

## Research Article

# A Two-Stage Wireless Sensor Grey Wolf Optimization Node Location Algorithm Based on K-Value Collinearity

Yinghui Meng , Qianying Zhi, Qiuwen Zhang , and Erlin Lin

*School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China*

Correspondence should be addressed to Yinghui Meng; [yinghuimeng@126.com](mailto:yinghuimeng@126.com)

Received 19 April 2020; Revised 19 June 2020; Accepted 6 July 2020; Published 25 July 2020

Academic Editor: Georgios Dounias

Copyright © 2020 Yinghui Meng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the practical application of WSN (wireless sensor network), location information of the sensor nodes has become one of the essential information pieces in the whole network. At present, some localization algorithms use intelligent optimization algorithm to optimize the node group directly. Although the overall localization error is reduced, the location deviation of individual unknown nodes will be larger, and the large number of iterations will cause a large energy consumption of nodes. Aiming at the above problems, this paper comes up with a two-stage WSN localization algorithm based on the degree of K-value collinearity (DC-K) and improved grey wolf optimization. The first stage is aiming at the defects of the existing collinearity algorithm, putting forward the concept of DC-K, according to the K-value to carry out the initial location in the first stage. The second stage is using the improved grey wolf optimization algorithm to optimize the location results which were obtained in the first stage, so as to get more accurate location results. The experimental results display that this localization algorithm with a better localization accuracy has high robustness and has fewer iterations in the optimization process, which greatly reduces the energy consumption of nodes.

## 1. Introduction

Localization is to identify the coordinates of something in a geographical environment. The significance of node localization technology in WSN is to deduce the coordinates about the unknown nodes which are located in the network. Due to the continuous development and progress of sensing technology, the use of the WSN (wireless sensor network) is more and more powerful. At present, the application of WSN in transportation, medical and electrical appliance, and other areas of life is wide. WSN deploys a large number of stationary or mobile sensor nodes in the area to be monitored, making these sensor nodes form a network system through radio communication. The purpose is to collaboratively perceive multidimensional information within the monitoring area (such as humidity, temperature, noise, geological characteristics, and life characteristics) and then process and transmit the collected data. When the sensor node is working, firstly, the node sends it, collecting the physical, chemical, biological, and other information to the terminal computer, and then feeds back the information

to the user. Finally, the user makes corresponding operations according to the information. In most applications, the information sent back by the node must be combined with the coordinate of the node to make sense. Sometimes, it is even necessary for the node to simply send the location information. In conclusion, sensor node location is the critical technology in sensor networks.

At present, according to whether it is necessary to acquire the angle information or distance information between nodes through physical measurement, sensor node location technology could be divided into two parts that are range-free algorithm and range-based algorithm [1]. Due to the use of measuring equipment, the location accuracy of range-based is higher, such as RSSI (received signal strength indication). However, the cost of adding additional distance measurement equipment to the node is very high; the lifetime of the sensor node is also greatly shortened by frequent receiving of ranging information. In addition, this method fails in extremely complex geographical environments. The range-free technique is using the localization algorithm to estimate the Euclidean distance from the

unknown sensor nodes to the beacon sensor nodes or estimating the possible area containing the unknown node according to some nodes with known coordinates (beacon nodes) and then inferring the coordinates of the unknown nodes [1]. This type of technology does not require hardware ranging modules. The classical range-free node localization methods are such as DV-hop (distance vector-hop algorithm), centroid algorithm, and APIT algorithm (approximate point-in-triangulation test). Range-free technology does not require additional hardware support; its advantages are easy implementation, having low cost, low energy consumption, and high computational efficiency, compared to the range-based algorithm which can be more suitable for the actual application of the sensor network.

For increasing the location accuracy of range-free location technology, many experts and scholars have studied and improved the DV-hop, which can be divided into two steps. So, the first stage is to estimate the Euclidean distance of beacon nodes to unknown nodes. And the second stage is to infer the coordinates of the unknown nodes from the distance. For raising the accuracy of distance estimation from unknown nodes to beacon nodes, in [2], a search algorithm for similar paths is proposed to find a path between beacon node pairs with the highest similarity according to the Ochiai coefficient and calculate the single hop correction value according to the hop's number and the Euclidean distance of the pair of beacon sensor nodes, thus raising the accuracy of Euclidean distance estimation; in [3], it is assumed that the communication radius of the sensor node is  $R$ , and the one-hop's distance within the whole wireless sensor network is modified to  $R/2$ , which reduces the error within the one-hop range; in [1], the authors put forward the concept of intersection ratio; first, they calculated the area of intersection between the unknown node and the neighboring beacon node communication circle, then they calculated the ratio of this area to the entire communication range, and finally they estimated the distance between the unknown node and the beacon node according to the ratio. In the second stage, more and more scholars apply the intelligent optimization method to location estimation of wireless sensor node; in [4], PSO (particle swarm optimization algorithm) is introduced, through hundreds of iterations of the particle swarm optimization model; the coordinates of the unknown nodes that are consistent with the fitness function are finally obtained; implementing localization algorithm [5] improved the particle swarm optimization localization method and came up with the concept of quantum particle swarm, choosing the wave function to describe the state of the particle and increasing the location accuracy, but the computational cost is higher. At present, there are few improved two-stage localization algorithms for both distance estimation and coordinate calculation. In summary, the existing researches still have large errors in the distance estimation stage; in the second stage, particle swarm optimization, simulated annealing optimization, and other intelligent optimization algorithms are used directly, which can easily fall into local optimization and cause huge energy consumption due to too many iterations. However, the existing two-stage positioning

algorithm directly refers to DV-hop and intelligent optimization algorithm which cannot solve the defects of intelligent optimization algorithm that consumes too much energy and easily falls into local optimization. In this paper, aiming at the shortcomings of the above studies, we came up with a two-stage wireless sensor grey wolf optimization node localization algorithm based on K-value collinearity. In the first stage, according to the defects of collinearity [6–8], in this paper, we propose the new collinearity based on K-value and then infer to the initial location of the node from the collinearity of K-value; in the second stage, the improved localization algorithm of GWO is used to optimize the node location. Experiments show that two-stage DC-K localization method could raise the location accuracy effectively. In the optimization stage, only 30 iterations are needed to obtain a better location result, which greatly reduces the energy consumption of nodes.

## 2. Selection of Location Unit Based on Collinearity

*2.1. Error Analysis.* After obtaining the Euclidean distance which is from beacon nodes to unknown node, localization algorithms perform the trilateral measurement or maximum likelihood method to infer the coordinate of unknown sensor nodes. In the process of location estimation, the location topology between beacon sensor nodes and topology shape from beacon nodes to unknown nodes will greatly affect the localization accuracy [9]. Generally, the more uniform the distribution of beacon nodes is, the more accurate the localization will be. In the following, we take three beacon nodes as an example to discuss the influence of location topology between nodes on localization accuracy.

*Case 1.* When any two of the three beacon nodes of the localization unit are too close to each other, meanwhile, the unknown sensor node is very far from the two beacon sensor nodes and there will be a large localization error. As is displayed in Figure 1,  $A$ ,  $B$ , and  $C$  are beacon sensor nodes and  $D$  is unknown sensor node in the localization unit, compared with the distance from node  $D$  to the localization unit; the distance between  $B$  and  $C$  is easy to be ignored; therefore, the localization result of the unknown node  $D$  may appear at the location of  $D'$ , which causes a large localization error.

*Case 2.* When the beacon nodes in the localization unit are collinear or close to collinear, the localization matrix is irreversible, resulting in a large localization error and even leading to the localization failure. From Figure 2 we could get that when the beacon nodes  $A$ ,  $B$ , and  $C$  are close to collinear, then the unknown node  $D$  is located that will lead to the localization result which may be at the location of  $D'$ .

*Case 3.* When the three beacon nodes in the localization unit are very close to each other and the unknown node is very far away from the localization unit, it will cause a large localization error. As shown in Figure 3,  $A$ ,  $B$ , and  $C$  are beacon nodes,  $D$  is unknown sensor node, and  $D'$  is localization result that may occur.

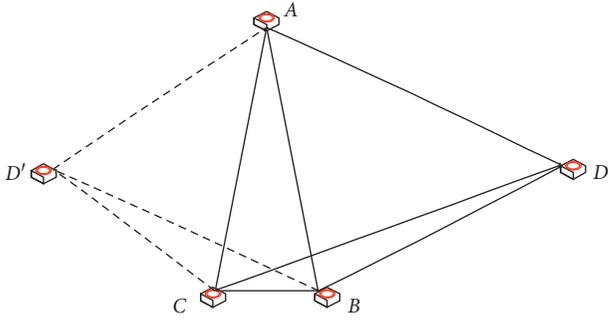


FIGURE 1: Two beacon nodes in the localization unit are close to each other.

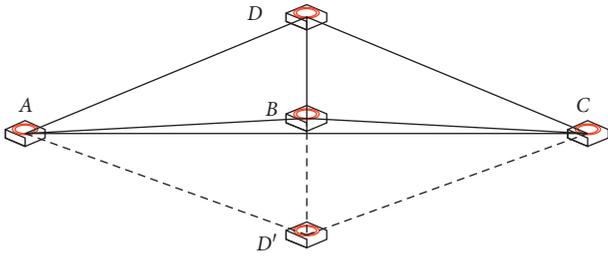


FIGURE 2: Beacon nodes in localization units are collinear or approximately collinear.

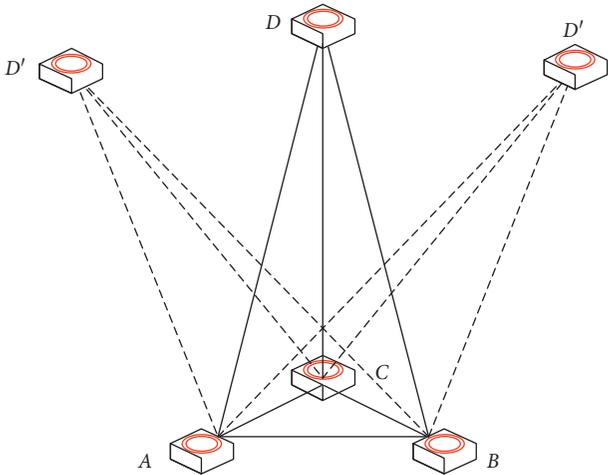


FIGURE 3: Three beacon nodes in the localization unit are close to each other.

2.2. Degree of Collinearity Based on Minimum Height (DC-H) and Minimum Cosine of Interior Angle (DC-A). In [8], the authors came up with the concept of collinearity. According to the collinearity estimate for whether the topological shape between beacon nodes is suitable for localization, this came up with the concept of DC-H. As shown in Figure 4,  $\triangle ABC$  is a triangle formed by beacon nodes A, B, and C. The minimum height is the minimum value of three heights in  $\triangle ABC$  (the height on the longest side). Therefore, the value of minimum height is the collinearity of the three beacon nodes:

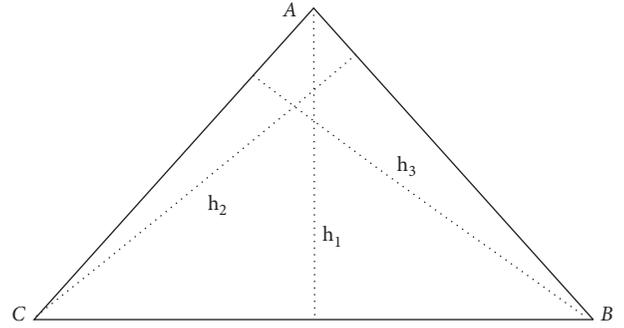


FIGURE 4: Diagram of DC-H.

$$DC - H = \min\{h_i\}, \quad i = 1, 2, 3. \quad (1)$$

Under this definition, the value range of collinearity is  $[0, (\sqrt{3}/2)R]$ ; the communication radius of the nodes is  $R$ , which is in the network. When the degree of collinearity is equal to 0, it indicates that the nodes A, B, and C are collinear and the localization accuracy is the worst; when the degree of collinearity is equal to  $(\sqrt{3}/2)R$ , beacon sensor nodes A, B, and C constitute an equilateral triangle, which is most suitable for locating unknown node. In summary, when the minimum height is closer to  $(\sqrt{3}/2)R$ , the localization accuracy is higher.

According to [7], the interior angle size of a triangle can better reflect the shape of a triangle; therefore, the degree of collinearity is defined by the cosine of the minimum interior angle, as shown in Figure 5.  $\triangle ABC$  is a triangle formed by beacon nodes A, B, and C:

$$DC - A = \max(\cos A, \cos B, \cos C). \quad (2)$$

In  $\triangle ABC$ , the cosine of the three interior angles is

$$\cos A = \frac{b^2 + c^2 - a^2}{2 \times b \times c}, \quad (3)$$

$$\cos B = \frac{a^2 + c^2 - b^2}{2 \times a \times c}, \quad (4)$$

$$\cos C = \frac{a^2 + b^2 - c^2}{2 \times a \times b}. \quad (5)$$

In a triangle, the value range of the minimum interior angle is  $[0, \pi/3]$ ; therefore, the value range of the cosine of the minimum internal angle is  $[0.5, 1]$ , which is inversely proportional to the angle. When the degree of collinearity is equal to 0.5, it means that the topology formed by the localization unit is an equilateral triangle; when the degree of collinearity is equal to 1, it means that the beacon nodes in the localization unit are collinear. In summary, the closer the DC-A is to 0.5, the higher the localization accuracy is and vice versa.

In [6], the authors considered the minimum height and the minimum angle at the same time and came up with the concept of Degree of Aggregation-Collinearity (DAC). However, in the first stage, it still uses the RSSI algorithm and needs to add peripheral devices to the nodes; it belongs

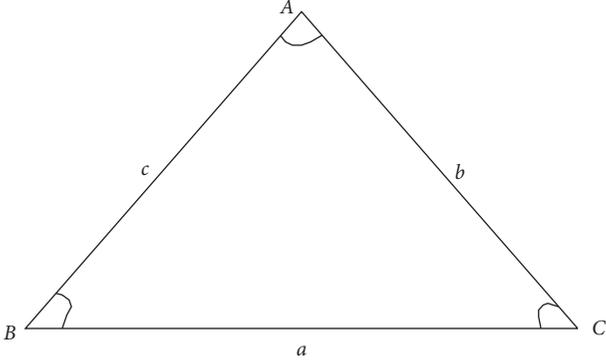


FIGURE 5: Diagram of DC-A.

to the category of the range-based algorithm, so it will not be described in this paper.

### 2.3. Defects of DC-H and DC-A

**2.3.1. Defects of DC-H.** Through the above research, it is not difficult to find that the collinearity localization algorithm based on the minimum height cannot accurately reflect the topological shape of the triangle. When the area of a triangle is also larger, the minimum height of the triangle is larger, but the topological shape may not be closer to an equilateral triangle. As shown in Figure 6, the minimum height of  $\triangle ABC$  is obviously larger than the minimum height of  $\triangle A_1B_1C_1$ ; however,  $\triangle A_1B_1C_1$  is an equilateral triangle with better localization effect, which is contrary to the statement in [8].

**2.3.2. Defects of DC-A.** If the topological shape of the three beacon nodes of a localization unit is an equilateral triangle, so, the DC-A is equal to 0.5; according to [7], the localization effect is the best. However, if the three beacon sensor nodes are close to each other, the unknown sensor nodes are far away from the beacon sensor nodes; as described in Case 3 of the above error analysis, the localization accuracy is very low.

At the same time, whether it is DC-H or DC-A, the DAC localization algorithm will have the following disadvantages: once a group of optimal localization units is found, the unknown nodes of the whole network will use it for location estimated. It can not only effectively make full use of other beacon sensor nodes' coordinate conditions which are in the WSN, but also cause large localization error.

**2.4. K-Value Collinearity (DC-K) Based on APIT Algorithm.** The theory of APIT (approximate point-in-triangulation test) is triangular coverage approximation [10], as shown in Figure 7. The unknown node randomly selects three beacon nodes from the whole sensor network and then determines whether it is located in the triangular topology formed by the three beacon nodes, repeating the process until enumerating all the beacon nodes units; finally, the center of mass of all the overlapped triangles that can make the unknown node inside is taken as its estimated location [11]. How to control the unknown sensor node is located inside the triangle which

consists of the beacon sensor nodes, being key of APIT algorithm. APIT algorithm adopts PIT test method; the principle of PIT is as follows: Figure 8 shows that  $A, B,$  and  $C$  are beacon nodes and  $D$  is unknown node; if there is a direction that enables the unknown node  $D$  to move in this direction, approaching or being away from the beacon sensor nodes  $A, B,$  and  $C$  at the same time, it indicates that  $D$  is located outside of the triangle; if it does not exist, this indicates that the unknown sensor node  $D$  is located inside of the triangle. In practice, it is impossible to test all directions [12]; so, T. HE came up with the APIT algorithm based on PIT; the principle of APIT is as follows: if there is a neighbor node away from or close to the beacon nodes at the same time relatively to  $D$ , it indicates that the unknown node is outside of the triangle; or, else,  $D$  is in the triangle [13].

In the above error analysis, it is pointed out that although the topological shape of beacon nodes in the localization unit is equilateral triangle or close to equilateral triangle, the Euclidean distance of the three beacon nodes is very close to the unknown sensor node which is very far away from the beacon nodes; it will cause a large localization error. This is a problem that has not been solved in previous research of DC algorithm. In this paper, aiming at the shortcomings and limitations of the existing DC algorithm, we came up with the K-value collinearity based on APIT algorithm; its basic principle is as follows: by executing APIT algorithm, we found the localization unit which could locate the unknown node inside the triangle and meet the K-value condition; then, these beacon nodes are used to locate the unknown node's coordinate in the localization unit. In this way, the unknown node could be controlled inside the triangle and consists of the three beacon nodes in the localization unit, which not only make full use of the location information of beacon nodes in the WSN, but also avoid the situation described in the error analysis above and make up for the common defects of other DC algorithms.

The calculation method of K-value is as follows:

$$K = \left| \frac{2h}{\sqrt{3}l} - 1 \right|, \quad (6)$$

$h$  is the minimum height (the height corresponding to the longest side or the largest interior Angle) and  $l$  is the value of the longest side in the triangle. When  $K$  is equal to the minimum value of 0,

$$\left| \frac{2h}{\sqrt{3}l} - 1 \right| = 0, \quad (7)$$

$$\frac{2h}{\sqrt{3}l} = 1, \quad (8)$$

$$\frac{h}{l} = \frac{\sqrt{3}}{2}. \quad (9)$$

So, when  $K = 0$ , the topological shape of beacon nodes is an equilateral triangle. In summary, when K-value is smaller, the distribution of beacon nodes  $A, B,$  and  $C$  is closer to equilateral triangle and the localization accuracy is higher. Using  $2h/\sqrt{3}l$ , the influence of the maximum angle and the

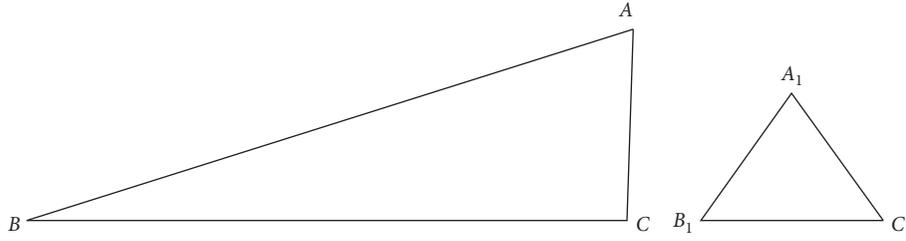


FIGURE 6: Schematic diagram of the minimum height contrast of a triangle.

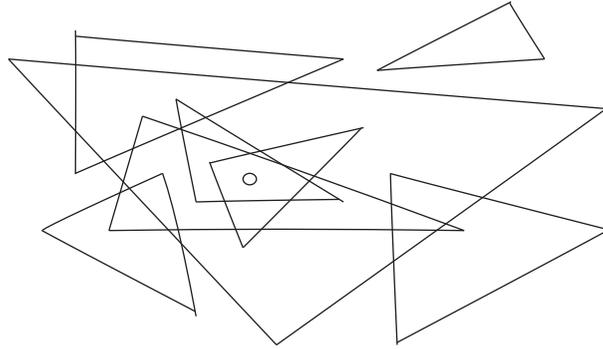


FIGURE 7: Schematic diagram of APIT algorithm.

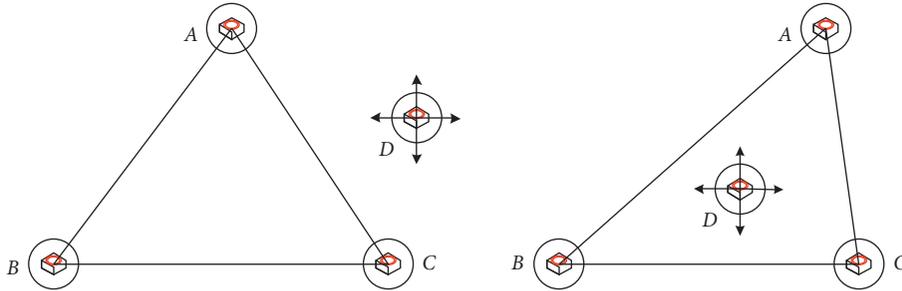


FIGURE 8: Example of APIT algorithm.

minimum height on the shape of the triangle is considered as shown in Figure 9.

Each time the algorithm is executed, an unknown node is located, so the maximum number of the iteration is set as

$$T_{\max} = \text{UNAmount (the count of unknown nodes)}. \quad (10)$$

The steps of wireless sensor node location algorithm based on K-value collinearity are as follows:

- (1) Execute the APIT algorithm, enumerate all locating units, and record the localization units in  $M$  set which can make the unknown sensor node  $u$  located inside topological triangle. If the set is empty, execute step (6) directly; otherwise, execute the next step:

$$M = \{\dots \Delta A_i B_i C_i \dots\}. \quad (11)$$

- (2) For the unknown node  $u$ , enumerate all the localization unit combinations which are in  $M$  set:  $\Delta A_1 B_1 C_1 \dots \Delta A_i B_i C_i \dots \Delta A_n B_n C_n$ . According to (6), respectively, calculate the K-value of this localization

unit; if the K-value meets (12), then record the localization units in  $N$  set. If  $N$  is empty, then execute step (6) directly. Otherwise, execute the next step:

$$0 \leq K \leq 0.3, \quad (12)$$

$$N = \{\dots \Delta A_i B_i C_i \dots\}. \quad (13)$$

- (3) At this moment, the unknown node  $u$  is inside the  $\Delta A_i B_i C_i$  formed by the localization unit in set  $N$ . If the actual distance from  $u$  to beacon nodes  $A_i$ ,  $B_i$ , and  $C_i$  is smaller, then, iteration error will be smaller when using the single hop size and hops to estimate the distance. Otherwise, the localization error will be larger [14]. Therefore, calculate the area of topological triangles in set  $N$  and take the combination of beacon nodes with the smallest area as the localization unit of unknown node  $u$ .

In  $\Delta A_i B_i C_i$ , the coordinates of beacon nodes  $A_i$ ,  $B_i$ , and  $C_i$  are  $A_i(x_{A_i}, y_{A_i})$ ,  $B_i(x_{B_i}, y_{B_i})$ ,  $C_i(x_{C_i}, y_{C_i})$ ;

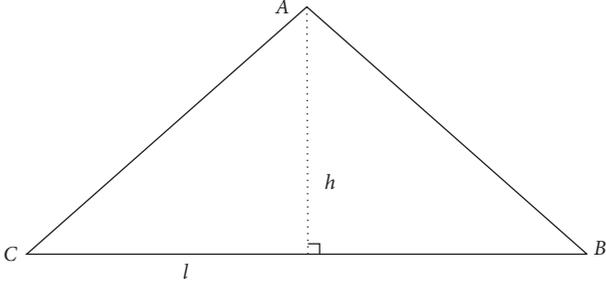


FIGURE 9: Diagram of DC-A.

the sides of the triangle are  $a$ ,  $b$ , and  $c$ , respectively;  $S$  is the area of the triangle:

$$a = \sqrt{(x_{Bi} - x_{Ci})^2 + (y_{Bi} - y_{Ci})^2}, \quad (14)$$

$$b = \sqrt{(x_{Ai} - x_{Ci})^2 + (y_{Ai} - y_{Ci})^2}, \quad (15)$$

$$c = \sqrt{(x_{Ai} - x_{Bi})^2 + (y_{Ai} - y_{Bi})^2}, \quad (16)$$

$$P = \frac{1}{2} \times (a + b + c), \quad (17)$$

$$S_i = \sqrt{p(p-a)(p-b)(p-c)}. \quad (18)$$

$$\text{LocalHopsize} = \frac{\sum_{i \neq j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\text{hop}_{ij}}, \quad x_i, x_j \in (x_{Ak}, x_{Bk}, x_{Ck}), \quad (19)$$

$$\text{LocalHopsize} * \text{hop}_{iu} = d_{iu}, \quad i \in (A, B, C). \quad (20)$$

If the estimated distance from  $u$  to  $A_k$ ,  $B_k$ , or  $C_k$  is larger than the longest side of the triangle, then the value of this distance is modified to the value of the longest side of the triangle. This is because when  $u$  is inside the triangle, the distance from  $u$  to any vertex cannot be larger than the longest side of the triangle. If

$$d_{iu} > \max\{a b c\}, \text{ then } d_{iu} = \max\{a b c\}. \quad (21)$$

② Preliminary localization of unknown node  $u$ :

$$\begin{cases} x_{Ak}^2 - x_{Ck}^2 - 2x_u(x_{Ak} - x_{Ck}) + y_{Ak}^2 - y_{Ck}^2 - 2y_u(y_{Ak} - y_{Ck}) = d_{Au}^2 - d_{Cu}^2, \\ x_{Bk}^2 - x_{Ck}^2 - 2x_u(x_{Bk} - x_{Ck}) + y_{Bk}^2 - y_{Ck}^2 - 2y_u(y_{Bk} - y_{Ck}) = d_{Bu}^2 - d_{Cu}^2. \end{cases} \quad (23)$$

So,  $S_k = \min\{\dots S_i \dots\}$ ; assuming that the three vertices are  $A_k$ ,  $B_k$ , and  $C_k$  of the triangle whose area is  $S_k$ , then the sensor nodes  $A_k(x_{Ak}, y_{Ak})$ ,  $B_k(x_{Bk}, y_{Bk})$ , and  $C_k(x_{Ck}, y_{Ck})$  are regarded as the localization unit of unknown sensor node  $u$ .

(4) Estimate the Euclidean distance of the unknown node  $u$  to beacon nodes  $A_k$ ,  $B_k$ , and  $C_k$ ; then, preliminarily locate node  $u$ .

① Estimate the Euclidean distance of the node  $u$  to beacon nodes  $A_k$ ,  $B_k$ , and  $C_k$ .

The sensor node  $u$  is inside triangle which consists of the beacon nodes which are in the localization unit, improving the single hop size in DV-hop sensor node localization method; in the case of only considering the distances and the number of hops between beacon nodes in the localization unit, calculate the local single hop size to estimate the distance. Moreover, because the topological triangle area constituted by the localization unit of unknown node  $u$  is the smallest among all triangles meeting (12), the calculation method of local single hop size can greatly improve the accuracy of distance estimation. The calculation method of local single hop size and distance estimation is as follows:

$$\begin{cases} (x_{Ak} - x_u)^2 + (y_{Ak} - y_u)^2 = d_{Au}^2, \\ (x_{Bk} - x_u)^2 + (y_{Bk} - y_u)^2 = d_{Bu}^2, \\ (x_{Ck} - x_u)^2 + (y_{Ck} - y_u)^2 = d_{Cu}^2. \end{cases} \quad (22)$$

Through carrying out the maximum likelihood estimation of (22), get

Transform (23) into a matrix:

$$B = \begin{bmatrix} x_{Ak}^2 - x_{Ck}^2 + y_{Ak}^2 - y_{Ck}^2 + d_{Cu}^2 - d_{Au}^2 \\ x_{Bk}^2 - x_{Ck}^2 + y_{Bk}^2 - y_{Ck}^2 + d_{Cu}^2 - d_{Bu}^2 \end{bmatrix}, \quad (24)$$

$$A = \begin{bmatrix} 2(x_{Ak} - x_{Ck}) & 2(y_{Ak} - y_{Ck}) \\ 2(x_{Bk} - x_{Ck}) & 2(y_{Bk} - y_{Ck}) \end{bmatrix}. \quad (25)$$

According to the least square method,

$$X = (A^T A)^{-1} A^T B, \quad (26)$$

$$X = \begin{bmatrix} x_u \\ y_u \end{bmatrix}. \quad (27)$$

- (5) Promote node  $u$  to anchor node and give it a low credibility, in the next round of localization; when anchor nodes are insufficient or the  $K$ -value of all localization units does not meet (12), node  $u$  will participate in locating as a beacon node.
- (6) If the execution time of the localization algorithm is less than  $T_{\max}$ , then select the next unknown node and execute step (1). Otherwise, the loop ends. When the execution times of the localization algorithm are equal to  $T_{\max}$ , there are still unknown nodes that have not been located; then, regard this part of unknown nodes as boundary nodes and execute step (7).
- (7) Dealing with boundary nodes, diffuse the beacon nodes in one-hop range of the boundary node by using

$$\begin{cases} x_i = a \cdot r \cos(2\pi b) + x_{Ai}, \\ y_i = a \cdot r \cos(2\pi b) + y_{Ai}, \end{cases} \quad (a < 1, b < 1). \quad (28)$$

The beacon node's original coordinate is  $(x_{Ai}, y_{Ai})$ , and  $(x_i, y_i)$  is the random coordinate after diffusion. After diffusion, reexecute the above DC- $K$  algorithm to infer the coordinate of the boundary nodes.

So far, the localization algorithm of the first stage has been completed.

### 3. Improved Grey Wolf Optimization Algorithm

The Grey Wolf Optimizer (GWO) is a swarm intelligence optimization algorithm. The paper [15] simulated the hierarchy and predation activities of wolves, proposing and analyzing the GWO algorithm. This algorithm has the characteristics of having a few parameters, strong convergence, and easy implementation. The principle of the GWO algorithm is as follows: all grey wolves were divided into groups of  $\alpha$ ,  $\beta$ ,  $\delta$ , and  $\omega$ ; according to the level of the social hierarchy, when hunting prey,  $\alpha$ ,  $\beta$ , and  $\delta$  will guide the lowest level of  $\omega$  wolf to search for prey, and the location of  $\alpha$  wolf is the final result at the end of hunting.

The classical DV-hop algorithm uses the least square method to get the solution of (22), and according to the estimated distance this completes the localization calculation

[16]. In this paper, in the second stage, the three beacon nodes in the localization unit and the estimated distance obtained by DC- $K$  algorithm participate in the iterative calculation, and the improved GWO is used to obtain the optimal solution of the (22). In order to reduce the impact of the Euclidean distance estimation error in the first stage on the final localization result, in this paper, the distance  $(d_{Au}, d_{Bu}, d_{Cu})$  of the beacon nodes and the sensor node  $u$  in the localization unit is included in the vector of each individual for synchronous iteration, according to the vector to design the fitness function. This method can not only solve the influence of distance estimation errors on the final localization results, but also add constraints of the fitness function, thereby greatly reducing the times of iterations [17]. In this paper, the specific steps of applying the improved grey wolf optimization algorithm to wireless sensor node location are as follows:

- (1) The coordinates of the unknown sensor nodes in the first stage and the three distances to the beacon nodes of the localization unit constitute such a 5-dimensional vector:  $[x_u, y_u, d_{Au}, d_{Bu}, d_{Cu}]$ .
- (2) Calculating the fitness function about each individual wolf,

$$\text{Fitness}_u = \sum_i \left( \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2} - d_{iu} \right)^2, \quad (29)$$

$(x_u, y_u)$  is the coordinate of the sensor node  $u$ ,  $(x_i, y_i)$  is the coordinate of the beacon node in localization unit of the unknown node  $u$ , and  $d_{iu}$  is the estimated Euclidean distance of the beacon node to unknown sensor node which is in localization unit. The three unknown nodes with the smallest fitness are  $\alpha$  wolf,  $\beta$  wolf, and  $\delta$  wolf, respectively, and the rest are  $\omega$  wolves.

- (3) The iterative model is as follows ( $\max = 20$ ):

$$\begin{cases} a = 2 - 2 \frac{t}{\max}, \\ C = 2r_1, \\ A = 2ar_2 - a, \end{cases} \quad (30)$$

$$\begin{cases} X_1 = X_\alpha(t) - A_1 \left[ |C_1 X_\alpha(t) - X_\omega(t-1)| \right], \\ X_2 = X_\beta(t) - A_2 \left[ |C_2 X_\beta(t) - X_\omega(t-1)| \right], \\ X_3 = X_\delta(t) - A_3 \left[ |C_3 X_\delta(t) - X_\omega(t-1)| \right], \end{cases} \quad (31)$$

$$X_\omega(t) = (X_1 + X_2 + X_3). \quad (32)$$

$A$  is the convergence factor;  $r_1, r_2$  are random numbers in  $[0, 1]$ ; and  $C$  is the swing factor;  $X(t)$  is the vector of the wolf when the times of iteration are equal to  $t$ ;  $X(t-1)$  is the vector of the wolf when the times of iteration are equal to  $t-1$ . In each iteration, the location of  $\alpha$  wolf changes according to

(30)–(32). The process of  $\omega$  wolf location changes is the process of hunting, which is the process of looking for the optimal solution.

- (4) After the iteration of step (3) is completed, the five-dimensional solution vector of the  $\alpha$  wolf is output and the solution vector is  $[x_\alpha, y_\alpha, d_{A\alpha}, d_{B\alpha}, d_{C\alpha}]$ ;  $(x_\alpha, y_\alpha)$  in the solution vector is taken as the ultimate location of the unknown sensor node  $u$ .
- (5) Back to step (2), calculate the fitness value of all individuals again according to (29) and select the new  $\alpha$  wolf,  $\beta$  wolf, and  $\delta$  wolf. Until the output of all individuals (the localization of all unknown nodes is completed), the algorithm ends.

#### 4. Experimental Results and Analysis

The experiment was conducted with Windows10 computer operating system, and the simulation experiment was carried out with matlab2016a. Randomly deploy 100 sensor nodes in a  $100 \times 100$  square area, compare and analyze the performance of the wireless sensor node localization algorithm from different communication radius and beacon node ratio, and compare with the DC-H, DC-A, GWO, and DV-hop. Average localization error is defined as

$$\text{average error} = \frac{1}{T} \sum_u \sqrt{(x - x_u)^2 + (y - y_u)^2} \quad (33)$$

$T$  is the time of the experiment;  $(x, y)$  is the actual coordinate of the sensor node  $u$ ;  $R$  is the communication radius; the amount of sensor nodes is  $NAmount$ , located in this network;  $BAmount$  is the number of beacon sensor nodes. When the beacon sensor node ratio is 20%, the distribution diagram about the nodes is shown in Figure 10.

**4.1. Influences of Beacon Node Ratio on Algorithm.** Under the same communication radius ( $R = 35$  m), the influence of different beacon node ratio on each algorithm is shown in Figure 11:

- (1) As shown in the figure, compared with DV-hop, the localization results accuracy of the other four localization algorithms is higher, especially when the beacon node ratio is more than 20%.
- (2) Using the localization method of intelligent optimization algorithm GWO, when beacon sensor node ratio is 10%, the average localization error is more than 50%.
- (3) The two-stage GWO algorithm has better localization accuracy than other algorithms and solves the problem that GWO algorithm cannot converge when the beacon node ratio is low. As shown in Figure 10, although the localization accuracy of DC-A and DC-H algorithms is less affected by the anchor node ratio, the localization error is large. This is because once the best localization unit in DC-A or DC-H is founded, all unknown nodes will use it to locate, and the coordinates of other beacon nodes in this network cannot be effectively utilized, resulting in a large localization error [18]. In this paper,

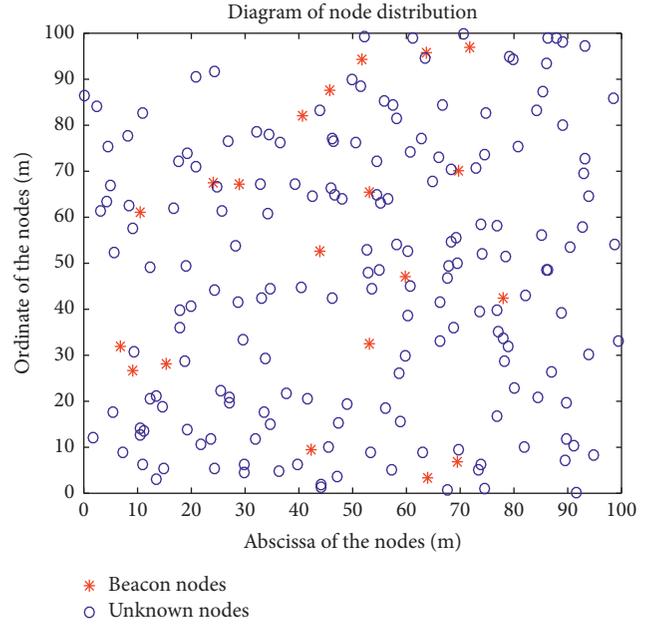


FIGURE 10: Diagram of node distribution.

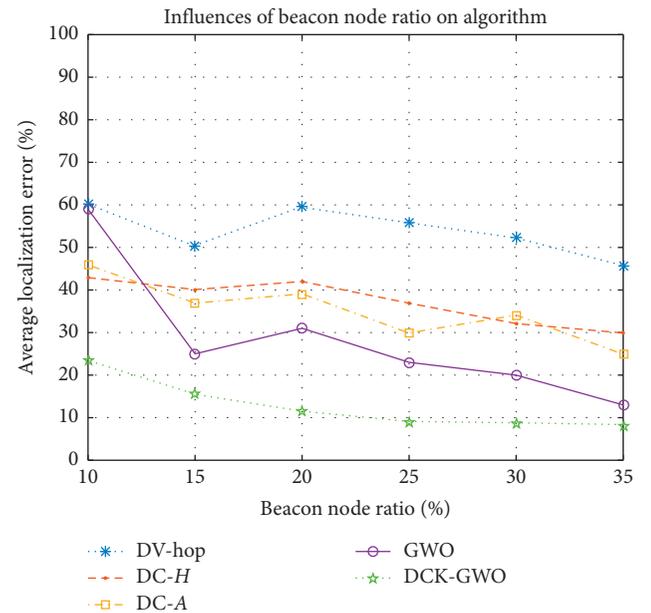


FIGURE 11: Average localization error with different beacon node ratio.

based on the APIT algorithm, we came up with the DC-K algorithm, by controlling the unknown nodes in the topological triangle formed by beacon nodes in the localization unit, which solves this problem effectively.

**4.2. Influences of Communication Radius on Algorithm.** Under the same beacon node ratio ( $\text{Beacon}/100 = 25\%$ ), the influence of different communication radius on the localization result of each algorithm is shown in Figure 12:

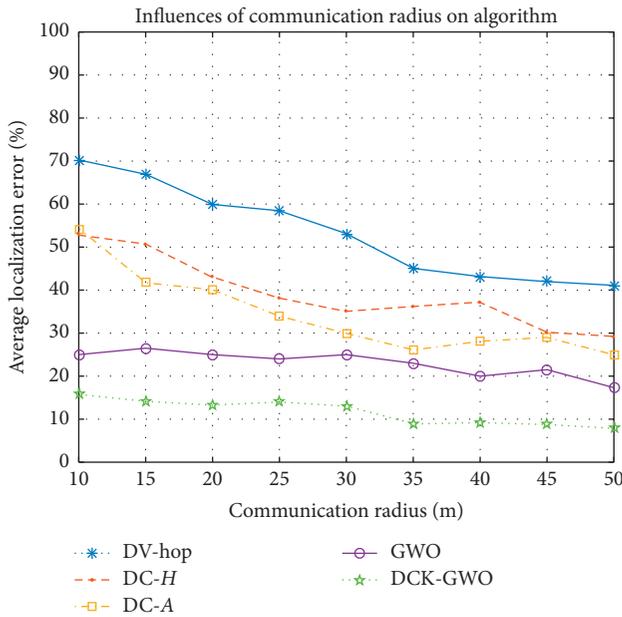


FIGURE 12: Average error with the different communication radius.

- (1) As shown in Figure 11, the DV-hop's localization error decreases when the communication radius increases; hence, the DV-hop's performance is greatly affected by the communication radius. Because of using the distance estimation method of single hop size and hops, when the communication radius decreases, the minimum hops from beacon nodes to unknown node increase, resulting in a large distance estimation error.
- (2) The DC-A and DC-H were improved in the localization stage but still fully use the distance estimation method of DV-hop localization method in the first stage. Therefore, the localization accuracy will be greatly affected by the communication radius. Because of the shortcomings of the collinearity algorithm, the overall localization error is still large.
- (3) Although GWO algorithm is less affected by the communication radius, the fitness function lacks constraints, resulting in the poor localization accuracy.
- (4) With the same beacon node ratio, the two-stage degree of K-value collinearity GWO algorithm integrates the advantages of DC-A, DC-H, and GWO, reducing the impact of node communication radius on the performance of the algorithm, and improves the localization accuracy.

## 5. Conclusion

In order to increase the localization accuracy of wireless sensor nodes, this paper studies the shortcomings of the existing collinearity algorithm and the disadvantages of intelligent optimization algorithm in WSN localization and proposes a two-stage wireless sensor grey wolf optimization node location algorithm based on K-value collinearity. Based

on the APIT algorithm and putting forward the concept of DC-K, organically combined with the improved grey wolf optimization algorithm, this not only makes up for the shortcomings of previous DC algorithms, but also solves the problem that intelligent optimization localization algorithm is easy to fall into the local optimization and reduces the times of iteration. The experimental results show that the two-stage DC-K localization algorithm with the better localization accuracy and strong robustness can greatly decrease the calculation energy consumption in the localization process and extend the service life of the sensor nodes.

The DC-K algorithm is suitable for the regions with random distribution of nodes. However, in concave regions, the shortest distance between sensor nodes is easily affected by concave boundaries, which will increase the error of distance estimation and affect the localization accuracy. In the next research, the localization algorithm with higher accuracy in concave regions will be studied; in the second stage, dynamic changes or other improvements to the weight factor of prey localization will be considered.

## Data Availability

The experimental code used to support the findings of this study is available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grants nos. 61501405 and 61771432) and the Science and Technology Planning Program of Henan Province (no. 202102210398).

## References

- [1] K. Qian, C. Bu, and Y. Wang, "An iteration localization algorithm in wireless sensor network based on intersection degree ratio," *Journal of Beijing University of Posts and Telecommunications*, vol. 42, no. 3, pp. 98–105, 2019.
- [2] Q. Shi, Q. Xu, and J. Zhang, "Improvement for DV-hop based on distance correcting and grey wolf optimization algorithm," *Journal of Transduction Technology*, vol. 32, no. 10, pp. 1549–1555, 2019.
- [3] J. Zhu and L. Song, "Hierarchical location algorithm in wireless sensor network," *Computer Engineering and Applications*, vol. 53, no. 22, pp. 61–65, 2017.
- [4] G. Lou and J. Zhang, "Ranging distance modified by particle swarm algorithm for WSN node localization," *Journal of Jilin University (Science Edition)*, vol. 56, no. 3, pp. 650–656, 2018.
- [5] Z. Zhang and L. Zhang, "Research on DV-Hop algorithm based on quantum particle swarm optimization," *Computer Technology and Development*, vol. 28, no. 5, pp. 81–85, 2018.
- [6] H. liang, "Improved wireless sensor network localization algorithm based on aggregation, collinearity and connectivity of

- anchor nodes,” *Journal of Computer Applications*, vol. 37, no. 2, pp. 427–431, 2017.
- [7] L. Wu, Q. Meng, and H. Liang, “A collinearity-based localization algorithm for wireless sensor network,” *Journal of Transduction Technology*, vol. 22, no. 5, pp. 722–727, 2009.
- [8] C. Poggi and G. Mazzini, “Collinearity for sensor network localization in ad-hoc sensor networks,” 2003.
- [9] W. Yi, “WSN node location optimization based on second-stage differential evolution,” *Computer Measurement and Control*, vol. 27, no. 8, pp. 286–290, 2019.
- [10] T. He, C. Huang, and B. M. Bulm, *Range Free Localization Schemes for Large Scale Sensor Networks*, in *Proceeding of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom 2003)*, pp. 81–95, San Diego, CA, USA, September 2003.
- [11] Y. Meng, *Research on Node Localization Methods for Wireless Sensor Network*, Northeastern University, Boston, MA, USA, 2014.
- [12] X. Zhang and X. Wang, “Comprehensive review of wolf optimization algorithm,” *Computer Science*, vol. 46, no. 3, pp. 30–38, 2019.
- [13] E. Erdemir and T. E. Tuncer, “Path planning for mobile-anchor based wireless sensor network localization: static and dynamic schemes,” *Ad Hoc Networks*, vol. 77, no. 8, pp. 1–10, 2018.
- [14] S. S. Banihashemian, F. Adibnia, and M. A. Sarram, “A new range-free and storage-efficient localization algorithm using neural networks in wireless sensor networks,” *Wireless Personal Communications*, vol. 98, no. 1, pp. 1547–1568, 2018.
- [15] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Advances in Engineering Software*, vol. 69, pp. 46–61, 2003.
- [16] N. Bulusu, J. Heidemann, and D. Estrin, “GPS-less low-cost outdoor localization for very small devices,” *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, 2000.
- [17] W. Zhang, X. Yang, and Q. Song, “Improved DV-Hop algorithm based on artificial bee colony,” *International Journal of Control and Automation*, vol. 8, no. 11, pp. 135–144, 2015.
- [18] W. Ben, W. Zheng, and F. Jiang, “Self-Localization algorithm in wireless sensor networks based on node promotion strategy,” *Command Information System and Technology*, vol. 7, no. 1, pp. 63–68, 2016.