

Article

Enhancing the Sensor Node Localization Algorithm Based on Improved DV-Hop and DE Algorithms in Wireless Sensor Networks

Dezhi Han ^{1,†} , Yunping Yu ^{1,†}, Kuan-Ching Li ^{2,*}  and Rodrigo Fernandes de Mello ³

¹ College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China; dzhan@shmtu.edu.cn (D.H.); yu850139@163.com (Y.Y.)

² Department of Computer Science and Information Engineering, Providence University, Taichung 43301, Taiwan

³ Department of Computer Science, University of Sao Paulo, Sao Carlos, SP 13566-590, Brazil; mello@icmc.usp.br

* Correspondence: kuancli@pu.edu.tw; Tel.: +886-4-2632-8001

† These authors contributed equally to this work.

Received: 13 November 2019; Accepted: 31 December 2019; Published: 7 January 2020



Abstract: The Distance Vector-Hop (DV-Hop) algorithm is the most well-known range-free localization algorithm based on the distance vector routing protocol in wireless sensor networks; however, it is widely known that its localization accuracy is limited. In this paper, DEIDV-Hop is proposed, an enhanced wireless sensor node localization algorithm based on the differential evolution (DE) and improved DV-Hop algorithms, which improves the problem of potential error about average distance per hop. Introduced into the random individuals of mutation operation that increase the diversity of the population, random mutation is infused to enhance the search stagnation and premature convergence of the DE algorithm. On the basis of the generated individual, the social learning part of the Particle Swarm (PSO) algorithm is embedded into the crossover operation that accelerates the convergence speed as well as improves the optimization result of the algorithm. The improved DE algorithm is applied to obtain the global optimal solution corresponding to the estimated location of the unknown node. Among the four different network environments, the simulation results show that the proposed algorithm has smaller localization errors and more excellent stability than previous ones. Still, it is promising for application scenarios with higher localization accuracy and stability requirements.

Keywords: wireless sensor networks; DV-Hop; differential evolution; node localization

1. Introduction

Wireless sensor networks (WSNs) are a form of network formed by freely organizing and combining tens of thousands of sensor nodes through wireless communication technology. It is a wireless network composed of several static or mobile sensors in a self-organizing and multi-hop way, aimed at exchanging the information processed by cooperative detection of perceived objects in the coverage area of the transmission network to users [1,2]. The wireless sensor has a far-reaching impact on human life and production as it integrates sensors, micro-electro-mechanical systems, and network communication technology [3–5]. At present, it has attracted significant attention from academia and industry and has become the focus and highlight of international competition. Localization technology is an essential supporting technology in wireless sensor networks where the estimated location information is crucial. A wide range of applications that require these easy-to-deploy, low-cost sensors

well suited in WSNs includes military, agricultural, intelligent transportation, and environmental protection [6–11].

Although GPS can locate accurately, it is unrealistic to equip all micro sensor nodes with GPS in WSNs due to their high price and limited usage environment. Also, the localization accuracy of GPS in indoor and other complex environments may not be satisfactory [6]. Therefore, it is a challenging task to design an intelligent and effective localization algorithm under restricted conditions. Presently, research aims to take advantage of the interaction and connection between sensor nodes to achieve the goal of localization [12–14]. On the basis of the need to measure the distance between the actual nodes in the localization process, it can divide the localization algorithm into a range-based localization algorithm and a range-free localization algorithm [15,16]. The former needs to measure the absolute distance or azimuth between adjacent nodes and utilize the actual distance between nodes to calculate the localization of unknown nodes; such algorithms include time-of-arrival (TOA) [17], time-difference of arrival (TDOA) [18], angle of arrival (AOA) [19], and received signal strength indicator (RSSI) [20]. Regarding the latter, it is based on the network connectivity between nodes, utilizing the estimated distance between nodes to calculate the localization of nodes without measuring the actual distance between them; it includes algorithms Centroid [21], Distance Vector-Hop (DV-Hop) [22], Amorphous [23], Multi-Dimensional Scaling-programming (MDS-MAP) [24], and Approximate Point in Triangle Test (APIT) [25].

Range-free algorithms have the advantages of being low-cost, with low power consumption robust anti-measurement noise ability, and has simple hardware equipment; they can provide acceptable localization accuracy and, consequently, have attracted much attention in recent years [26]. DV-Hop, as one of the distributed localization methods based on distance vector routing and localization, is an algorithm that has attracted significant attention due to its simplicity and low equipment requirements [27].

There are two types of sensor nodes in WSNs: the anchor node with a known location, and the unknown node, the one to be located [28,29]. The purpose of the localization algorithm is to calculate the location of the unknown nodes in various ways. DV-Hop comprises three steps: the first step is that all nodes in the network get a hop-count value from the anchor nodes; next is calculated the average distance per hop for all anchor nodes; then, the estimated distance is obtained by multiplying the average distance per hop and the minimum hop-count value. Finally, the third step is to compute the localization of the unknown node according to the least-squares method [30,31], when the unknown node retrieves an estimated distance from three or more anchor nodes. The DV-Hop algorithm is easily implemented, and the localization accuracy it attains mainly depends on the estimated accuracy of average distance per hop and hop-count value between nodes. As known, the calculation error of the average distance per hop and the estimation error of hop-count value are the two primary sources of the estimation distance error.

On the basis of the abovementioned discussions, a wireless sensor node localization algorithm based on the improved DV-Hop and differential evolution (DE) algorithms is proposed in this paper, namely DEIDV-Hop. It can effectively reduce the localization error of nodes without increasing network traffic and hardware, and consists of three steps: the first two steps estimate the distance between unknown nodes and anchor nodes throughout information such as the average distance per hop and hop-count value, and the third step uses DE to determine the location of unknown nodes.

The main contributions of this paper include:

- The equation formula for calculating the average distance per hop of anchor nodes is improved. The minimum mean square error criterion more excellent to calculate the average distance per hop, and the average distance per hop of anchor nodes is obtained by applying its average value.
- DE algorithm has the defects of search stagnation and premature convergence; this improves the mutation formula and introduces random variation. In the crossover operation, the social learning part of the particle swarm optimization algorithm is introduced, and the heuristic information carried by the group optimal value is used to permit the current individual to learn from the group

optimal individual, which is then intersected with the mutation vector. With the improvement of mutation operation and crossover operation of the basic DE algorithm, the diversity of the population is enhanced as well as information of the optimal individual in the population is effectively used to raise the efficiency of the algorithm.

- To validate and show the effectiveness of the proposed algorithm, the proposed DEIDV-Hop algorithm is evaluated under four different network environments. In comparison with known algorithms, DV-Hop, PSODV-Hop [32], and GSODV-Hop [33], experimental results show that the proposed DEIDV-Hop algorithm has smaller localization errors and greater stability, yet more suitable for application scenarios with higher localization accuracy and stability requirements.

The remainder of this paper is organized as follows. In Section 2, we summarize the related work, while in Section 3 we introduce the original DV-Hop and DE algorithms. The target algorithm DEIDV-Hop is presented in Section 4, then experimental results and performance evaluation are given in Sections 5 and 6. Finally, the concluding remarks and future work direction of this paper are provided in Section 7.

2. Related Work

The range-free algorithm has several advantages: it is low cost and has low power consumption and simple hardware. DV-Hop localization algorithm has become an economical yet exceptionally important localization algorithm due to its simple algorithm, high coverage, and good feasibility that converts the estimated distance between nodes into the product of hop-count value and average distance per hop [32–34]. Even so, the disadvantage is that the localization accuracy is not good enough; accordingly, improvements in the localization accuracy are the key issues to this research. In this section, we introduce some related work of the DV-Hop localization algorithm.

Chen et al. proposed to estimate the distance of hop-count based on the value of neighbors in the same block [35]; to reduce the localization error, this study used weighted node distance to calculate the final location of nodes. Omar et al. used RSSI values to estimate the distance between the anchor node and the adjacent sensor node in one hop; such an estimated distance is applied to replace the average distance per hop in the original algorithm [31]. As a sensor node is located, it is elevated to act as an additional anchor node for subsequent use in other sensor nodes since the availability of the additional anchor node improves the accuracy of locating the remaining sensor nodes. Zhao et al. proposed a DV-Hop algorithm based on local weighted linear regression [35], where the unknown nodes attempt to estimate their locations by using nearby anchor nodes that maintain the proportional relationship between the geometric distance and the average distance per hop. The kernel method to improve the localization accuracy by increasing the weight of adjacent anchor nodes is applied.

Besides traditional optimization algorithms, many meta-heuristic algorithms to improve the localization algorithm have been considered, as they are competent to increase the accuracy of the original localization algorithm. Peng et al. proposed a genetic-based localization algorithm to improve the localization accuracy and convergence speed, limiting the feasible population region when the initial population was created. Three anchor nodes as the center to construct three squares are applied, in which the shadow area of the intersection of three squares is the overall workable area of the unknown nodes, and the initial population is generated randomly in the functional area. Singh et al. applied an improved algorithm for localization and then used particle swarm optimization to refine the results [36]. Zhao et al. proposed an improved localization algorithm based on hybrid chaotic strategy [10]. Harikrishnan et al. used the differential evolution algorithm to minimize localization error in wireless sensor network [12], while Cui et al. improved the values of hop-count by the values of common single-hop nodes between adjacent nodes and converted the discrete hop-count values into more accurate continuous values [37]. The differential evolution algorithm is introduced to achieve the optimal global solution corresponding to the estimated location of unknown nodes although it induces vast time overhead and energy consumption while improving the localization accuracy.

3. Distance Vector-Hop (DV-Hop) and Differential Evolution (DE) Algorithms

In this section, the process of the DV-Hop and DE algorithm is briefly introduced.

3.1. DV-Hop Algorithm

The three steps of the DV-Hop algorithm are discussed as follows.

Step 1: Flooding

In this step, all anchor nodes broadcast data packets with their localizations to their neighbors and hop digital segments are initially set to zero. The structure of the data packet is $\{id, x_i, y_i, h_i\}$, including identifier id , the coordinate of the anchor node i is (x_i, y_i) , h_i the minimum hop-count value from the anchor node i , and the initial value of h_i is zero. Once the neighbor node receives a packet with a smaller hop-count value from a particular anchor node, the localization of the anchor node is recorded and adds “1” to the hop-count value before sending it to other neighbor nodes [38–41]. Packets with high hop-count values are defined as obsolete information to be ignored. As soon as this step is completed, each one of the unknown nodes in the network retrieves the value of the minimum hop-count for each anchor node.

Step 2: Distance Estimation between Nodes

In this step, the distance between each node is estimated. First, it calculates the average distance per hop of each anchor node. For the anchor node, it uses Equation (1) to calculate the average distance per hop:

$$HopSize_i = \frac{\sum_{i \neq j} d_{i,j}}{\sum_{i \neq j} h_{i,j}} \quad (1)$$

where $d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, (x_i, y_i) and (x_j, y_j) represent the location of anchor nodes i and j , and $h_{i,j}$ represents the minimum hop-count value in anchor nodes i and j .

After calculating $HopSize_i$, each anchor node broadcasts its $HopSize_i$ in the system through controlled flooding. Equation (2) is used to determine the estimated distance between the unknown node and the anchor node i .

$$d_{u,i} = HopSize_i \times h_{u,i} \quad (2)$$

where $h_{u,i}$ represents the minimum value of hop-count between anchor node i and unknown node u .

Step 3: Estimate the Location of Nodes

In step 3, it is required to find out the location of all unknown nodes. For the unknown node u , the multilateration method [38] is used to estimate its location.

Let (x_u, y_u) be the location of the unknown node u and (x_i, y_i) be the location of the anchor node, where $i = 1, 2, \dots, m$. Therefore, the distance between unknown nodes u and m anchor nodes is specified by the following equation:

$$\begin{cases} (x_1 - x_u)^2 + (y_1 - y_u)^2 = d_1^2 \\ (x_2 - x_u)^2 + (y_2 - y_u)^2 = d_2^2 \\ \dots\dots\dots \\ (x_m - x_u)^2 + (y_m - y_u)^2 = d_m^2 \end{cases} \quad (3)$$

Equation (3) can be extended to:

$$\begin{cases} x_1^2 - x_m^2 - 2(x_1 - x_m)x_u + y_1^2 - y_m^2 - 2(y_1 - y_m)y_u = d_1^2 - d_m^2 \\ x_2^2 - x_m^2 - 2(x_2 - x_m)x_u + y_2^2 - y_m^2 - 2(y_2 - y_m)y_u = d_2^2 - d_m^2 \\ \dots\dots\dots \\ x_{m-1}^2 - x_m^2 - 2(x_{m-1} - x_m)x_u + y_{m-1}^2 - y_m^2 - 2(y_{m-1} - y_m)y_u = d_{m-1}^2 - d_m^2 \end{cases} \quad (4)$$

The above Equation (4) can be expressed in matrix form as $AX = B$, where A , X and B are expressed by Equations (5)–(7), respectively.

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

$$B = \begin{bmatrix} x_1^2 - x_m^2 + y_1^2 - y_m^2 + d_m^2 - d_1^2 \\ x_2^2 - x_m^2 + y_2^2 - y_m^2 + d_m^2 - d_2^2 \\ \dots\dots\dots \\ x_{m-1}^2 - x_m^2 + y_{m-1}^2 - y_m^2 + d_m^2 - d_{m-1}^2 \end{bmatrix} \quad (6)$$

$$X = \begin{bmatrix} x_u \\ y_u \end{bmatrix} \quad (7)$$

It utilizes the least square method to solve the equation. The coordinate location of the unknown node u is calculated:

$$X = (A^T A)^{-1} A^T B \quad (8)$$

3.2. Differential Evolution Algorithm

Differential evolution (DE) algorithm is a new evolutionary computing technology [13], designed to keep the population-based global search strategy of the evolutionary algorithm utilizing real coding, i.e., a simple mutation operation based on difference and one-to-one elimination mechanism to update the population that reduces the complexity of genetic algorithm operation.

3.2.1. Initialization

DE uses NP D -dimensional real-valued parameter vectors as the population of each generation, and the t -th generation is denoted as $x_i^t = [x_{i,1}^t, x_{i,2}^t, \dots, x_{i,D}^t]$. Let the boundary of the parameter variable be $x_j^{(L)} < x_j < x_j^{(U)}$, the initial value of the i -th individual is generated by Equation (9).

$$x_{i,j}^0 = x_j^{(L)} + (x_j^{(U)} - x_j^{(L)}) * rand[0, 1] \quad (9)$$

where $rand[0, 1]$ represents a uniform random number generated between $[0, 1]$, $i = 1, 2, \dots, NP$, $j = 1, 2, \dots, D$.

3.2.2. Mutation

For each target individual x_i^t , where $i = 1, 2, \dots, NP$, its mutant v_i^{t+1} is generated by randomly select individuals that are used as the parent basis vector (rand) and a difference vector is used to generate the mutant individuals:

$$v_i^{t+1} = x_{r_1}^t + F(x_{r_2}^t - x_{r_3}^t) \quad (10)$$

Among them, r_1 , r_2 and r_3 represent random numbers and different unequal natural numbers in the range $[1, NP]$. $F \in [0, 2]$, as F represents the scaling factor.

3.2.3. Crossover

The target individual x_i^t and variant individuals v_i^{t+1} are crossed to generate an experimental individual $u_i^{t+1} = [u_{i,1}^{t+1}, u_{i,2}^{t+1}, \dots, u_{i,D}^{t+1}]$. The crossover process is shown in Equation (11).

$$u_{i,j}^{t+1} = \begin{cases} v_{i,j}^{t+1}, & \text{if } r_j \leq CR \text{ or } j = rnbr_i \\ x_{i,j}^t, & \text{otherwise} \end{cases} \quad (11)$$

where r_j is the j -th random number between $[0, 1]$; $rnbr_i$ is the random natural number between $[1, D]$; CR is the crossover rate, $CR \in [0, 1]$.

3.2.4. Selection

DE uses the greedy operation to select between x_i^t and u_i^{t+1} , and generate the next generation of the individual x_i^{t+1} :

$$x_i^{t+1} = \begin{cases} u_i^{t+1}, & \text{if } f(u_i^{t+1}) < f(x_i^t) \\ x_i^t, & \text{otherwise} \end{cases} \quad (12)$$

where $f(\cdot)$ is the objective function. Given that the optimization problem in this paper is in minimization form, the selection condition is $f(u_i^{t+1}) < f(x_i^t)$. The overall flow chart of the DE algorithm is shown in Figure 1.

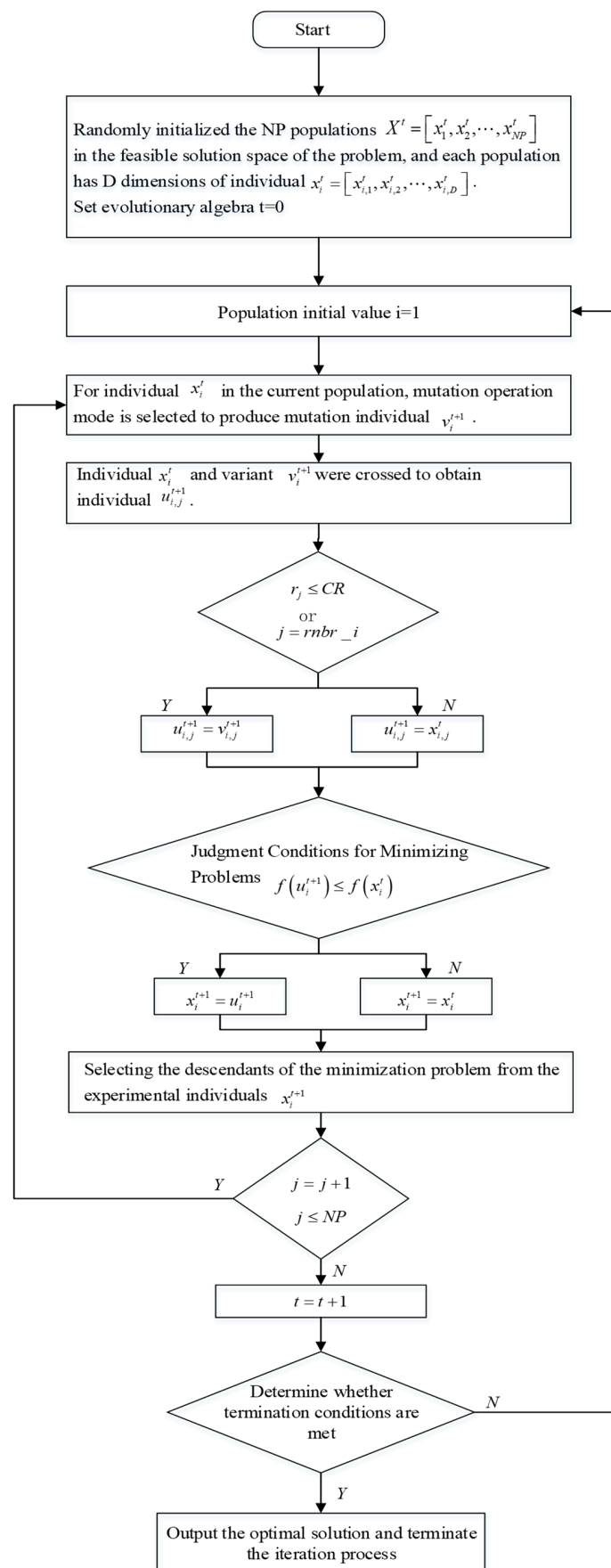


Figure 1. Flow chart of the differential evolution (DE) algorithm.

4. Proposed DEIDV-Hop Algorithm

4.1. Goal

Given that the distance between the unknown node and the anchor node is an estimated value, the error is significant in the DV-Hop localization algorithm. In the original localization algorithm, it uses the unbiased estimation criterion to calculate the average distance per hop, and the estimation error of the average distance per hop achieved through this method is zero. However, the error complies with the Gaussian distribution. According to the parameter estimation theory, as the cost function of estimating the sub-error, it is more reasonable to use the mean square error than use only the variance or deviation. Therefore, the calculation formula of the average distance per hop is improved in this investigation.

4.2. Process of Flooding

Flooding, as a classic routing algorithm, is the simplest and most reliable one. Nevertheless, it excessively occupies network resources and generates a large amount of communication overhead, making routing and link resources inefficient.

The routing algorithms involved in the first and second step of the DV-Hop algorithm in this paper are based on the Flooding, including the original DV-Hop algorithm. To reduce network communication overhead, reduce network energy consumption as well as to extend the network life cycle, the EEDCF routing algorithm [42] is considered and applied.

4.3. Process of Calculating the Average Distance per Hop

From the traditional method on the unbiased estimation criterion to calculate the average distance of per-hop $HopSize_i$, let the value of the Equation (13) be zero.

$$f_1 = \frac{1}{m-1} \sum_{i \neq j}^m (d_{i,j} - HopSize_i \cdot h_{i,j}) \quad (13)$$

The minimum mean square error criterion is used to calculate the average distance per hop, and calculated by minimizing the Equation (14).

$$f_2 = \frac{1}{m-1} \sum_{i \neq j}^m (d_{i,j} - HopSize'_i \cdot h_{i,j})^2 \quad (14)$$

Let $\frac{\partial f_2}{\partial HopSize'_i} = 0$, the estimated average distance per hop based on the minimum mean square error criterion can be obtained as:

$$HopSize'_i = \frac{\sum_{i \neq j}^m h_{i,j} d_{i,j}}{\sum_{i \neq j}^m h_{i,j}^2} \quad (15)$$

The average distance per hop is determined through Equation (16):

$$HopSize'' = \frac{\sum_{i=1}^m HopSize'_i}{m} \quad (16)$$

After calculating $HopSize''$, each anchor node broadcasts its average distance per hop $HopSize''$ in the system through controlled flooding. When the unknown node u retrieves information from the anchor node i , it uses Equation (17) to determine the distance between the unknown node u and anchor node i .

$$d'_{u,i} = HopSize'' \times h_{u,i} \quad (17)$$

4.4. Improvement of Differential Evolution Algorithm

In the basic differential evolution algorithm, three different individuals are randomly used to carry out mutation, crossover and individual update. This method is relatively simple, however, the update of individuals is unsighted without using the information of the optimal individuals obtained in the algorithm search that reduces the efficiency of the algorithm. Therefore, in the improved algorithm, the social part of PSO is introduced into the basic differential evolution algorithm. By learning from the optimal individual $X_g = (x_{g1}, x_{g2}, \dots, x_{gn})$ in the population, the updated individual can obtain the optimal individual heuristic information and promote the convergence speed of the algorithm.

With the introduction of the heuristic information of the global optimal individual, individuals tend to converge to the global optimal individual with the operation of the algorithm, leading to the convergence of individuals and the decline of population diversity. To improve the global searching ability of the algorithm, the diversity of the population and the global searching ability are increased by improving the mutation and crossover operation, and they are presented next.

4.4.1. Crossover

In the basic differential evolution algorithm, the target individual x_i^t and the mutant individual v_i^{t+1} are crossed to generate the experimental individual u_i^{t+1} . In the improved algorithm, the social learning part of the particle swarm optimization algorithm is introduced to make the current individual x_i^t learn from the optimal individual x_{best}^t of the group, and the results are crossed with the mutation vector v_i^{t+1} . By using the heuristic information carried by the group optimal value, the optimization efficiency of the algorithm is improved. The cross operation is as follows:

$$u_{i,j}^{t+1} = \begin{cases} v_{i,j}^{t+1}, & \text{if } r_j \leq CR \text{ or } j = \text{rnbr_}i \\ x_{i,j}^t + \text{rand}[0,1] \times c \times (x_{best}^t - x_{i,j}^t), & \text{otherwise} \end{cases} \quad (18)$$

where $c = 2$. $x_{i,j}^t + \text{rand}[0,1] \times c \times (x_{best}^t - x_{i,j}^t)$, and signifies that an individual x_i^t learns from the optimal individual x_{best}^t of the population.

4.4.2. Selection

In the improved differential evolution algorithm, the diversity of the population and the global search ability are increased by carrying out small probability random variation on the variation vector. The operation of mutation is as follows:

$$v_i^{t+1} = \begin{cases} x_j^{(L)} + \text{rand}[0,1] \times (x_j^{(U)} - x_j^{(L)}), & \text{if } \text{rand}[0,1] \leq P_r \\ x_{r_1}^t + F \times (x_{r_2}^t - x_{r_3}^t), & \text{otherwise} \end{cases} \quad (19)$$

P_r is the probability of random variation. When the random number $\text{rand}[0,1]$ is higher than the random mutation probability P_r , the mutation operation mode of the basic difference evolution algorithm is adopted; random mutation is carried out, if otherwise.

4.5. Differential Evolution Algorithm Implementation

In this paper, an optimized mathematical model is established by taking into consideration the distance between the unknown node and the anchor node, the location estimation process is simplified to minimize the optimization problem, and next, DE is used to provide a solution for the optimization model. First, the objective function is defined as:

$$f(x_u, y_u) = \text{Min} \sum_{i=1}^m \left| \sqrt{(x_u - x_i)^2 + (y_u - y_i)^2} - d'_{u,i} \right| \quad (20)$$

Among them, (x_u, y_u) is the coordinate of the unknown node, (x_i, y_i) is the coordinate of the anchor node, and $d'_{u,i}$ is the estimated distance between the unknown node u and the anchor node i .

According to the aim function $f(x_u, y_u)$, the fitness function is expressed as:

$$\begin{aligned} & \text{fitness}(x_u, y_u) \\ &= \sum_{u=1}^n \left(\frac{1}{h_{u,i}} \right)^2 f(x_u, y_u) \\ &= \sum_{u=1}^n \left(\frac{1}{h_{u,i}} \right)^2 \left(\sum_{i=1}^m \left| \sqrt{(x_u - x_i)^2 + (y_u - y_i)^2} - d'_{u,i} \right| \right) \end{aligned} \quad (21)$$

4.6. Complete DEIDV-Hop Algorithm

By combining Sections 3 and 4, this paper proposes an enhanced DV-Hop algorithm based on DE called DEIDV-Hop, with the corresponding pseudo-code depicted in Algorithm 1.

The flow of the algorithm follows with DEIDV-Hop initializing network parameters and generating simulated network topology (random or grid topology) yet utilizing the shortest path method to retrieve the hop-count value between nodes, corresponding to lines 1–13. After that, the information on a hop-count value between nodes is retrieved, and the average distance per hop is calculated by Equation (16). Next, the estimated distance between unknown nodes and anchor nodes is calculated according to line 15. As the third step and corresponding to lines 16–36, the DE algorithm is utilized to estimate the location of unknown nodes by initializing the parameters of DE and have the maximum number of iterations set. It mutates the target individual selection Equation (18) in line 22, and Equation (19) is used for the cross operation to obtain test individuals in lines 23–25. The greedy criterion is used to select the next generation of individuals for the target and experimental individuals in lines 27–31; finally, the optimal individual in the group is the estimated location of the unknown node at the end of the iteration.

Algorithm 1: The procedure of DEIDV-Hop

```

1: Initialization: total number of nodes  $N$ , Percentage  $p$  of anchor nodes, communication radius  $R$ ;
2: Input: Parameter settings of DEIDV-Hop and the experimental area;
   Parameter settings of DEIDV-Hop: see Table 1
   Experimental area is  $100 \times 100 \text{ m}^2$ 
3: Network deployment nodes to generate simulated network topology;
4: Calculate the hop-count value  $h_{i,j}$  according to the shortest path algorithm
5: for  $k = 1$  to  $N$ 
6:   for  $i = 1$  to  $N$ 
7:     for  $j = 1$  to  $N$ 
8:       if  $\text{short\_path}(i,k) + \text{short\_path}(k,j) < \text{short\_path}(i,j)$ 
9:          $\text{short\_path}(i,j) = \text{short\_path}(i,k) + \text{short\_path}(k,j)$ ;
10:      end
11:    end
12:  end
13: end
14: Calculate the average distance per hop,  $\text{HopSize''}$  according to Equation (16);
15: Calculate the unknown distances;
16: for  $k = 1$  to  $NP$ 
17: Initialization: Generates  $NP$  individuals that contain 2 dimensions of variables according to Equation (9);
18:   Calculate and evaluate each individual  $x_i$ ;
19:    $t = 1$ ;
20:   While  $t < t_{max}$  do
21:     for  $i = 1$  to  $NP$  //Mutation and Crossover
22:       According to Equation (18), mutation vector  $v_i^t$  is generated;
23:       for  $j = 1$  to  $D$ 
24:         Trial vector  $u_i^t$  was obtained by crossover according to Equation (19);
25:       end
26:       Trial vector  $u_i^t$  was selected;
27:       if  $f(u_i^t) < f(x_i^t)$  then //Minimize optimization problems
28:          $x_i^{t+1} = u_i^t$ ;
29:       else
30:          $x_i^{t+1} = x_i^t$ ;
31:       end
32:     end
33:      $t = t + 1$ ;
34:   end
35:   The best individual is the location of the unknown node;
36: end

```

Table 1. Parameter settings of DEIDV-Hop.

Parameter	Value
Deployment area size	100 m \times 100 m
Percentage of the anchor node	10–40%
Communication radius R	20–40 m
NP	20
F	0.9
CR	0.2
t_{max}	20
P_r	0.02

5. Experimental Results and Analysis

Experimental analyses are carried out under four different network simulation conditions to show the effectiveness of the proposed algorithm, listed as random topology, grid topology, C-Shaped random topology, and C-Shaped grid topology, as shown in Figure 2 respectively. In random topology, all nodes are randomly deployed in the corresponding experimental area, while in grid topology, each node is deployed at the intersection of gridlines and differentiated according to the proportion of the deployment deviation, in which ratio of 0 means that the nodes are precisely distributed at the gridline intersection. In comparison with random topology, the node distribution of grid topology is uniform.

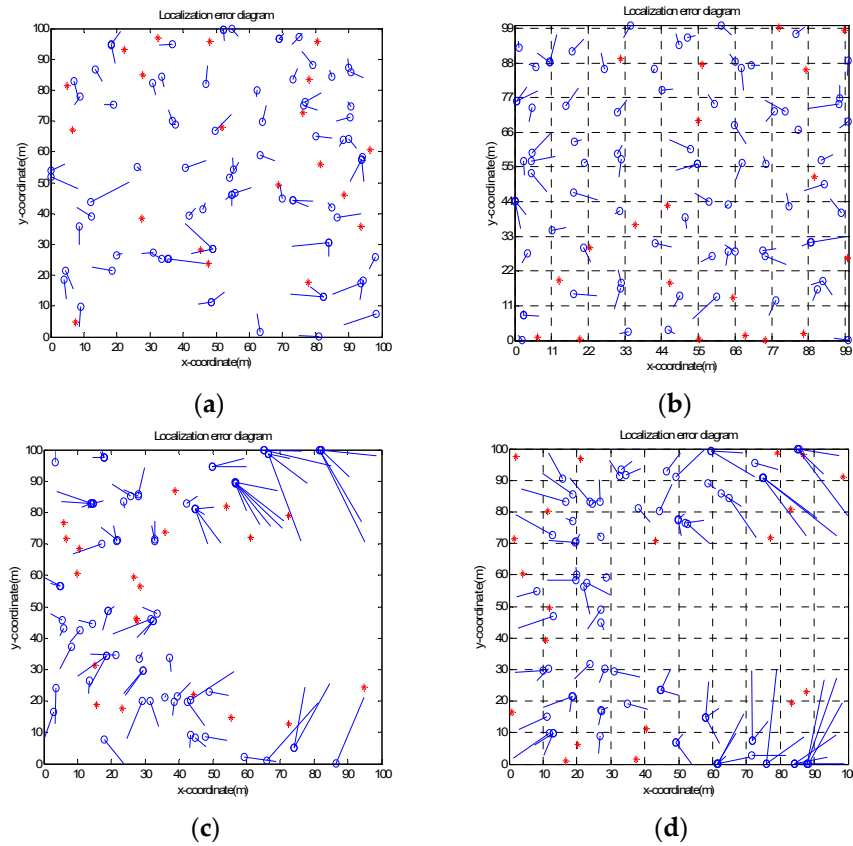


Figure 2. (a) Node error distribution diagram in square random topology; (b) Node error distribution diagram in a square grid topology; (c) Node error distribution diagram in C-Shaped random topology; (d) Node error distribution diagram in C-Shaped grid topology.

When designing the C-shaped area, we considered excavating a part of the square area during the simulation. The cut-out area is $30 \times 70 \text{ m}^2$ (the original area was $100 \times 100 \text{ m}^2$).

The DEIDV-Hop algorithm is compared with DV-Hop, PSODV-Hop [43], and GSODV-Hop [44] through simulation implemented on MATLAB 2014a [45], running on a desktop PC with one Intel(R) Core(TM) i5-6500 CPU @3.20 GHz processor, 8 GB RAM, and Windows 7 OS.

In this investigation, only the two-dimensional coordinate plane is considered. All experimental results are averaged by running 100 times independently, and the size of the experimental area is $100 \times 100 \text{ m}^2$.

The localization error of unknown nodes u is:

$$Error_u = \sqrt{(x_u^{est} - x_u^{act})^2 + (y_u^{est} - y_u^{act})^2} \quad (22)$$

The average localization error (ALE) is taken as the evaluation criterion, and calculated as:

$$\text{Average Localization Error (ALE)} = \frac{\sum_{i=1}^n \sqrt{(x_u^{\text{est}} - x_u^{\text{act}})^2 + (y_u^{\text{est}} - y_u^{\text{act}})^2}}{n \times R} \quad (23)$$

where $(x_u^{\text{est}}, y_u^{\text{est}})$ and $(x_u^{\text{act}}, y_u^{\text{act}})$ is the estimated location and actual location of the unknown node u . ALE is the average localization error, N the total number of nodes, n is the number of unknown nodes.

In Equation (23), the numerator represents the Euclidean distance between the estimated location and the actual location of the unknown node, and the estimated error distance. This article includes R in the denominator, thus, Equation (23) with R is called the normalized average localization error.

Thus, n is given as:

$$n = N - m \quad (24)$$

Experiments verify its influence on the average localization error in three aspects: the communication radius of nodes, the percentage of anchor nodes, and the total number of nodes. The parameters used in the DEIDV-Hop experiment are shown in Table 1, while the parameter settings for PSODV-Hop [43] and GSODV-Hop [44] are identical to original references and shown in Tables 2 and 3, respectively.

Table 2. Parameter settings of PSODV-Hop.

Parameter	Value
C_1	2.05
C_2	2.05
No of particles	20
V_{\max}	10
No of iterations	20

Table 3. Parameter settings of GSODV-Hop.

Parameter	Value
ρ	0.4
γ	0.6
β	0.08
I_o	5
n_t	5
N	100

The error graphs of nodes in four topologies, namely square random topology, square grid topology, C-Shaped random topology, and C-Shaped grid topology are listed in Figure 2. As settings for the investigation, the communication radius R of the node is 20 m, the total number of deployment nodes is 100, and the percentage of anchor nodes is 20%.

As shown in Figure 2, when the node deployment area changes from square topology to C-shaped topology, the average localization error of the node rises sharply. On the other hand, from the perspective of node distribution, random topology and grid topology can be considered a sparse network and intensive network. It can be observed that the nodes of the grid topology are evenly distributed if compared with the random topology; the error of nodes is smaller as well. As the node deployment changes from the random topology to the grid topology, the average localization error of the nodes decreases. The main reason is due to the influence of the deployment area, where the node distribution is not uniform and the hop-count value between nodes is no longer accurate.

5.1. Effect of Deployment Deviation Ratio λ in a Grid Topology

In this section, we study the effect of different deployment deviation factors on ALE for two grid topologies. When grid topology nodes are deployed, different values λ can be selected, $\lambda \in [0, 1]$. As settings for the investigation environment, the total number of deployment nodes is 100, the percentage of anchor nodes is 20%, and the communication radius is 20 m. As discussed above, a smaller grid deployment deviation factor will have a better effect in the case of node deployment in a grid topology.

As shown in Figure 3, with the increase of λ , the ALE of DEIDV-Hop in two grid topologies also increases, though different λ values have a tiny influence on ALE results. When $\lambda = 0$, the grid topology ALE is the smallest and the localization effect is the best. However, in practical production and living applications, it is challenging to deploy nodes with precision at fixed points, and often at random. In this investigation, $\lambda = 0.4$ is adopted as a condition in the distribution of real network simulations, although other values such as 0.3 and 0.5 are also suitable.

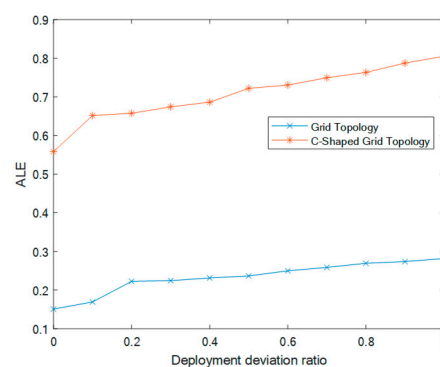


Figure 3. Effect of deployment deviation factors on two grid topology average localization error (ALE).

5.2. Effect of Communication Radius on Average Localization Error (ALE)

In this section, as settings for the investigation, the communication radius R of the node varies from 20 m to 40 m, the total number of deployment nodes is 100, and the percentage of anchor nodes is 20%. Experimental results are shown in Figures 4 and 5.

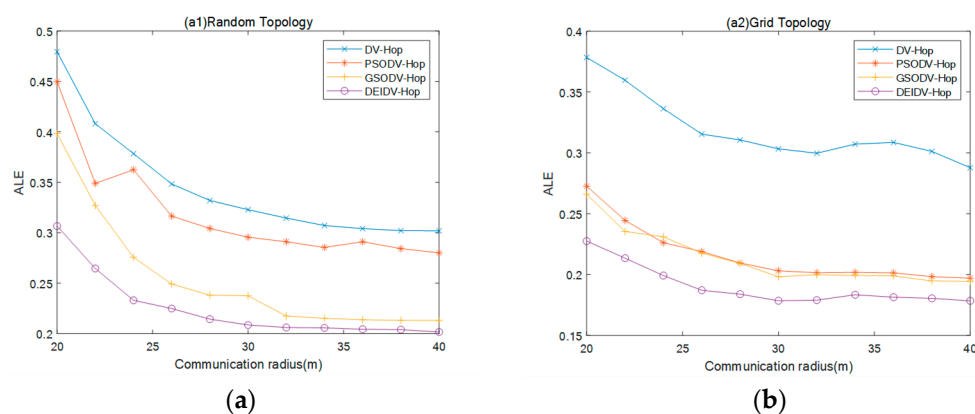


Figure 4. Cont.

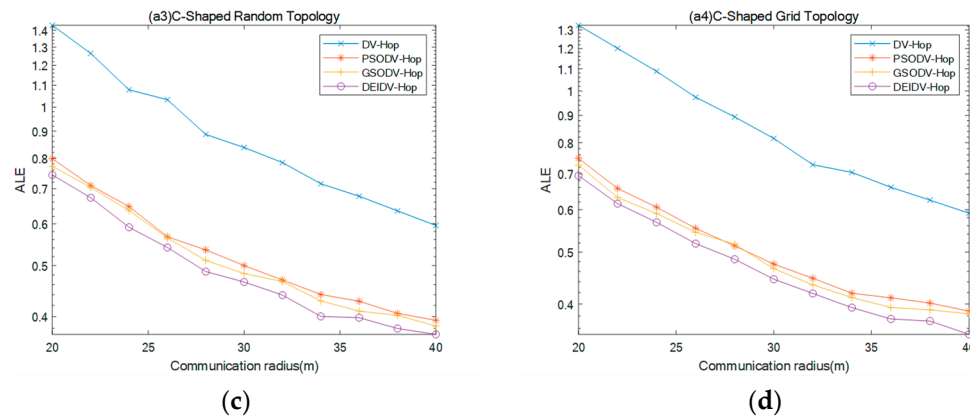


Figure 4. (a) Effect of communication radius on ALE in square random topology; (b) Effect of communication radius on ALE in a square grid topology; (c) Effect of communication radius on ALE in C-Shaped random topology; (d) Effect of communication radius on ALE in C-Shaped grid topology.

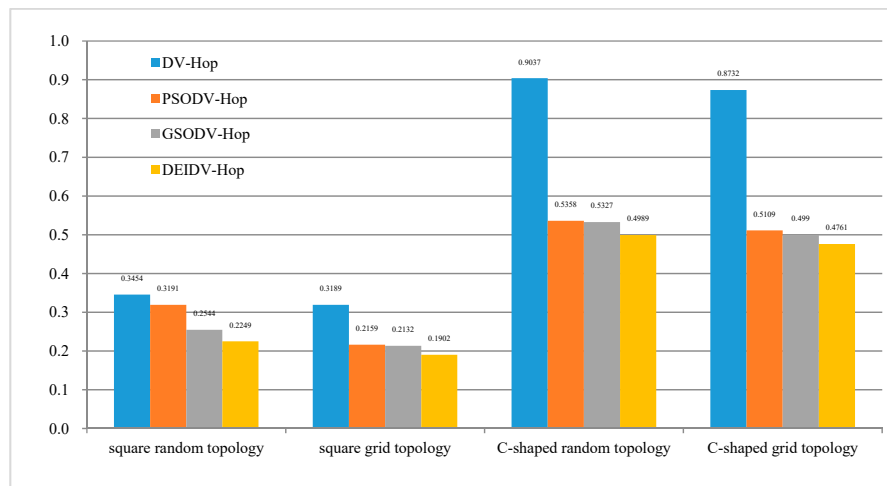


Figure 5. The average value of ALE for four topologies and algorithm.

From Figure 4, it can be seen that, under the four network topologies, the average ALE on four localization algorithms gradually decreases with the increase of the communication radius of the nodes. Among these four algorithms, DEIDV-Hop always has the smallest ALE. As shown in Figure 5:

- PSODV-Hop reduces the average ALE by 7.61%, GSODV-Hop reduces by 26.35%, and DEIDV-Hop reduces by 34.88% when compared with DV-Hop in the square random topology,
- PSODV-Hop reduces the average ALE by 32.3%, GSODV-Hop reduces by 33.15% and DEIDV-Hop reduces by 40.36% when compared with DV-Hop in the square grid topology,
- Three meta-heuristic localization algorithms PSODV-Hop, GSODV-Hop, and DEIDV-Hop improve in average 40.71%, 41.05%, and 44.79%, respectively on localization accuracy when compared with DV-Hop in the C-Shaped random topology,
- Three meta-heuristic localization algorithms PSODV-Hop, GSODV-Hop, and DEIDV-Hop improve on average 41.49%, 42.85%, and 45.48%, respectively, on localization accuracy when compared with DV-Hop in C-Shaped grid topology.

With the increase of communication radius, the communication range of unknown nodes becomes significant; there will have more single-hop nodes that can establish direct contact with more adjacent nodes. Also, it reduces the hop-count value between some of the unknown nodes and anchor nodes. The average distance per hop estimated by the algorithm and the hop-count value between nodes also tends to be accurate. Therefore, the estimated distance between the unknown node and the anchor

node is also more accurate, and a more accurate estimation of the location of the unknown nodes is obtained. The Euclidean distance between the estimated and actual location of the unknown node decreases, that is, the numerator of the ALE formula decreases, and the denominator R increases, which will inevitably bring about a decrease in the value of the entire formula.

In comparison with the random topology, the four algorithms under the grid topology have a significantly smaller average of ALE under the same conditions, which is caused by the denser distribution of network nodes. Due to the poor connectivity of nodes in sparse networks, some of the isolated nodes may be challenging to locate. The uneven distribution of nodes increases the hop-count value of unknown nodes from anchor nodes, and the estimated distance error also increases. The distribution of hop-count values between dense network nodes is relatively uniform, and the network connectivity between nodes is functional; thus, the average of ALE will be smaller.

5.3. Effect of the Percentage of Anchor Nodes on ALE

In this section, as settings for the investigation, the percentage of anchor nodes gradually changes from 10% to 40%, with the number of deployment nodes 100 and communication radius of nodes 20 m. Experimental results are shown in Figures 6 and 7.

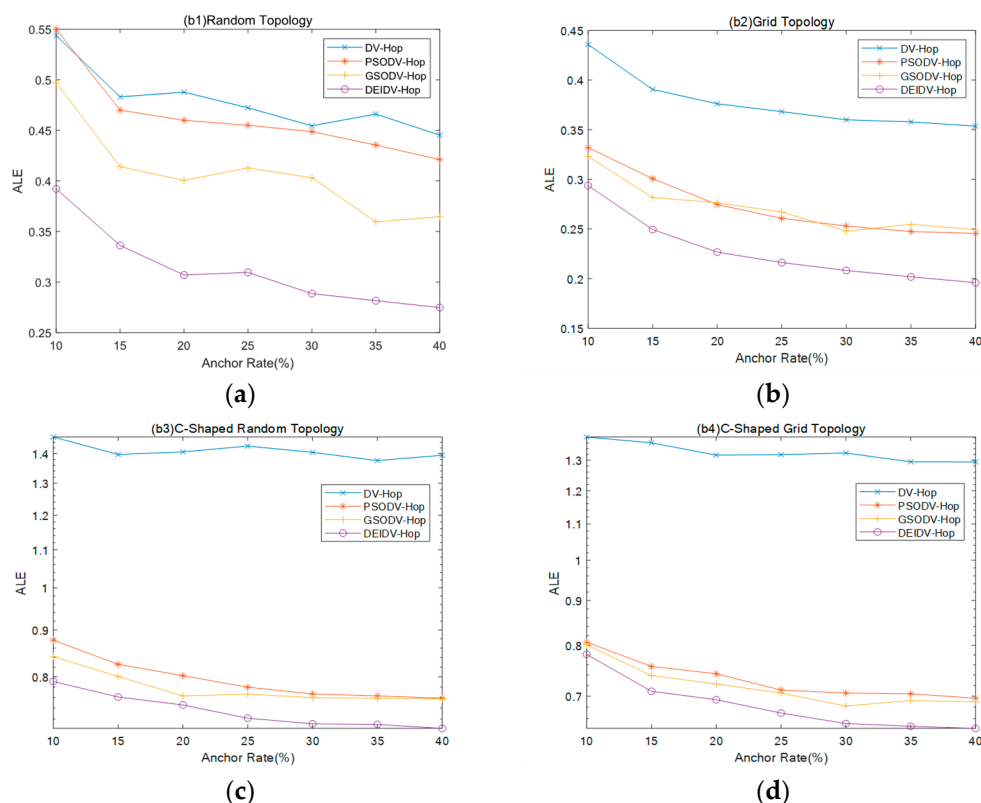


Figure 6. (a) Effect of the percentage of anchor nodes in square random topology on ALE; (b) Effect of the percentage of anchor nodes in square grid topology on ALE; (c) Effect of the percentage of anchor nodes in C-Shaped random topology on ALE; (d) Effect of the percentage of anchor nodes in C-Shaped grid topology on ALE.

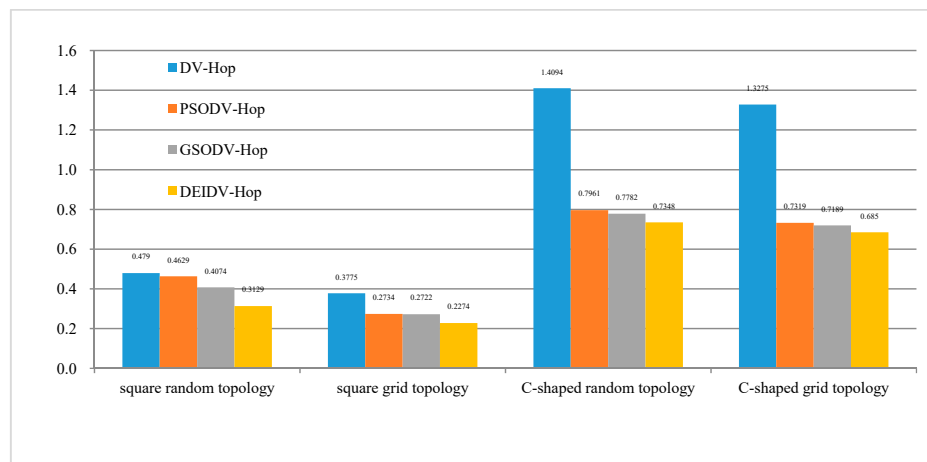


Figure 7. The average value of ALE for four topologies and algorithm.

As shown in Figure 6, the ALE of four algorithms decreases with the increasing anchor node ratio, and DEIDV-Hop always has the smallest ALE. As shown in Figure 7 and in comparison with DV-Hop:

- PSODV-Hop reduces the average of ALE by 3.36%, GSODV-Hop reduces by 14.95%, and DEIDV-Hop reduces by 34.68% in the square random topology;
- PSODV-Hop reduces the average of ALE by 27.58%, GSODV-Hop reduces by 27.89%, and DEIDV-Hop reduces by 39.76% in the square grid topology;
- Three meta-heuristic localization algorithms PSODV-Hop, GSODV-Hop, and DEIDV-Hop improve average 43.51%, 44.79%, and 47.86%, respectively, on localization accuracy in the C-shaped random topology;
- Three meta-heuristic localization algorithms PSODV-Hop, GSODV-Hop, and DEIDV-Hop improve average 44.87%, 45.85%, and 48.4%, respectively, on localization accuracy in the C-shaped grid topology.

With the increasing ratio of anchor nodes, there are more anchor nodes in the deployment area; accordingly the accuracy of the average distance per hop of anchor nodes is higher. DV-Hop is a calculation method based on the estimated distance, and the localization error decreases proportionally with the decrease of the estimated distance error. The increasing value of anchor nodes provides more favorable conditions for meta-heuristic localization algorithms and support to find the optimal solution. As a result, the average ALE is reduced because the location of nodes is accurately estimated.

5.4. Effect of Node Density on ALE

In this section, the percentage of anchor nodes is 20% and the number of square topology deployment nodes is 100 to 400. Besides, the number of deployment nodes in the C-shaped deployment area is 100–406, and the communication radius 20 m. Experimental results are shown in Figures 8 and 9.

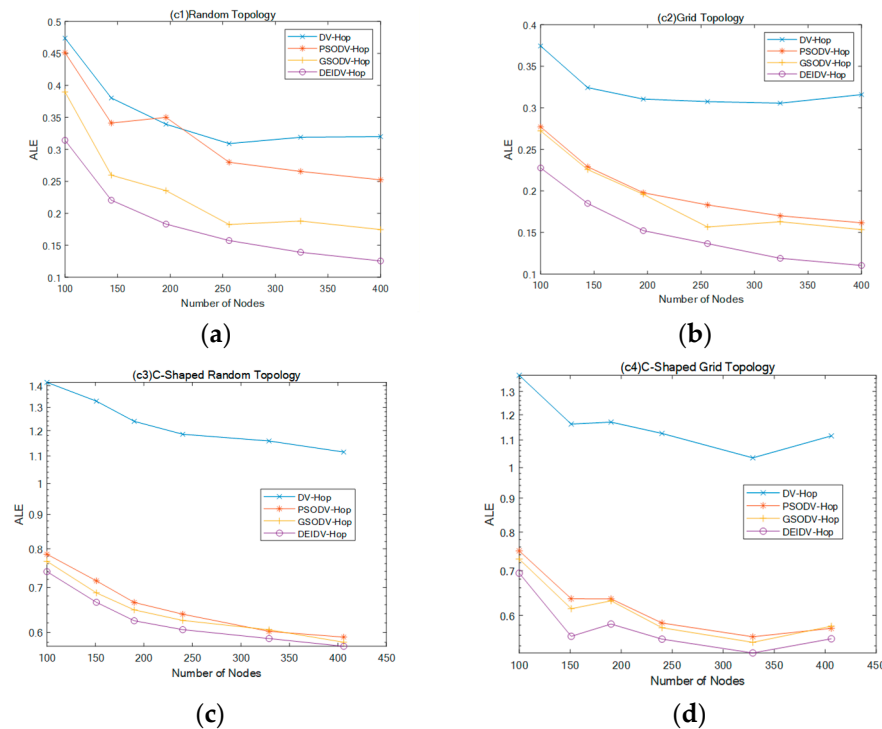


Figure 8. (a) Effect of the total number of nodes on ALE in square random topology; (b) Effect of the total number of nodes on ALE in a square grid topology; (c) Effect of the total number of nodes on ALE in C-Shaped random topology; (d) Effect of the total number of nodes on ALE in C-Shaped grid topology.

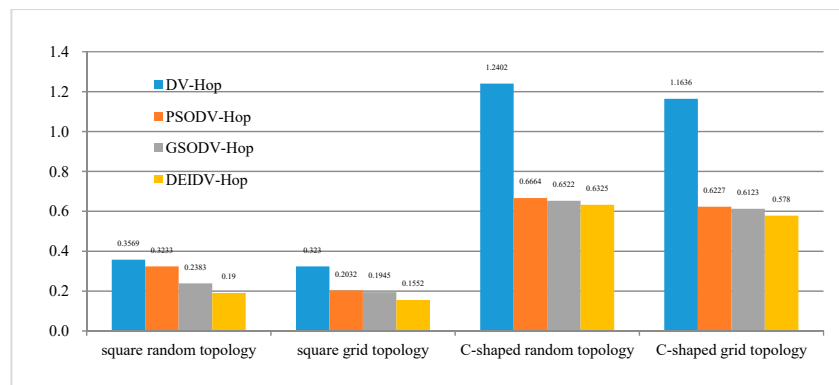


Figure 9. The average value of ALE for four topologies and algorithm.

As seen in Figure 8, as the percentage of anchor nodes increases, the ALE of the four algorithms decreases, and DEIDV-Hop always has the smallest ALE amongst the four network topologies. Using a comparison in Figure 9 with DV-Hop,

- PSODV-Hop reduces the average of ALE by 9.41%, GSODV-Hop reduces by 33.23%, and DEIDV-Hop reduces by 46.76% in the square random topology;
- PSODV-Hop reduces the average of ALE by 37.09%, GSODV-Hop reduces by 39.78%, and DEIDV-Hop reduces by 51.95% in the square grid topology;
- Three meta-heuristic localization algorithms PSODV-Hop, GSODV-Hop, and DEIDV-Hop improve average 46.25%, 47.41%, and 49%, respectively, on localization accuracy in the C-shaped random topology;

- Three meta-heuristic localization algorithms PSODV-Hop, GSODV-Hop, and DEIDV-Hop improve average 46.49%, 47.39%, and 50.33%, respectively, on localization accuracy in the C-shaped grid topology.

With the increase in the number of deployed nodes, the distribution of nodes becomes denser and tends to be in intensive networks. As each node has more single-hop connected nodes, the network connectivity is stronger. The deviation of the shortest path and straight line between nodes is relatively small, as is also the estimated distance error between the unknown nodes and anchor nodes. As an overall evaluation, with the increase in the number of deployed nodes, the average localization error of nodes also decreases, becoming more effective and productive.

6. Convergence Speed of the Algorithm and the Localization Error of Unknown Nodes

Experimental analysis was performed under four different network simulation conditions to prove the effectiveness of the convergence speed of the proposed algorithm and the localization error of each unknown node. The four network topologies are square random topology, square grid topology, C-shaped random topology, and C-shaped grid topology. Experimental results are shown in Figures 10 and 11. The size of the experimental area is $100 \times 100 \text{ m}^2$.

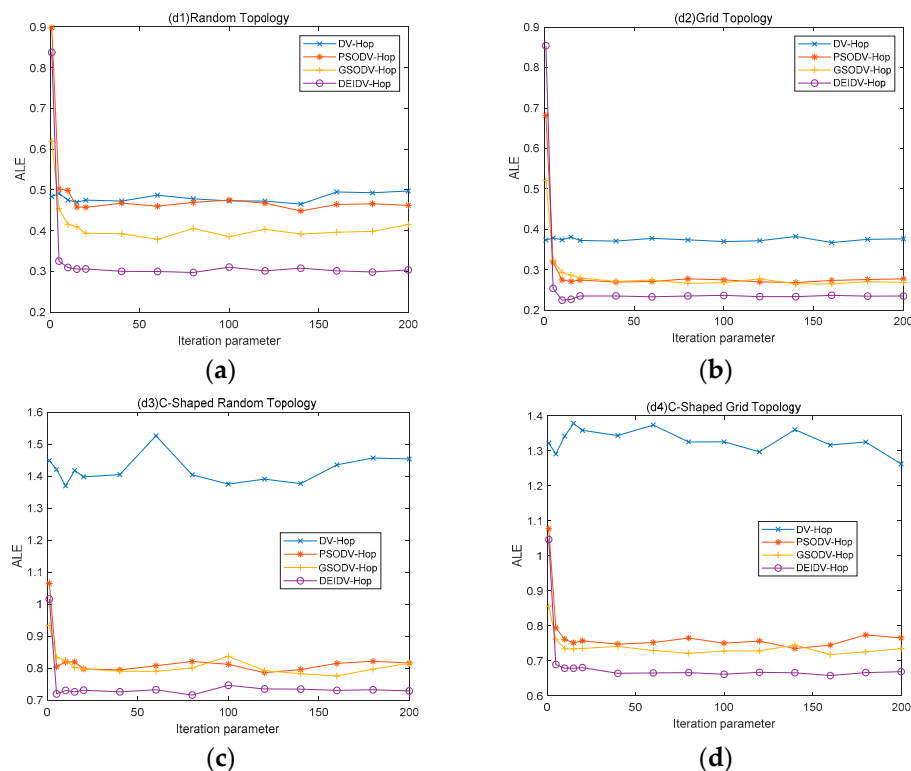


Figure 10. (a) Effect of the Iteration parameter on ALE in square random topology; (b) Effect of the Iteration parameter on ALE in a square grid topology; (c) Effect of the Iteration parameter on ALE in C-Shaped random topology; (d) Effect of the Iteration parameter on ALE in C-Shaped grid topology.

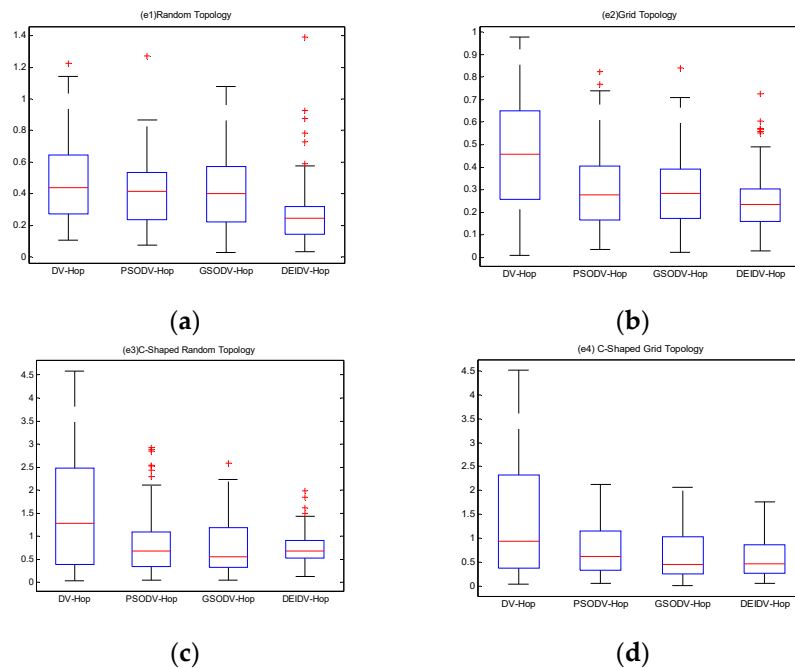


Figure 11. (a) The ALE of each unknown node in square random topology; (b) The ALE of each unknown node in a square grid topology; (c) The ALE of each unknown node in C-Shaped random topology; (d) The ALE of each unknown node in C-Shaped grid topology.

6.1. Convergence Speed of the Algorithm

In this section, the percentage of anchor nodes is 20% and the number of deployment nodes is 100. The communication radius is 20 m. The number of iterations of the algorithm is 1–200. All experimental results are averaged by executing 100 times independently. Experimental results are shown in Figure 10.

According to the results of Figure 10, three meta-heuristic algorithms have faster convergence speeds yet better effects in the square topology. It can be clearly seen that the convergence speed of the three meta-heuristic algorithms is relatively fast; notably, the algorithm DEIDV-Hop proposed in this paper has a fast convergence speed and small localization error.

In the C-shaped topology, it is easy to see that the DV-Hop algorithm is very unstable. Because of changes in the deployment environment, there will be some isolated single-hop nodes, so the localization effect of the algorithm is very unstable. Looking at the curve in the Figure 10, the localization error values of the PSODV-Hop algorithm and GSODV-Hop algorithm continue to fluctuate with the increase of the number of iterations; in contrast, the DEIDV-Hop algorithm tends to have a stable value, which demonstrates the robustness of the algorithm.

The relationship between the number of iterations of the simulation algorithm and ALE in four topological environments proves that the DEIDV-Hop algorithm has fast convergence speed, robustness, and stability.

6.2. The Localization Error of Unknown Nodes

In this section, the percentage of anchor nodes is 20% and the number of nodes is 100. The communication radius is 20 m. Experimental results are shown in Figure 11. In this section, $ALE \geq 1$ is considered as an isolated node, and it is considered as a node that cannot be located.

As in Figure 11, boxplots are applied to show the following ALE results: minimum/maximum of the localization error, the median, and the 25th and 75th percentiles of the independent simulations of each unknown node in the algorithms of four network topologies. The DEIDV-Hop algorithm is relatively stable, and the localization error is lower than the original algorithms DV-Hop algorithm, PSODV-Hop algorithm, and GSODV-Hop algorithm,

- In the square random topology, there are four nodes in the DV-Hop algorithm that cannot be located, and 33 nodes are higher than the average of the localization error. In the PSODV-Hop algorithm, one node cannot be located, and the localization error of 44 nodes is higher than the average. In the GSODV-Hop algorithm, two nodes cannot be located, and the localization error of 38 nodes is higher than the average. In the DEIDV-Hop algorithm, one node cannot be located, and 28 nodes are higher than the average.
- In the square grid topology, among the four algorithms, no unknown node cannot be located. In the DV-Hop algorithm, the localization error of 41 nodes is higher than the average of the localization errors. The localization error of 36 unknown nodes in the PSODV-Hop algorithm is higher than the average. In the GSODV-Hop algorithm, the localization error of 36 nodes is higher than the average. In the DEIDV-Hop algorithm, the localization error of 36 nodes is higher than the average.
- In the C-shaped random topology, there are 46 nodes in the DV-Hop algorithm that cannot be located, and the localization error of 37 nodes is higher than the mean of the localization error. In the PSODV-Hop algorithm, there are 25 nodes cannot be located, and the localization error of 29 nodes is higher than the average. In the GSODV-Hop algorithm, 25 nodes cannot be located, and the localization error of 26 nodes is higher than the average. In the DEIDV-Hop algorithm, 14 nodes cannot be located, and the localization error of 36 nodes is higher than the average.
- In the C-shaped grid topology, there are 38 nodes in the DV-Hop algorithm that cannot be located, and the localization error of 34 nodes is higher than the mean of the localization error, where 26 nodes cannot be located and the localization error of 31 nodes is higher than the average in the PSODV-Hop algorithm. In the GSODV-Hop algorithm, 20 nodes cannot be located, and the localization error of 31 nodes is higher than the average. In the DEIDV-Hop algorithm, 18 nodes cannot be located, and the localization error of 30 nodes is higher than the average.

When the number of deployed nodes and the communication radius are the same, the four network topologies of the nodes will affect the distribution of the nodes. The grid topology will make the distribution of the nodes more uniform, denser, and tend to be intensive networks. In this way, each node has more single-hop connection nodes, and the hop-count value between nodes will become smaller and more regular. Thus, network connectivity is stronger. In random topology, the random distribution of nodes makes the hop-count value between nodes irregular, the connectivity between nodes deteriorates, and it tends to lead to sparse networks. There will be some isolated nodes that cannot be located, which increases the average localization error of nodes.

7. Conclusions and Future Work

In this paper, an enhanced algorithm for sensor node localization based on improved DV-Hop and DE Algorithms for WSNs is proposed, namely DEIDV-Hop. From evaluations and analysis, it is observed that the average distance per hop of anchor nodes is improved, confirming its advantages and efficacy.

In the localization estimation phase, the DE algorithm is introduced to locate unknown nodes, further reducing the localization error. To solve the problems of early maturity, low convergence, and low efficiency, the individual variation and cross operation formulas are improved. In the mutation operation, an attempt to increase the diversity of the population and the global search ability, the mutation vector is randomly mutated. Moreover, the social learning part of PSO is introduced in the crossover operation, and the optimization efficiency of PSO is improved by using the heuristic information carried by the optimal group value. Simulation results of the four network topology environments show that the average localization error of nodes can be effectively reduced by the DEIDV-Hop algorithm. Without the need for additional hardware, it shows excellent advantages and is effective. The simulation results show that the DEIDV-Hop algorithm has the fastest convergence speed and is the most stable. So far, a two-dimensional node location is the constraint to discussions in

this investigation. The way of applying the algorithm to conduct node location in a more sophisticated space-three-dimensional node location is a direction to consider in terms of future research. In addition, based on the existing research, how to accurately apply the improved routing algorithm to the DV-Hop algorithm and make appropriate improvements to the routing algorithm to reduce the communication overhead of wireless threaded sensor networks and extend the life cycle of wireless sensor networks should be considered.

Author Contributions: Conceptualization, Methodology, and Investigation, D.H. and Y.Y.; Validation and Data Curation, Y.Y. and K.-C.L.; Formal Analysis, Y.Y. and D.H.; Investigation, Writing—original draft preparation, Y.Y. and D.H.; Writing—supervision, review, and editing, K.-C.L. and R.F.d.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (No. 61672338 and No. 61873160).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, Z.; Wang, X.; Liu, L.; Huang, M.; Zhang, Y. Decentralized feedback control for wireless sensor and actuator networks with multiple controllers. *Int. J. Mach. Learn. Cybern.* **2017**, *8*, 1471–1483. [\[CrossRef\]](#)
2. Bhatti, G. Machine Learning Based Localization in Large-Scale Wireless Sensor Networks. *Sensors* **2018**, *18*, 4179. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Wang, J.; Ghosh, R.K.; Das, S.K. A survey on sensor localization. *J. Control Theory Appl.* **2010**, *8*, 2–11. [\[CrossRef\]](#)
4. Han, G.; Xu, H.; Duong, T.Q.; Jiang, J.; Hara, T. Localization algorithms of wireless sensor networks: A survey. *Telecommun. Syst.* **2013**, *52*, 2419–2436. [\[CrossRef\]](#)
5. Liang, W.; Tang, M.; Long, J.; Peng, X.; Xu, J.; Li, K.-C. A Secure Fabric Blockchain-based Data Transmission Technique for Industrial Internet-of-Things. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3582–3592. [\[CrossRef\]](#)
6. Romer, K.; Mattern, F. The design space of wireless sensor networks. *IEEE Wirel. Commun.* **2004**, *11*, 54–61. [\[CrossRef\]](#)
7. Liang, W.; Li, K.-C.; Long, J.; Kui, X.; Zomaya, A.Y. An Industrial Network Intrusion Detection Algorithm based on Multi-Feature Data Clustering Optimization Model. *IEEE Trans. Ind. Inform.* **2019**. [\[CrossRef\]](#)
8. Cui, M.; Han, D.; Wan, J. An Efficient and Safe Road Condition Monitoring Authentication Scheme Based on Fog Computing. *IEEE Internet Things J.* **2019**, *6*, 9076–9084. [\[CrossRef\]](#)
9. Liang, W.; Long, J.; Weng, T.-H.; Chen, X.; Li, K.-C.; Zomaya, A.Y. TBRS: A trust based recommendation scheme for vehicular CPS network. *Future Gener. Comput. Syst.* **2019**, *92*, 383–398. [\[CrossRef\]](#)
10. Zhao, W.; Su, S.; Shao, F. Improved DV-hop algorithm using locally weighted linear regression in anisotropic wireless sensor networks. *Wirel. Pers. Commun.* **2018**, *98*, 3335–3353. [\[CrossRef\]](#)
11. Peng, B.; Li, L. An improved localization algorithm based on genetic algorithm in wireless sensor networks. *Cogn. Neurodyn.* **2015**, *9*, 249–256. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Hari Krishnan, R.; Kumar, V.J.S.; Ponmalar, P.S. Differential Evolution Approach for Localization in Wireless Sensor Networks. In Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, India, 18–20 December 2014; pp. 1–4.
13. Storn, R.; Price, K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [\[CrossRef\]](#)
14. Sarker, R.A.; Elsayed, S.M.; Ray, T. Differential evolution with dynamic parameters selection for optimization problems. *IEEE Trans. Evol. Comput.* **2013**, *18*, 689–707. [\[CrossRef\]](#)
15. Xie, P.; You, K.; Song, S.; Wu, C. Distributed Range-free Localization via Hierarchical Nonconvex Constrained Optimization. *Signal Process.* **2019**, *164*, 136–145. [\[CrossRef\]](#)
16. Bachrach, J.; Nagpal, R.; Salib, M.; Shrobe, H. Experimental results for and theoretical analysis of a self-organizing global coordinate system for ad hoc sensor networks. *Telecommun. Syst.* **2004**, *26*, 213–233. [\[CrossRef\]](#)

17. He, T.; Huang, C.; Blum, B.M.; Stankovic, J.A.; Abdelzaher, T. Range-Free Localization Schemes for Large Scale Sensor Networks. In Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, San Diego, CA, USA, 14–19 September 2003; pp. 81–95.
18. Zhang, S.; Liu, X.; Wang, J.; Cao, J.; Min, G. Accurate range-free localization for anisotropic wireless sensor networks. *ACM Trans. Sens. Netw.* **2015**, *11*, 51. [\[CrossRef\]](#)
19. Manickam, M.; Selvaraj, S. Range-based localisation of a wireless sensor network using Jaya algorithm. *IET Sci. Meas. Technol.* **2019**. [\[CrossRef\]](#)
20. Wei, H.; Wan, Q.; Chen, Z.; Ye, S. A novel weighted multidimensional scaling analysis for time-of-arrival-based mobile location. *IEEE Trans. Signal Process.* **2008**, *56*, 3018–3022. [\[CrossRef\]](#)
21. Xiao, J.; Ren, L.; Tan, J. Research of Tdoa Based Self-Localization Approach in Wireless Sensor Network. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 2035–2040.
22. Niculescu, D.; Nath, B. Ad Hoc Positioning System (APS) Using AOA. In Proceedings of the IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428), San Francisco, CA, USA, 30 March–3 April 2003; pp. 1734–1743.
23. Xie, H.; Li, W.; Li, S.; Xu, B. An Improved DV-Hop Localization Algorithm Based on RSSI Auxiliary Ranging. In Proceedings of the 35th Chinese Control Conference (CCC), Chengdu, China, 27–29 July 2016; pp. 8319–8324.
24. Kulkarni, V.R.; Desai, V.; Kulkarni, R.V. A comparative investigation of deterministic and metaheuristic algorithms for node localization in wireless sensor networks. *Wirel. Netw.* **2019**, *25*, 2789–2803. [\[CrossRef\]](#)
25. Čapkun, S.; Hamdi, M.; Hubaux, J. GPS-free positioning in mobile ad hoc networks. *Clust. Comput.* **2002**, *5*, 157–167. [\[CrossRef\]](#)
26. Niculescu, D.; Nath, B. DV based positioning in ad hoc networks. *Telecommun. Syst.* **2003**, *22*, 267–280. [\[CrossRef\]](#)
27. Zhang, Y.; Xiong, W.; Han, D.; Chen, W.; Wang, J. Routing algorithm with uneven clustering for energy heterogeneous wireless sensor networks. *J. Sens.* **2016**, *2016*, 9. [\[CrossRef\]](#)
28. Garimella, R.M.; Edla, D.R.; Kuppili, V. *Energy Efficient Design of Wireless Sensor Network: Clustering*; Centre for Communications International Institute of Information Technology Hyderabad-500 032: Hyderabad, India, 2018.
29. Thiagarajan, R. Energy consumption and network connectivity based on Novel-LEACH-POS protocol networks. *Comput. Commun.* **2020**, *149*, 90–98.
30. Ji, W.; Liu, Z. Study on the application of DV-Hop localization algorithms to random sensor networks. *J. Electron. Inf. Technol.* **2008**, *30*, 970–974. [\[CrossRef\]](#)
31. Niculescu, D.; Nath, B. Ad Hoc Positioning System (APS). In Proceedings of the GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No.01CH37270), San Antonio, TX, USA, 25–29 November 2001; pp. 2926–2931.
32. Shang, Y.; Ruml, W. Improved MDS-Based Localization. In Proceedings of the IEEE INFOCOM, Hong Kong, China, 7–11 March 2004; pp. 2640–2651.
33. Nian-qiang, L.; Ping, L. A Range-Free Localization Scheme in Wireless Sensor Networks. In Proceedings of the IEEE International Symposium on Knowledge Acquisition and Modeling Workshop, Wuhan, China, 21–22 December 2008; pp. 525–528.
34. Gumaida, B.F.; Luo, J. A hybrid particle swarm optimization with a variable neighborhood search for the localization enhancement in wireless sensor networks. *Appl. Intell.* **2019**, *49*, 3539–3557. [\[CrossRef\]](#)
35. Girod, L.; Bychkovskiy, V.; Elson, J.; Estrin, D. Locating Tiny Sensors in Time and Space: A Case Study. In Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors, Freiburg, Germany, 18 September 2002; pp. 214–219.
36. Singh, S.P.; Sharma, S.C. Critical analysis of distributed localization algorithms in wireless sensor networks. *Int. J. Wirel. Microw. Technol.* **2016**, *4*, 72–83. [\[CrossRef\]](#)
37. Cui, L.; Xu, C.; Li, G.; Ming, Z.; Feng, Y.; Lu, N. A high accurate localization algorithm with DV-Hop and differential evolution for wireless sensor network. *Appl. Soft Comput.* **2018**, *68*, 39–52. [\[CrossRef\]](#)
38. Wu, L.; Meng, M.Q.; Huang, J.; Liang, H.; Lin, Z. An Improvement of DV-Hop Algorithm Based on Collinearity. In Proceedings of the International Conference on Information and Automation, Zhuhai, Macau, China, 22–24 June 2009; pp. 90–95.

39. Xu, S.; Wang, X.; Wang, Y.; Wang, J. Iterative Cooperation DV-Hop Localization Algorithm in Wireless Sensor Networks. In Proceedings of the IEEE 71st Vehicular Technology Conference, Taipei, Taiwan, 16–19 May 2010; pp. 1–5.
40. Chen, K.; Wang, Z.; Lin, M.; Yu, M. An Improved DV-Hop Localization Algorithm for Wireless Sensor Networks. In Proceedings of the IET International Conference on Wireless Sensor Network 2010 (IET-WSN 2010), Beijing, China, 15–17 November 2010.
41. Cheikhrouhou, O.; Bhatti, G.M.; Alroobaea, R. A hybrid DV-hop algorithm using RSSI for localization in large-scale wireless sensor networks. *Sensors* **2018**, *18*, 1469. [[CrossRef](#)]
42. Zhang, Y.; Wang, J.; Han, D.; Wu, H.; Zhou, R. Fuzzy-logic based distributed energy-efficient clustering algorithm for wireless sensor networks. *Sensors* **2017**, *17*, 1554. [[CrossRef](#)]
43. Singh, S.P.; Sharma, S.C. A PSO based improved localization algorithm for wireless sensor network. *Wirel. Pers. Commun.* **2018**, *98*, 487–503. [[CrossRef](#)]
44. Song, L.; Zhao, L.; Ye, J. DV-Hop Node Location Algorithm Based on GSO in Wireless Sensor Networks. *J. Sens.* **2019**, *2019*, 9. [[CrossRef](#)]
45. Li, H.; Han, D. EduRSS: Blockchain-Based Educational Records Secure Storage and Sharing Scheme. *IEEE Access.* **2019**, *7*, 179273–179289. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).