nor_j$	
10: $NS_{ij}^i \leftarrow 0$	
11: $NS_{ij}^j \leftarrow 0$	
12: for all $s_k \in Nbr_i$ do	
13: if $s_k \in Nbr_j$ then	
14: if $RSSI_{ik} \ge RSSI_{jk}$ then	
15: $NS_{ij}^i \leftarrow NS_{ij}^i + 1$	
16: end if	
17: if $RSSI_{ik} \leq RSSI_{jk}$ then	
18: $NS_{ij}^j \leftarrow NS_{ij}^j + 1$	
19: end if	
20: end if	
21: end for	
22: $h_{ij} \leftarrow $ solution of (4)-(5)	
23: end for	

### 3.2 Shortest path calculation

Since the message flooding method consumes much energy and an unknown node can be localized with at least three non-colinear beacons, WRCDV-Hop does not require each sensor node to relay all the packets it receives.

Each anchor  $s_j$  (j = 1, 2, ..., M) broadcasts a beacon ( $s_j, x_j, y_j, h_{jj}$ ) to its neighbours, where  $h_{jj} = 0$  initially. Each sensor node  $s_i$  (i = 1, 2, ..., N) maintains a beacon list  $L_i = \{B_j || B_j = (s_j, x_j, y_j, h_{ij})\}$ , where  $B_j$  is the beacon from anchor  $s_j$ , and  $h_{ij}$  is the continuous hop count from  $s_j$  to  $s_i$ .  $L_i = \emptyset$  initially. When a sensor node  $s_i$  receives a beacon  $B_j$  from neighbor  $s_k$ , it updates  $B_j$  to ( $s_j, x_j, y_j, h_{ik} + h_{kj}$ ), and then it executes the following procedure:

- 1 If  $s_i$  received the beacon of  $s_j$  in previous steps, i.e., there is a beacon  $B'_j = (s'_j, x'_j, y'_j, h'_{ij})$ whose  $s'_i = s_j, x'_i = x_j, y'_i = y_j$ .
  - (a) If  $h_{ij} < h'_{ij}$ ,  $s_i$  deletes  $B'_j$  from  $L_i$  and inserts  $B_j$  into  $L_i$ . Furthermore,  $s_i$  broadcasts  $B_j$  to its neighbors.
  - (b) If  $h_{ij} > h'_{ij}$ ,  $s_i$  ignores  $B_j$ .
- 2 If  $s_i$  did not receive the beacon of  $s_j$  in previous steps, and the number of beacons in  $L_i$  is less than 3 or these beacons are colinear,  $s_i$  inserts  $B_j$  into  $L_i$  and broadcasts  $B_j$  to its neighbors. In order to determine whether the beacons of  $L_i$  are colinear, we can take any two beacons to establish a straight line and calculate the distances from the other beacons to the straight line. The beacons of  $L_i$  are colinear if all the distances are approximately equal to 0.
- 3 If  $s_i$  did not receive the beacon of  $s_j$  in previous steps, and the number of beacons in  $L_i$  is not less than 3 and these beacons are non-colinear,  $s_i$  computes  $h_{\max}$  as

$$h_{\max} = \max\{h_{il} \| h_{il} \in B_l \land B_l \in L_i\}.$$
(6)

- (a) If  $h_{ij} < h_{max}$ ,  $s_i$  inserts  $B_j$  into  $L_i$  and relays  $B_j$  to its neighbors. Furthermore,  $s_i$  removes one beacon containing  $h_{max}$  from  $L_i$ .
- (b) If  $h_{ij} > h_{max}$  and  $s_i$  ignores  $B_j$ .
- (c) If  $h_{ij} = h_{max}$ ,  $s_i$  inserts  $B_j$  into  $L_i$  and relays  $B_j$  to its neighbors.

The significance of the above process is to ensure that the number of beacons recorded and relayed by each sensor node is as few as possible but sufficient for localization. On the one hand, each sensor node can receive at least 3 beacons because the WSN is connected. On the other hand, when a sensor node has at least 3 non-colinear beacons, it ignores the beacons that are sent by the anchors further away than the current anchors, and it does not relay these beacons.

Algorithm 2 presents the details of the shortest path calculation, and all the sensor nodes run it simultaneously. In this algorithm, Lines 2–5 are executed by anchors to broadcast beacons. Lines 6–35 are the iterative process to calculate the shortest

paths. After receiving a new beacon of anchor  $s_j$  through sensor node  $s_k$  (Line 6), Line 7 calculates  $h_{ij}$ , and Line 8 initializes the *flag*. If the beacon of  $s_j$  has been received, Lines 9–19 update the records if the new  $h_{ij}$  is less than the old one, and relay the new value to its neighbours. Otherwise, Lines 20–34 append the beacon to  $L_i$  if  $L_i$  has only 1 or 2 beacons (Lines 21) or the new  $h_{ij}$  is not greater than  $h_{max}$ (Lines 26–32).

Algorithm 2 Shortest path calculation

```
Require: a sensor node s_i (i = 1, 2, \dots, N)
Ensure: beacon list L_i
1: L_i \leftarrow \emptyset
 2: if s_i is an anchor then
 3:
          h_{ii} \leftarrow 0
 4:
          broadcast (s_i, x_i, y_i, h_{ii}) to its neighbors
 5 end if
 6: while s_i receives a beacon (s_j, x_j, y_j, h_{kj}) from s_k do
          \begin{array}{c} h_{ij} \leftarrow h_{ik} + h_{kj} \\ flag \leftarrow FALSE \end{array}
 7:
 8:
          for all beacon (s_l, x_l, y_l, h_{il}) \in L_i do
if s_l = s_j then
 9:
10:
                    if h_{ij} < h_{il} then
11:
                         L_i \leftarrow L_i - \{(s_l, x_l, y_l, h_{il})\}L_i \leftarrow L_i \cup \{(s_j, x_j, y_j, h_{ij})\}
12:
13:
                         s_i relays (s_j, x_j, y_j, h_{ij}) to its neighbors flag \leftarrow TRUE
14.
15
16:
                     end if
17:
                     break
18:
                end if
           end for
19:
20:
           if flag = FALSE then
21:
                if number of elements of L_i < 3 or elements of L_i are colinear then
22:
                     L_i \leftarrow L_i \cup \{(s_j, x_j, y_j, h_{ij})\}
23:
                    s_i relays (s_j, x_j, y_j, h_{ij}) to its neighbors
24.
                else
                     calculate h_{max} using (6)
25
26:
                     if h_{ij} \leq h_{max} then
27:
                          \vec{if} \ h_{ij} < h_{max} then
28:
                              remove a beacon containing h_{max} form L_i
29:
                          end if
30:
                          L_i \leftarrow L_i \cup \{(s_j, x_j, y_j, h_{ij})\}
31:
                          s_i relays (s_j, x_j, y_j, h_{ij}) to its neighbors
32:
                     end if
33:
                end if
34
           end if
35: end while
```

### 3.3 Hop distance calculation

Suppose the beacon list of sensor node  $s_i$  is  $L_i = \{B_{i_j} || j = 1, 2, ..., K_i\}$  where  $K_i$  is the number of beacons of  $L_i$ , and  $B_{i_j} = (s_{i_j}, x_{i_j}, y_{i_j}, h_{i,i_j})$ . Because each anchor may not receive all the beacons of the other anchors, the average distance per hop of anchor  $s_i (i = 1, 2, ..., M)$  should change to

Avg Dis<sub>i</sub> = 
$$\frac{\sum_{j=1}^{K_i} \sqrt{(x_i - x_{i_j})^2 + (y_i - y_{i_j})^2}}{\sum_{j=1}^{K_i} h_{i,i_j}}.$$
(7)

After calculating Avg Dis<sub>i</sub>, anchor  $s_i$  broadcasts this value to its neighbours. Once a sensor node  $s_j$  receives an Avg Dis<sub>i</sub>, it records and relays Avg Dis<sub>i</sub> to its neighbors if and only if a beacon  $B_{j_k} \in L_j$  satisfies  $s_{j_k} = s_i$ , namely,  $s_j$  records the beacon of  $s_i$  during the shortest path calculation. The unknown node  $s_j(j = M + 1, M + 2, ..., N)$  calculates its own average distance per hop HopDis<sub>i</sub> by

$$HopDis_{j} = \sum_{k=1}^{K_{j}} w_{j_{k}} Avg Dis_{j_{k}},$$
(8)

where Avg  $\text{Dis}_{j_k}$  is the average distance per hop calculated by  $s_{j_k}(s_{j_k} \in B_k, B_k \in L_j)$ , and  $w_{j_k}$  is the weight of Avg  $\text{Dis}_{j_k}$  defined as

$$w_{j_k} = \frac{\frac{1}{h_{j_{j_k}}}}{\sum_{j=1}^{K_i} \frac{1}{h_{j_{j_k}}}}.$$
(9)

Finally, the unknown node  $s_j$  calculates its distance to anchor  $s_{j_k}$  of  $L_j$  by

$$d_{j,j_k} = \text{HopDis}_{j_k} h_{j,j_k} \tag{10}$$

# 3.4 WOA-based position estimation

WRCDV-Hop uses an improved WOA to estimate the positions of the unknown nodes. WOA mimics the hunting behavior of humpback whales [29]. Given an optimization problem in *D*-dimensional space, the WOA initializes *P* search agents. The position of the *i*th (i = 1, 2, ..., P) agent is  $\overrightarrow{Z_i}$ . Next, the WOA iteratively seeks the optimal solution. During each iteration, each agent searches for the optimal prey by encircling the prey, executing the spiral bubble-net feeding manoeuvre and searching for the prey. Taking the *i*th agent as an example, the mathematical models of these three behaviours are as follows.

1 Encircling prey. The agent updates its position to the best agent as

$$\overrightarrow{Z_i} = \overrightarrow{Z^{\star}} - \overrightarrow{A} \cdot |\overrightarrow{C} \cdot \overrightarrow{Z^{\star}} - \overrightarrow{Z_i}|, \tag{11}$$

where  $\overrightarrow{Z^{\star}}$  is the position of the best agent obtained so far, and  $\overrightarrow{C}$  and  $\overrightarrow{A}$  are coefficients defined as

$$\vec{C} = 2\vec{r_1}, \vec{A} = 2\vec{a} \cdot \vec{r_1} - \vec{a},$$
(12)

where  $\vec{a}$  is linearly decreased from 2 to 0 during iteration, and  $\vec{r_1} \in [0, 1]$  is a random vector.

2 *Bubble-net attacking*. It is modelled by two approaches. The first is the shrinking encircling mechanism, which is implemented by decreasing  $\vec{a}$  in (12). The second is to mimic the spiral movement of the whale defined as:

$$\overrightarrow{Z_i} = |\overrightarrow{Z^{\star}} - \overrightarrow{Z_i}| \cdot e^{br_2} \cdot \cos\left(2\pi r_2\right) + \overrightarrow{Z^{\star}},\tag{13}$$

where  $r_2 \in [-1, 1]$  is a random number, and *b* is a constant. The choice between these two approaches is determined by a random number  $r_3 \in [0, 1]$ : If  $r_3 < 0.5$ , it uses (11); otherwise, it uses (13).

3 Searching for prey. It is formulated as

$$\overrightarrow{Z_i} = \overrightarrow{Z_{rand}} - \overrightarrow{A} \cdot |\overrightarrow{C} \cdot \overrightarrow{Z_{rand}} - \overrightarrow{Z_i}|, \qquad (14)$$

where  $\overrightarrow{Z_{rand}}$  is the position of an agent randomly selected from the current population.

As noted in [30], the WOA has less space complexity than the other swarm optimization algorithms, and it has a strong local search ability. However, it also has some defects. It easily falls into a local optima and has insufficient global search abilities, so it does not maintain an effective balance between exploration and exploitation. We introduce opposite agents into the WOA to balance exploration and exploitation.

After updating all the agents, some of them are randomly chosen to generate the opposite agents to the best agent  $\overrightarrow{Z^{\star}}$  as

$$\overrightarrow{Z_j} = \overrightarrow{LB} + \overrightarrow{UB} - r_4 \overrightarrow{Z^*},\tag{15}$$

where  $\overrightarrow{Z_j}$  is the position of a randomly chosen agent,  $r_4$  is a random number, and  $\overrightarrow{LB}$  and  $\overrightarrow{UB}$  are respectively the lower-bound and upper-bound of the search space.

The fitness function of localizing unknown node  $s_j (j = M + 1, M + 2, ..., N)$  is defined as

$$f(x,y) = \frac{1}{K_j} \sum_{k=1}^{K_j} w_{j_k} \left( \sqrt{(x - x_{j_k})^2 + (y - y_{j_k})^2} - d_{j,j_k} \right)^2$$
(16)

Algorithm 3 presents the details of WRCDV-Hop. This algorithm is run by all the sensor nodes simultaneously. Given a sensor node  $s_i$ , it calculates the continuous hop counts and the shortest paths (Lines 1–2) first. If  $s_i$  is an anchor, it calculates and broadcasts Avg Dis<sub>i</sub>, and then they exit (Lines 3-7). If  $s_i$  is an unknown node, it continues the remaining steps. It calculates HopDis<sub>i</sub> and the distances to the anchors within  $L_i$  (Lines 8–9). Lines 10–35 utilize the improved WOA for estimation. Lines 10–11 initialize a swarm of agents and obtain their fitness, and Line 12 assigns the current best agent to  $\vec{Z^*}$ . During each iteration, Lines 14–15 calculates  $\vec{a}$ ,  $\vec{A}$  and  $\vec{C}$ . Lines 16–28 update all the agents according to the original WOA. Line 29 applies the proposed improvement. Line 30 limits all the agents within the search space. Lines 31–33 prepare for the next iteration. Finally, it returns the best coordinates  $\vec{Z^*}$  as  $(\hat{x_i}, \hat{y_i})$  (Line 35).

Algorithm 3 WRCDV-Hop algorithm **Require:** A sensor node  $s_i (i = 1, 2, \dots, N)$ **Ensure:**  $(\hat{x_i}, \hat{y_i})$  if  $s_i$  is an unknown node  $\triangleright$  position of  $s_i$ 1: Call Algorithm 1 to calculate the hop counts 2: Call Algorithm 2 to calculate the shortest paths 3: if  $s_i$  is an anchor then Calculate  $AvgDis_i$  using (7) 4. 5. Broadcast  $AvgDis_i$ 6: return 7: end if 8: Calculate  $HopDis_i$  using (8) 9: Calculate distances using (10) 10: Initialize P agents with coordinates  $\overrightarrow{Z}_i (i = 1, 2, \cdots, P)$ 11:  $f_i \leftarrow f(\overrightarrow{Z_i})$  using (16) for all agents 12:  $\overrightarrow{Z^{\star}} \leftarrow$  position of the current best agent 13: for  $t \leftarrow 1$  to  $t_{max}$  do 14:  $\overrightarrow{a} \leftarrow \left(2 - \frac{2t}{t_{max}}, 2 - \frac{2t}{t_{max}}\right)$  $\triangleright t_{max}$  is the maximum number of iteration Calculate  $\overrightarrow{A}$  and  $\overrightarrow{C}$  using (12) 15: 16: for all agents do 17.  $r \leftarrow random number in [0, 1]$ 18: if r < 0.5 then 19: if ||A|| < 1 then Update  $\overrightarrow{Z_i}$  using (11) 20: 21. else  $Z_{rand} \leftarrow$  the position of a random agent 22: 23. Update  $\overrightarrow{Z_i}$  using (14) 24: end if 25: else Update  $\overrightarrow{Z_i}$  using (13) 26: end if 27: 28. end for 29: Choose some agents randomly, and update them using (15) 30: Amend the agent beyond the search space 31:  $f_i \leftarrow f(\overrightarrow{Z_i})$  for all agents using (16)  $\overrightarrow{Z^{\star}} \leftarrow$  position of the current best agent 32: 33:  $t \leftarrow t+1$ 34: end for 35: return  $\overrightarrow{Z^{\star}}$ 

# 4 Results and discussion

The simulations are performed in MATLAB R2018b. The sensor nodes are deployed in a square region with a side length of 100 m. The simulations are performed in four different deployments, random topology, grid topology, C-random topology and C-grid topology. Figure 3 demonstrates these four different topologies, where the red stars and blue circles are the anchors and unknown nodes, respectively.

The performance of the proposed WRCDV-Hop is compared with PSODV-Hop [20], DECHDV-Hop [21] and PMDV-Hop [28]. We utilize each algorithm to localize the above four different WSNs by 100 runs, and the results are the average of these runs. For WRCDV-Hop, the parameters are b = 1 and  $t_{max} = 200$ . The parameters of the other algorithms are set according to the corresponding references.

# 4.1 Localization error analysis

#### 4.1.1 Localization error comparison

In this section, N = 100, M = 30, R = 20 m and DOI = 0.02. As Fig. 4 shows, in all the topologies, WRCDV-Hop has the lowest median, followed by PMDV-Hop, DECHDV- Hop and PSODV-Hop. Taking random topology as an example, the medians of PMDV-Hop,

![](_page_11_Figure_2.jpeg)

1.2 1.2 1.WRCDV-Hop 1.WRCDV-Hop 2.PMDV-Hop 1 1 2.PMDV-Hop 3.DECHDV-Hop 3.DECHDV-Hop localization error localization error 0.8 0.8 4.PSODV-Hop 4.PSODV-Hop 0.6 0.6 0.4 0.4 Ŧ 0.2 0.2 1 0 0 1 2 localization algorithms (a) Random topology 4 1 localization algorithms (b) Grid topology 4 1.6 0.8 +++ 1.WRCDV-Hop 1.WRCDV-Hop 1.4 2.PMDV-Hop 2.PMDV-Hop localization error 0.4 0.2 1.2 3.DECHDV-Hop 3.DECHDV-Hop localization error 4.PSODV-Hop 4.PSODV-Hop 1 ŧ 0.8 0.6 0.2 0.4 \_\_\_\_ 0.2 0 0 1 4 1 4  $2^{2}_{10}$  localization algorithms localization algorithms (c) C-random topology (d) C-grid topology

Fig. 4 Boxplots of localization errors

DECHDV-Hop and PSODV-Hop are 164.3%, 104.4% and 28.5% larger than WRCDV-Hop, respectively. For all the algorithms, the median of the C-random topology is always the largest. The interquartile ranges of all the algorithms are within 0.061 to 0.086. The first two maximum interquartile ranges, 0.086 and 0.083, are PMDV-Hop and PSODV-Hop in the grid topology. The other interquartile ranges are all less than 0.08. For all the algorithms, the number of outliers of the C-random topology is the highest. The numbers of outliers of the same, which are 2 or 3. PSODV-Hop has the largest outlier in all the topologies. Furthermore, the distributions of the localization errors in all situations are right-skewed or symmetric.

Figure 5 shows the cumulative distribution function (CDF) curves. For all the algorithms in all the topologies, 98% of the sensor nodes can be localized with a normal localization error. For example, in the random topology, WRCDV-Hop, PMDV-Hop, DECHDV- Hop and PSODV-Hop localize 98.57% of the sensor nodes with localization errors less than 0.25 m, 0.2797 m, 0.3652 m and 0.4716 m, respectively. In each topology, the distributions of the localization errors of all the algorithms are almost identical, and PSODV-Hop is always the largest, followed by DECHDV-Hop, PMDV-Hop and WRCDV-Hop.

To further compare the different algorithms, the impacts of *R*, *M*, and DOI on the average localization error, ALE, are analysed. ALE is defined as

ALE = 
$$\frac{\sum_{i=M+1}^{N} \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2}}{(N - M)R}.$$
(17)

![](_page_12_Figure_6.jpeg)

Fig. 5 CDFs of localization errors

![](_page_13_Figure_2.jpeg)

![](_page_13_Figure_3.jpeg)

(1, 1)(1, 2)(1, 2)(1, 2)(2, 3)(2, 5)(3, 3)(3, 5)(4)(5) Grid topology

■ WRCDV-Hop ■ PMDV-Hop ■ DECHDV-Hop ■ PSODV-Hop

■ WRCDV-Hop ■ PMDV-Hop ■ DECHDV-Hop ■ PSODV-Hop

![](_page_13_Figure_6.jpeg)

![](_page_13_Figure_7.jpeg)

Table 1 Average Al F	under	different	R
----------------------	-------	-----------	---

Topology	WRCDV-Hop	PMDV-Hop	DECHDV-Hop	PSODV-Hop
Random	0.130	0.174	0.246	0.337
Grid	0.111	0.154	0.180	0.279
C-random	0.230	0.310	0.390	0.873
C-grid	0.129	0.248	0.320	0.529

#### 4.1.2 Impacts of R on ALE

In this simulation, N=100, M=30, R=20 to 40 at a step of 5, and DOI = 0.02. The experimental results are shown in Table 1 and Fig. 6. The results show that WRCDV-Hop has the smallest ALE among all the algorithms. The ALE of all the algorithms decreases with an increasing *R*. As shown in Fig.6, on average, the ALE of R = 40 m improves by 49.67%, 46.75%, 67.79% and 59.44% compared with R = 20 m for all the algorithms in the random, grid, C-random and C-grid topologies, respectively. Considering the different algorithms, the ALE of R = 40 m improves by 73.04%, 64.50%, 48.03% and 38.08% compared to R = 20 m for all the topologies using WRCDV-Hop, PMDV-Hop, DECHDV-Hop and PSODV-Hop, respectively.

As R increases, a sensor node has more neighbours, so the hop counts between the sensor nodes are more accurate, leading to the distances calculated by (10) being more accurate. On the other hand, a larger R enhances the connectivity of the WSN, so each unknown node can utilize more anchors to localize itself.

#### 4.1.3 Impacts of M on ALE

In this simulation, R=20 m, M=10 to 30 at a step of 5, N=100, and DOI = 0.02. The results are shown in Table 2 and Fig. 7, which shows that the ALE decreases with an

![](_page_14_Figure_2.jpeg)

М (b) Grid topology

15

10

■ WRCDV-Hop ■ PMDV-Hop ■ DECHDV-Hop ■ PSODV-Hop

![](_page_14_Figure_5.jpeg)

![](_page_14_Figure_6.jpeg)

20

■ WRCDV-Hop ■ PMDV-Hop ■ DECHDV-Hop ■ PSODV-Hop

25

 Table 2
 Average ALE under different M

Тороlоду	WRCDV-Hop	PMDV-Hop	DECHDV-Hop	PSODV-Hop
Random	0.232	0.290	0.314	0.467
Grid	0.136	0.184	0.216	0.363
C-random	0.475	0.542	0.631	0.870
C-grid	0.291	0.354	0.424	0.756

increasing M. Compared with the other three localization algorithms, WRCDV-Hop has the smallest ALE. For all the localization algorithms, the ALE of the random and grid topologies is smaller than that of the C-random and C-grid topologies, and the ALE of the grid topology is smaller than that of the random topology. In these four topologies, WRCDV-Hop always has the highest localization accuracy, followed by PMDV-Hop, DECHDV-Hop and PSODV-Hop. As shown in Fig. 7, on average, the ALE of M = 30 improves 17.51%, 26.31%, 60.44% and 64.43% compared to M = 10 for all the algorithms in the random, grid, C-random, and C-grid topologies, respectively. Considering the different algorithms, the ALE of M = 30 improves by 42.88%, 44.61%, 43.43% and 37.77% compared to M = 10 for all the topologies using WRCDV-Hop, PMDV-Hop, DECHDV-Hop and PSODV-Hop, respectively.

As M increases, there are more anchors to assist the localization, and the average Avg Dis decreases and becomes more accurate, so the localization accuracy becomes higher.

# 4.1.4 Impacts of DOI on ALE

In this simulation, R=20 m, M=30, N=100, DOI = 0.05 to 0.25 at a step of 0.05. The results are shown in Table 3 and Fig. 8. For all the topologies, the ALE of the four algorithms increases with an increasing DOI, and the ALE of WRCDV-Hop is the smallest among all

![](_page_15_Figure_2.jpeg)

WRCDV-Hop PMDV-Hop DECHDV-Hop PSODV-Hop

![](_page_15_Figure_3.jpeg)

■ WRCDV-Hop ■ PMDV-Hop ■ DECHDV-Hop ■ PSODV-Hop

WRCDV-Hop PMDV-Hop DECHDV-Hop PSODV-Hop

■ WRCDV-Hop ■ PMDV-Hop ■ DECHDV-Hop ■ PSODV-Hop

![](_page_15_Figure_5.jpeg)

![](_page_15_Figure_6.jpeg)

## Table 3 Average ALE under different DOI

Topology	WRCDV-Hop	PMDV-Hop	DECHDV-Hop	PSODV-Hop
Random	0.358	0.440	0.466	0.556
Grid	0.349	0.404	0.454	0.538
C-random	0.375	0.430	0.473	0.642
C-grid	0.388	0.406	0.462	0.613

the algorithms in all cases. For all the algorithms and DOIs, the ALE of the random and grid topologies are smaller than those of the C-random and C-grid topologies. Moreover, the ALE of WRCDV-Hop is the least affected by DOI among the four algorithms. As shown in Fig. 8, on average, the ALE of DOI = 0.25 decreases by 14.77%, 16.60%, 15.29%, and 22.54% compared to DOI = 0.05 for all the algorithms in the random, grid, C-random, and C-grid topologies, respectively. Considering the different algorithms, the ALE of DOI = 0.25 decreases by 20.68%, 22.68%, 11.50% and 14.33% relative to DOI = 0.05 for all the topologies using WRCDV-Hop, PMDV-Hop, DECHDV-Hop and PSODV-Hop, respectively.

As DOI increases, the radio range of the sensor nodes becomes more irregular, resulting in an increasingly inaccurate hop count estimation. Moreover, the larger DOI makes the sensor nodes record beacons of the "wrong" anchors. For example, suppose  $s_i$  is an anchor, and an unknown node  $s_j$  is one of its neighbours, which means  $s_j$  can receive and record the beacon of  $s_i$ . However, when DOI becomes larger,  $s_j$  may not receive the beacon of  $s_i$ because the radio range of  $s_i$  cannot cover  $s_j$ .

#### 4.2 Energy consumption analysis

The low energy consumption requirement is one of the most important constraints on WSNs, so the localization algorithm should try to reduce the energy consumption as well as improve the localization accuracy, and then the lifetime of the WSN can be as long as

![](_page_16_Figure_2.jpeg)

Fig. 9 First order radio model [31]

possible. In DV-Hop and most of its improved algorithms, most of the energy is consumed for message communication, especially in calculating the shortest path and broadcasting the average distance per hop. To analyse the communications energy consumption during localization, we use the first-order radio model [31] shown in Fig. 9, where  $\eta$  denotes the number of bits of the packet to be transmitted, and d is the distance between the sensor nodes. The radio dissipates  $E_{elec}$  to run the transmitter or receiver circuitry for every bit and  $\epsilon_{amp}$  for the transmit amplifier. The energy consumed by transmitter to transmit a  $\eta$  bits packet at distance d is

$$E_{T_x}(\eta, d) = \eta E_{\text{elec}} + \eta d^2 \epsilon_{\text{amp}}.$$
(18)

The energy consumed by receiver to receive this message is

$$E_{R_x}(\eta) = \eta E_{\text{elec}}.$$
(19)

In this simulation, N=100, M=20 to 40 at steps of 5, R=20 m, DOI = 0.02,  $E_{\text{elec}}$ =50 nJ/bit and  $\epsilon_{amp}=100 \text{ pJ/bit/}m^2$ . The result is shown in Fig. 10. Note that the energy

![](_page_16_Figure_9.jpeg)

■ WRCDV-Hop ■ DECHDV-Hop ■ PMDV-Hop ■ PSODV-Hop

![](_page_16_Figure_11.jpeg)

![](_page_16_Figure_12.jpeg)

Fig. 10 Energy consumption

![](_page_16_Figure_14.jpeg)

Topology	WRCDV-Hop	PMDV-Hop	DECHDV-Hop	PSODV-Hop
Random	5675.6	7088.8	6515.2	8842
Grid	6280	7670.6	7147	9473.4
C-random	6062.4	7285.4	6969.6	9187.4
C-grid	6471.6	7756.2	7348.8	9665.8

 Table 4
 Average of energy consumption (Unit: nJ)

consumption of WRCDV-Hop and DECHDV-Hop contains the energy consumed during the continuous hop count calculation. Under all the topologies, the energy consumption of the four algorithms increases as the number of anchor nodes increases because more packets are broadcast and relayed with more anchors. The energy under the random and C-random topology is smaller than that of the grid and C-grid topologies. WRCDV-Hop always has the smallest energy consumption under the four topologies, followed by DECHDV-Hop, PMDV-Hop and PSODV-Hop. This can also be proven by Table 4.

For all the algorithms, the more anchors there are, the more packets there are to be received and relayed, so the energy consumption increases with the number of anchors. WRCDV-Hop and DECHDV-Hop need to broadcast additional packets to calculate the continuous hop count, so they need more energy in this phase. However, the continuous hop count leads to a more accurate hop distance estimation, so each sensor node can adjust the power to relay the packet according to the estimated hop distance, and thus, the energy consumption is reduced in calculating the shortest path and broadcasting the average distance per hop. WRCDV-Hop constrains each sensor node to only relaying the necessary packets, so the number of packets transmitted in the entire network is less than that of DECHDV-Hop; thus, WRCDV-Hop can further reduce the energy consumption.

## 4.3 Complexity performance analysis

The four algorithms utilize swarm intelligence optimizers to estimate the locations of the sensor nodes, so their complexity depends on many factors, including the size of the swarm, the number of anchors, the number of iterations and the update equations. Instead of using a theoretical analysis, the complexity of these algorithms is compared by the average execution time of localizing one sensor node.

In this section, N=100, M=20, R=20 m and DOI = 0.02. As Table 5 shows, WRCDV- Hop consumes the least time under all the topologies, followed by PMDV-Hop, DECHDV- Hop and PSODV-Hop. On average, PMDV-Hop, DECHDV-Hop and

Тороlоду	WRCDV-Hop	PMDV-Hop	DECHDV-Hop	PSODV-Hop
Random	0.2057	0.2368	0.2571	0.3402
Grid	0.2012	0.2213	0.2515	0.3353
C-random	0.2529	0.2762	0.3161	0.4215
C-grid	0.2166	0.2471	0.3031	0.4113

Table 5 Average execution time per sensor node (Unit: s)

PSODV-Hop consume 11.98%, 28.69% and 72.10% more time than WRCDV-Hop, respectively. WRCDV-Hop localizes each sensor node by a few but a sufficient number of anchors, while the other three algorithms use all the anchors. Moreover, WRCDV-Hop requires each sensor node to record and relay some of the received beacons, while the other algorithms record and relay all the beacons. These two reasons shorten the execution time of WRCDV-Hop.

The deployment of sensor nodes in the grid and C-grid topologies is regular, which speeds up the shortest path calculation, so the execution time of the random topology is longer than that of the grid topology. On average, the random topology needs 3% more time to localize a sensor node than the grid topology, and the C-random topology consumes 7% more time than the C-grid topology. Furthermore, the C-random and C-grid topologies have holes, so they need more time to calculate the shortest path, and subsequently, they consume more time than the random and grid topologies. On average, the C-random topology consumes 21.82% more time than the random topology, and the C-grid topology, and the C-grid topology.

# 5 Conclusion

This paper proposes a low-power and high-accuracy range-free localization algorithm, WRCDV-Hop, to address the deficiencies of DV-Hop. WRCDV-Hop can achieve higher localization accuracy and lower energy usage because:

- 1 It expresses the hop counts between the sensor nodes by continuous values, which is more accurate than integer values, so a sensor node can adjust its radio power while broadcasting beacons instead of always using the maximum radio power. Furthermore, the continuous hop counts make the distance estimation between the unknown nodes and anchors more accurate, which increases the localization accuracy.
- 2 It ensures that each sensor node only receives and relays the beacons of the nearest and sufficient anchors, so WRCDV-Hop consumes less energy. WRCDV-Hop uses fewer anchors to localize the unknown nodes than the other algorithms but it represents the hop counts between the sensor nodes by a continuous value, and it introduces a weighted idea in calculating the average distance per hop, so the more accurate distance estimation helps it to achieve a higher accuracy.
- 3 It improves the WOA to jump out of the local optimum, and the improved WOA outperforms PSO in PMDV-Hop and PSODV-Hop and DE in DECHDV-Hop. This further improves the localization accuracy.

The simulations under four different network topologies show that WRCDV-Hop has a higher localization accuracy and smaller energy consumption than PMDV-Hop, PSODV-Hop and DECHDV-Hop. In future, we will extends WRCDV-Hop to threedimensional sensor networks, and try to find the other better optimization algorithms to estimate the positions.

#### Abbreviations

Wireless sensor network WSN RSSI Received signal strength indicator TOA Time of arrival TDOA Time difference of arrival AUV Autonomous underwater vehicle DV-Hop Distance-vector hop WRCDV-Hop Weighted, RSSI-based, and continuous hop count DV-Hop WOA Whale optimization algorithm DOL Degree of irregularity ALE Average localization error CDF Cumulative distribution function

#### Author contributions

HC proposed the main idea and analyzed the result. He is the main writer of this paper. SW performed the simulation. CZ revised the paper. All authors read and approved the final manuscript.

#### Funding

This work is supported by the National Key R&D Program of China (Grant No. 2018YFC1406203), Natural Science Foundation of China (Grant No. 62072287), Natural Science Foundation of Shandong Province, China (Grant No. ZR2021LZH004), Shandong Postgraduate Tutor Guidance Ability Improvement Project (Grant No. SDYY17040), and Science and Technology Support Plan of Youth Innovation Team of Shandong Higher School, China (Grant No. 2019KNJ024).

#### Availability of data and materials

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

#### Declaration

#### **Competing interests**

The authors declare that they have no competing interests.

Received: 8 September 2022 Accepted: 21 April 2023 Published online: 02 May 2023

#### References

- Z. Fei, B. Li, S. Yang, C. Xing, H. Chen, L. Hanzo, A survey of multi-objective optimization in wireless sensor networks: metrics, algorithms, and open problems. IEEE Commun. Surv. Tutor. 19(1), 550–586 (2017). https://doi.org/10.1109/ COMST.2016.2610578
- Z. Lalama, S. Boulfekhar, F. Semechedine, Localization optimization in WSNs using meta-heuristics optimization algorithms: a survey. Wirel. Pers. Commun. 122(2), 1197–1220 (2022). https://doi.org/10.1007/s11277-021-08945-8
- E.M. Malathy, M. Asaithambi, A. Dheeraj, K. Arputharaj, Hybrid bird swarm optimized quasi affine algorithm based node location in wireless sensor networks. Wirel. Pers. Commun. 122(2), 947–962 (2022). https://doi.org/10.1007/ s11277-021-08934-x
- T. Ojha, S. Misra, M.S. Obaidat, SEAL: Self-adaptive AUV-based localization for sparsely deployed underwater sensor networks. Comput. Commun. 154, 204–215 (2020). https://doi.org/10.1016/j.comcom.2020.02.050
- 5. P. Wang, G. Tu, Localization algorithm of wireless sensor network based on matrix reconstruction. Comput. Commun. **154**, 216–222 (2020). https://doi.org/10.1016/j.comcom.2020.01.051
- X. Yang, Y. She, M. Zhou, W. Nie, Service antenna array-based parameter estimation for decimeter level indoor localization. IEEE Sens. Lett. 6(7), 1–4 (2022). https://doi.org/10.1109/LSENS.2022.3184348
- C.-Y. Wei, C.-C. Chen, Cube-based multitarget 3d localization using bayesian learning-based turbo decoding in wireless sensor networks. IEEE Sens. J. 22(17), 17291–17306 (2022). https://doi.org/10.1109/JSEN.2022.3193021
- 8. J. Luomala, I. Hakala, Analysis and evaluation of adaptive RSSI-based ranging in outdoor wireless sensor networks. Ad Hoc Netw. **87**, 100–112 (2019). https://doi.org/10.1016/j.adhoc.2018.10.004
- R.M. Buehrer, H. Wymeersch, R.M. Vaghefi, Collaborative sensor network localization: algorithms and practical issues. Proc. IEEE 106(6), 1089–1114 (2018). https://doi.org/10.1109/JPROC.2018.2829439
- T. Das, P.M. Khilar, R.R. Swain, Range-free intersecting chord-based geometric localization scheme for wireless sensor networks. Int. J. Commun. Syst. 33(3), 1–23 (2020). https://doi.org/10.1002/dac.4217
- S.S. Banihashemian, F. Adibnia, A new range-free PCA-based localization algorithm in wireless sensor networks. Int. J. Commun. Syst. 33(6), 1–16 (2020). https://doi.org/10.1002/dac.4291
- F. Darakeh, G.-R. Mohammad-Khani, P. Azmi, CRWSNP: cooperative range-free wireless sensor network positioning algorithm. Wirel. Netw. 24(8), 2881–2897 (2018). https://doi.org/10.1007/s11276-017-1505-2
- F.J. Mao Li, C. Pei, Improvement of triangle centroid localization algorithm based on PIT criterion (ITCL-PIT) for WSNs. EURASIP J. Wirel. Commun. Netw. 19, 1–16 (2022). https://doi.org/10.1186/s13638-022-02109-3
- J. Wang, L. Cheng, Y. Tu, S. Gu, A novel localization approach for irregular wireless sensor networks based on anchor segmentation. IEEE Sens. J. 22(7), 7267–7276 (2022). https://doi.org/10.1109/JSEN.2022.3143826
- D. Niculescu, B. Nath, DV based positioning in Ad Hoc networks. Telecommun. Syst. 22(1), 267–280 (2003). https:// doi.org/10.1023/A:1023403323460

- J. Mass-Sanchez, E. Ruiz-Ibarra, J. Cortez-González, A. Espinoza-Ruiz, L.A. Castro, Weighted hyperbolic DV-hop positioning node localization algorithm in WSNs. Wirel. Pers. Commun. 96(4), 5011–5033 (2017). https://doi.org/10.1007/ s11277-016-3727-5
- 17. S. Tomic, I. Mezei, Improvements of DV-hop localization algorithm for wireless sensor networks. Telecommun. Syst. 61(1), 93–106 (2016). https://doi.org/10.1007/s11235-015-0014-9
- F. Shahzad, T.R. Sheltami, E.M. Shakshuki, DV-maxHop: a fast and accurate range-free localization algorithm for anisotropic wireless networks. IEEE Trans. Mob. Comput. 16(9), 2494–2505 (2017). https://doi.org/10.1109/TMC.2016. 2632715
- J. Zhu, C. Lv, Z. Tao, An improved localization scheme based on IMDV-hop for large-scale wireless mobile sensor aquaculture networks. EURASIP J. Wirel. Commun. Netw. (2018). https://doi.org/10.1186/s13638-018-1190-6
- S.P. Singh, S.C. Sharma, A PSO based improved localization algorithm for wireless sensor network. Wirel. Pers. Commun. 98(1), 487–503 (2018). https://doi.org/10.1007/s11277-017-4880-1
- L. Cui, C. Xu, G. Li, Z. Ming, Y. Feng, N. Lu, A high accurate localization algorithm with DV-hop and differential evolution for wireless sensor network. Appl. Soft Comput. 68, 39–52 (2018). https://doi.org/10.1016/j.asoc.2018.03.036
- L. Jian Yin, A new distance vector-hop localization algorithm based on half-measure weighted centroid. Mob. Inf. Syst. (2019). https://doi.org/10.1155/2019/9892512
- X. Yu, M. Hu, Hop-count quantization ranging and hybrid cuckoo search optimized for DV-hop in WSNs. Wirel. Pers. Commun. 108(4), 2031–2046 (2019). https://doi.org/10.1007/s11277-019-06507-7
- S. Messous, H. Liouane, N. Liouane, Improvement of DV-hop localization algorithm for randomly deployed wireless sensor networks. Telecommun. Syst. 73, 75–86 (2020). https://doi.org/10.1007/s11235-019-00592-6
- P. Wang, F. Xue, H. Li, Z. Cui, L. Xie, J. Chen, A multi-objective DV-hop localization algorithm based on NSGA-II in Internet of Things. Mathematics (2019). https://doi.org/10.3390/math7020184
- D. Xue, Research of localization algorithm for wireless sensor network based on DV-hop. EURASIP J. Wirel. Commun. Netw. (2019). https://doi.org/10.1186/s13638-019-1539-5
- X. Cai, P. Wang, L. Du, Z. Cui, W. Zhang, J. Chen, Multi-objective three-dimensional DV-hop localization algorithm with NSGA-II. IEEE Sens. J. 19(21), 10003–10015 (2019). https://doi.org/10.1109/JSEN.2019.2927733
- 28. Q. Shi, Q. Xu, J. Zhang, An improved DV-hop scheme based on path matching and particle swarm optimization algorithm. Wirel. Pers. Commun. **104**(4), 1301–1320 (2019). https://doi.org/10.1007/s11277-018-6084-8
- S. Mirjalili, A. Lewis, The whale optimization algorithm. Adv. Eng. Softw. 95, 51–67 (2016). https://doi.org/10.1016/j. advengsoft.2016.01.008
- X. Zhang, S. Wen, Hybrid whale optimization algorithm with gathering strategies for high-dimensional problems. Expert Syst. Appl. 179, 115032 (2021). https://doi.org/10.1016/j.eswa.2021.115032
- W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, vol. 8 (2000). https://doi.org/10.1109/HICSS.2000.926982

# **Publisher's note**

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

# Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- ► Open access: articles freely available online
- ► High visibility within the field
- Retaining the copyright to your article

#### Submit your next manuscript at > springeropen.com