Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6292629, 20 pages
https://doi.org/10.1155/2022/6292629

Research Article

Localization for Wireless Sensor Networks Assisted by Two Mobile Anchors with Improved Grey Wolf Optimizer

Huanqing Cui,[1] Junyi Zhao,[1] Chuanai Zhou,[2] and Na Zhang[1]

[1]*College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China*
[2]*College of Business, Qingdao Binhai University, Qingdao 266555, China*

Correspondence should be addressed to Huanqing Cui; cuihq@sdust.edu.cn

Received 21 June 2022; Revised 30 November 2022; Accepted 13 December 2022; Published 29 December 2022

Academic Editor: Javier Prieto

Copyright © 2022 Huanqing Cui 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.

Localization is crucial to wireless sensor networks. Among the recently proposed localization algorithms, the mobile anchor-assisted localization (MAL) algorithm seems promising. A MAL algorithm using a single mobile anchor has low energy consumption but a high localization error. Conversely, a MAL algorithm with three or more mobile anchors has minor localization errors but high energy consumption. By balancing energy consumption and localization accuracy, our study developed a localization algorithm assisted by two mobile anchors. A mobile anchor traverses the network along a double anchor SCAN (DASCAN) path, which divides the deployment region into grids and requires the two mobile anchors to traverse different horizontal lines in a zigzag pattern. Sensor nodes estimate their locations using a multiple-disturbance strategy grey wolf optimization (MDS-GWO) algorithm, which improves optimization by introducing a nonlinearly decreasing weight, a random perturbation of grey wolves and a mirror grey wolf. Using MATLAB, DASCAN was compared with GTURN, GSCAN, PP-MMAN, H-Curves, M-Curves, and SCAN paths by their energy consumption and localization rates. The localization error of MDS-GWO was compared with trilateration, PSO, WOA, and GWO. The impacts of radio irregularity, radio radius, and fading effect on MDS-GWO with different paths were also analyzed. The simulation results showed that the energy consumption of DASCAN was, on average, 30.1% less than GSCAN, GTURN, and PP-MMAN, but they had almost the same localization accuracy. The energy consumption of DASCAN was an average of 18.67% more than M-Curves, H-Curves, and SCAN, but the localization error of DASCAN was average of 32.3% less than SCAN, H-Curves, and M-Curves. The localization error of MDS-GWO was average of 25.5% less than trilateration, PSO, WOA, and GWO. Moreover, the performance of the proposed algorithm was less affected by different setups than the compared methods.

1. Introduction

The wireless sensor network (WSN) consists of a large amount of sensor nodes deployed in a given region of interest (ROI). It is widely applied to environmental surveys, habitat monitoring, medical diagnosis, and disaster rescue. Node localization is a key technology of WSNs. However, equipping each sensor node with a global navigation satellite system (GNSS) device is impractical due to cost and energy consumption constraints. Only a few sensor nodes, called *anchors,* know their positions. Other sensor nodes, *unknown nodes,* are localized with the help of *beacons* broadcast by anchors.

More anchors result in higher localization accuracy, but the anchors are more expensive than unknown nodes, so the mobile anchor-assisted localization (MAL) algorithms have attracted considerable interest. MAL algorithms require the mobile anchor to traverse the ROI along a given path and broadcast beacons periodically. The locations where the mobile anchor broadcasts beacons are called *virtual anchors*. If the MAL algorithm only uses one mobile anchor, obtaining high localization accuracy with a short path is difficult. If the MAL algorithm uses three or more mobile anchors, localization becomes more accurate, but at the cost of high energy consumption. This paper proposes a MAL algorithm that uses two mobile anchors, balancing energy consumption and localization accuracy. Our algorithm uses a double anchor scan (DASCAN) as the moving path and a multiple disturbance strategy for the grey wolf optimizer (MDS-

GWO) to estimate the locations of unknown nodes. The contributions of our study are:

(1) It presents the DASCAN path for two mobile anchors, reducing the number of beacons and path length. This allows the two anchors to scan adjacent rows of ROI simultaneously so that three neighboring virtual anchors form an equilateral triangle.

(2) It presents an improved GWO algorithm with multiple disturbance strategies to estimate the locations of unknown nodes, which can improve localization accuracy.

(3) It reports extensive experiments that compared the performance of the proposed algorithm with similar current algorithms. DASCAN was compared with GTURN, GSCAN, PP-MMAN, H-Curves, M-Curves, and SCAN paths in terms of energy consumption and localization rate. MDS-GWO was compared with trilateration, particle swarm optimization (PSO), whale optimization algorithm (WOA), and grey wolf optimizer (GWO) regarding localization errors. The impacts of radio irregularity, radio radius, and fading effect on MDS-GWO with different paths were also analyzed.

The rest of this paper is organized as follows. Section 2 introduces related work, and Section 3 presents the problem statements and performance metrics. Section 4 describes the proposed algorithm, including the DASCAN path and MDS-GWO algorithm. Section 5 summarizes the simulations and analysis. Section 6 concludes the paper. Table 1 lists the acronyms used in this paper and their definitions.

2. Related Works

The MAL algorithm needs to define a moving trajectory for a mobile anchor, and the location algorithm to estimate the location of unknown nodes. The path-planning algorithm may be dynamic or static [1]. The dynamic form determines the path according to the distribution of unknown nodes and environments, which may not cover all unknown nodes and requires additional hardware support. The static form determines the path in advance and requires the mobile anchor to move along a given path during localization, which is cheaper than the dynamic approach. Song et al. [2] proposed a path for self-adaptive anchors based on the Gaussian–Markov model, and they applied an alternating minimization algorithm to estimate the positions of unknown nodes. However, the random path may not cover the whole ROI. SCAN, Double-SCAN, and Hilbert proposed in [3] are simple, but their parameters must be carefully designed to avoid collinearity. H-curve [4] is an H-shaped path that generates collinear beacons. SLMAT (a mobile anchor node based on trilateration and scan) [5] ensures that each unknown node is covered by an equilateral triangle formed by beacons. ∑-SCAN [6] combines SCAN and zigzag paths to achieve high localization accuracy and cover the ROI with a short path. M-curve [7] is an M-shaped path and applies dolphin optimization algorithm to localize unknown nodes, but it may not localize unknown nodes near the borders of ROI. The static search-and-decide (SSD) and dynamic search-and-decide (DSD) [8] paths are, respectively, static and dynamic, which consist of a search phase and a decision phase. In the first phase, SSD visits a subset of virtual anchors to determine the grids occupied by unknown nodes. In the decision phase, the mobile anchor revisits the grids containing unknown nodes for localization. DSD differs from SSD in the second phase: It generates anchors based on perpendicular bisectors. The localization accuracy of these MAL algorithms with a single mobile anchor is low, but they consume comparatively little energy. GSCAN and GTURN [9] use three mobile anchors. The mobile anchors in GSCAN repeatedly broadcast beacons at the same location; GTURN avoids this problem using a longer path. However, GSCAN and GTURE need to use two boundary strategies to guarantee that they can cover the whole ROI. PP-MMAN [10] also uses three mobile anchors, but it requires them to move horizontally simultaneously to overcome the drawbacks of GSCAN and GTURN. These MAL algorithms using multiple mobile anchors improve localization accuracy at the cost of high energy consumption. The enhanced RSSI-based tree-climbing mechanism (ERTC) [11] requires the mobile anchor to be equipped with both omnidirectional and directional antennae, where the omnidirectional antenna broadcasts the message and directional antennas to receive messages of sensor nodes. It identifies the trajectory of the mobile anchor with the virtual force of unknown nodes in the network, and it uses a circumcenter algorithm to localize unknown nodes. It is more complicated than the other MAL algorithms and needs additional hardware support.

The location of unknown nodes can be estimated as an optimization problem. Nowadays, swarm intelligence optimization has been widely applied in localization due to its low complexity and easy implementation. H-Best PSO [12] is an improved PSO algorithm to estimate the locations of unknown nodes where the mobile anchor traverses along a Hilbert curve. Song et al. [13] proposed a localization algorithm based on glowworm swarm optimization of a hybrid chaotic strategy to control the moving distance of each firefly by chaos mutation and chaotic inertial weight when the firefly falls into a local optimum. The algorithm proposed in [14] reduces the hop distance error by leading to the average hop distance error correction value. It applies differential evolution to optimize the localization result of the unknown node. Considering the impact of obstacles, the algorithm proposed in [15] divides the anchors into multiple groups. If the anchors of a group cannot accurately localize the unknown node, the other anchors in the nearby groups are used to assist localization. In [16], WOA is applied to optimize the relationship between RSSI (received signal strength indicator) and signal transmission distance to improve localization accuracy.

Because the traversal of mobile anchors forms some specific curves, the location of unknown nodes can be estimated by some geometry principles. For example, PI (perpendicular intersection) [17] is based on the perpendicular

Table 1: List of acronyms.

Acronym	Definition
WSN	Wireless sensor network
ROI	Region of interest
GNSS	Global navigation satellite system
MAL	Mobile-anchor-assisted localization
GWO	Grey wolf optimizer
DASCAN	Double anchor scan
MDS-GWO	Multiple-disturbance strategy grey wolf optimization
SLMAT	Mobile anchor node based on trilateration and scan
SSD	Static search-and-decide
DSD	Dynamic search-and-decide
PP-MMAN	Path planning method for multiple mobile anchor nodes
ERTC	Enhanced RSSI-based tree-climbing mechanism
PSO	Particle swarm optimization
WOA	Whale optimization algorithm
PI	Perpendicular intersection
RMSE	Root mean square error
RSSI	Received signal strength indicator
DOI	Degree of irregularity
CRLB	Cramer-Rao lower bound

intersection principle, which utilizes the geometric relationship of a perpendicular intersection to compute the locations of unknown nodes. The algorithm proposed in [18] is based on the principle that the perpendicular bisector of a chord of a circle passes through the center of the circle, so the unknown node is localized by two chords constructed by a mobile anchor. These methods are easy to implement, but they heavily depend on the mobile anchor's path and the density of virtual anchors.

The MAL algorithm with one mobile anchor has difficulty obtaining high localization accuracy with low energy consumption, while the MAL algorithm using three or more mobile anchors has high localization accuracy but high energy consumption. To balance energy consumption and localization accuracy, this paper proposes a MAL algorithm with two mobile anchors. DASCAN produces a path that is as short as possible and provides sufficient virtual anchors to each unknown node. Compared with the other evolutionary algorithms, GWO has high optimization efficiency and is simple to implement because of fewer control parameters. MDS-GWO makes the best of these advantages and improves its ability to jump out of local optima.

3. Problem Statements and Performance Metrics

3.1. Problem Statement. A WSN consists of a set of unknown nodes $\{U_i | i = 1, 2, \cdots, N\}$ randomly deployed in an ROI with height H and length L. The two mobile anchors are denoted by MA_1 and MA_2. Let (ux_i, uy_i) be the actual location of U_i, and $(\widehat{ux}_i, \widehat{uy}_i)$ be its estimated location, denoted by $\widehat{U}_i$. The proposed algorithm applies RSSI to measure the distance between the mobile anchor and the unknown node. The RSSI-based ranging technique is based on

$$P_r(d) = P_0(d_0) - \eta 10 \log\left(\frac{d}{d_0}\right) + X_\sigma, \tag{1}$$

where $P_r(d)$ is the received power at distance d, $P_0(d_0)$ denotes the received power at reference distance d_0, η is the path-loss exponent, and X_σ is a log-normal random variable with variance σ^2 to account for fading.

Ideally, the communication range of a sensor node is a circle with radius R. However, signal propagation is easily affected by the environment. This paper uses the degree of irregularity (DOI) to represent communication irregularity. DOI is defined as:

$$K_i = \begin{cases} 1 & i = 0 \\ K_{i-1} \pm r_1 \times \text{DOI} & 0 < i < 360 \end{cases}, \tag{2}$$

where K_i represents the DOI in the i-th direction and satisfies $|K_0 - K_{359}| \leq \text{DOI}$, and r_1 is a random number in (0,1).

Let the neighboring virtual anchors of unknown node U_i be $\{V_j | j = 1, 2, \cdots, M_i\}$ where M_i is the number of neighboring virtual anchors. The objective of localization is formulated as

$$\min f(x, y) = \frac{1}{M_i} \sum_{j=1}^{M_i} \left| \sqrt{\left(x - vx_j\right)^2 + \left(y - vy_j\right)^2} - d_{ij} \right|, \tag{3}$$

where (vx_j, vy_j) is the location of V_j, and d_{ij} is the measured

Table 2: Symbol list.

Symbol	Meaning
U_i	i -th unknown node
N	Number of unknown nodes
MA_1, MA_2	Two mobile anchors
(ux_i, uy_i)	Actual position of U_i
$(\widehat{ux}_i, \widehat{uy}_i), \widehat{U}_i$	Estimated position of U_i
η	Path-loss exponent
σ	Variance of fading effect
R	Radio radius of sensor nodes
DOI	Degree of irregularity
$\left(vx_j, vy_j\right)$	Position of j -th virtual anchor
LR	Localization rate
e	Average localization error
RMSE	Root mean square error of localization error
$\mathrm{RMSE}_{\mathrm{CRLB}}$	RMSE of localization error variance
E_{va}	Energy consumption for the mobile anchor to broadcast a beacon
E_{len}	Energy consumption for the mobile anchor to move a meter
MaxIter	Number of iterations of MDS-GWO
l	Distance between two neighboring virtual anchors horizontally

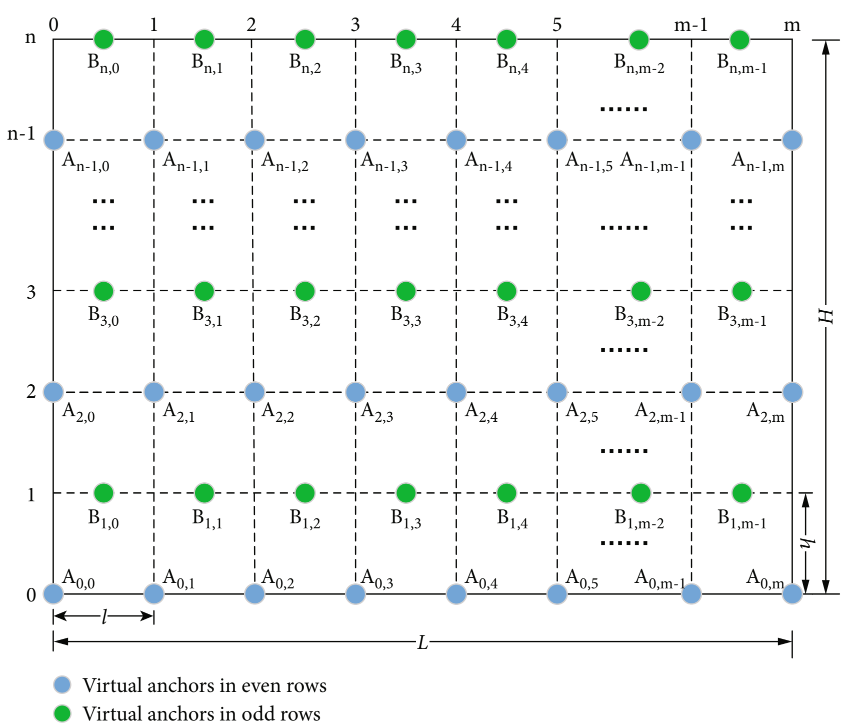


Figure 1: Virtual anchors in the deployment area.

```
Input: H, L, h, l.
Output: List_1, List_2: The lists of virtual anchors of MA_1 and MA_2.
List_1 ⟵ {A_{0,0}}, List_2 ⟵ {B_{1,0}}
flag ⟵ 1, i ⟵ 0, j ⟵ 1, k ⟵ 0
While (i < n − 1)
  If(flag = 1)
    List_1 ⟵ List_1 ∪ {A_{i,1}, A_{i,2}, ⋯, A_{i,m}, B_{i+1,m−1}}
    List_2 ⟵ List_2 ∪ {B_{i+1,0}, B_{i+1,1}, ⋯, B_{i+1,m−2}, B_{i+1,m−2}, A_{i+2,m}}
    flag ⟵ 2
  Else if(flag = 2)
    List_1 ⟵ List_1 ∪ {A_{i,m−1}, A_{i,m−2}, ⋯, A_{i,0}, B_{i+1,0}}
      List_2 ⟵ List_2 ∪ {B_{i+1,m−1}, B_{i+1,m−2}, ⋯, B_{i+1,1}, B_{i+1,1}, A_{i+2,0}}
      flag ⟵ 1
    End if
    i ⟵ i + 2
  End while
  If(flag = 1)
    If(i = n)
      List_1 ⟵ List_1 ∪ {A_{n,1}, A_{n,2}, ⋯, A_{n,m}}
    Else if(i = n − 1)
      List_1 ⟵ List_1 ∪ {A_{n−1,1}, A_{n−1,2}, ⋯, A_{n−1,m}}
      List_1 ⟵ List_1 ∪ {B_{n,0}, B_{n,1}, ⋯, B_{n,m−1}}
    End if
  Else if(flag = 2)
    If(i = n)
      List_1 ⟵ List_1 ∪ {A_{n,m−1}, A_{n,m−2}, ⋯, A_{n,0}}
    Else if(i = n − 1)
      List_1 ⟵ List_1 ∪ {A_{n−1,m−1}, A_{n−1,m−2}, ⋯, A_{n−1,0}}
      List_1 ⟵ List_1 ∪ {B_{n,m−1}, B_{n,m−2}, ⋯, B_{n,0}}
    End if
  End if
```

Algorithm 1: DASCAN Path.

distance between U_i and V_j. (x, y) is the unknown quantity to be found, and its solution is $(\widehat{ux}_i, \widehat{uy}_i)$.

3.2. Performance Metrics. The localization error of U_i is defined as

$$e_i = \sqrt{\left(\widehat{ux}_i - ux_i\right)^2 + \left(\widehat{uy}_i - uy_i\right)^2}. \quad (4)$$

An unknown node in two-dimensional space can be localized only if it receives at least three noncollinear beacons. Let N_{loc} be the number of localized unknown nodes, and LR be the localization rate, that is

$$LR = \frac{N_{\text{loc}}}{N}. \quad (5)$$

The average localization error is

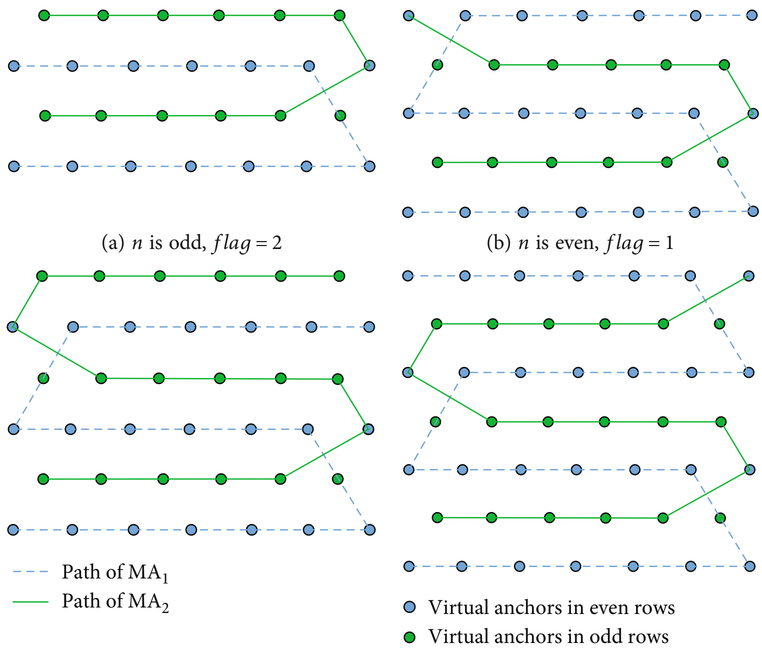
$$e = \frac{1}{N}\sum_{i=1}^{N} e_i. \quad (6)$$

In addition, the root mean square error (RMSE) of localization error is

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} e_i^2}. \quad (7)$$

The Cramer–Rao lower bound (CRLB) provides the lower bound on the variance of any unbiased estimator. This paper uses the CRLB introduced in [19] as the standard benchmark where η is a fixed constant. Let $S = \{s_i | i = 1, 2, \cdots, N + M\}$ be the union set of unknown nodes and virtual anchors, where $\{s_i | i = 1, 2, \cdots, N\}$ is the set of unknown nodes and $\{s_i | i = N + 1, N + 2, \cdots, N + M\}$ is the set of virtual anchors. The real position of s_i is (x_i, y_i), so $(x_i, y_i) = (ux_i, uy_i)$ for $i = 1, 2, \cdots, N$ and $(x_i, y_i) = (vx_i, vy_i)$ for $i = N + 1, N + 2, \cdots, N + M$. Let $(\tilde{x}_i, \tilde{y}_i)$ be the CRLB of error variance of estimating (x_i, y_i); it is proved in [19] that

$$\begin{aligned} \tilde{x}_i &= \left(J_{xx} - J_{xy} J_{yy}^{-1} J_{xy}^{T}\right)_{ii}^{-1}, \\ \tilde{y}_i &= \left(J_{yy} - J_{xy}^{T} J_{xx}^{-1} J_{xy}\right)_{ii}^{-1}, \end{aligned} \quad (8)$$



(a) n is odd, $flag = 2$ (b) n is even, $flag = 1$

(c) n is odd, $flag = 1$ (d) n is even, $flag = 2$

Figure 2: Four types of DASCAN.

where

$$[J_{xx}]_{ij} = \begin{cases} \left(\dfrac{10\eta}{\sigma \log 10}\right)^2 \displaystyle\sum_{k \in H(i)} \dfrac{(x_i - x_k)^2}{\left((x_i - x_k)^2 + (y_i - y_k)^2\right)^2} & i = j \\ -\left(\dfrac{10\eta}{\sigma \log 10}\right)^2 \boldsymbol{I}_{H(i)}(j) \dfrac{(x_i - x_j)^2}{\left((x_i - x_j)^2 + \left(y_i - y_j\right)^2\right)^2} & i \neq j \end{cases},$$

$$[J_{xy}]_{ij} = \begin{cases} \left(\dfrac{10\eta}{\sigma \log 10}\right)^2 \displaystyle\sum_{k \in H(i)} \dfrac{(x_i - x_k)(y_i - y_k)}{\left((x_i - x_k)^2 + (y_i - y_k)^2\right)^2} & i = j \\ -\left(\dfrac{10\eta}{\sigma \log 10}\right)^2 \boldsymbol{I}_{H(i)}(j) \dfrac{(x_i - x_j)\left(y_i - y_j\right)}{\left((x_i - x_j)^2 + \left(y_i - y_j\right)^2\right)^2} & i \neq j \end{cases}, \text{and}$$

$$[J_{yy}]_{ij} = \begin{cases} \left(\dfrac{10\eta}{\sigma \log 10}\right)^2 \displaystyle\sum_{k \in H(i)} \dfrac{(y_i - y_k)^2}{\left((x_i - x_k)^2 + (y_i - y_k)^2\right)^2} & i = j \\ -\left(\dfrac{10\eta}{\sigma \log 10}\right)^2 \boldsymbol{I}_{H(i)}(j) \dfrac{\left(y_i - y_j\right)^2}{\left((x_i - x_j)^2 + \left(y_i - y_j\right)^2\right)^2} & i \neq j \end{cases}. \quad (9)$$

In the abovementioned equations, $H(i)$ is the set of sensor nodes and virtual anchors that make a pairwise observation with s_i. $\boldsymbol{I}_{H(i)}(j)$ is defined as

$$\boldsymbol{I}_{H(i)}(j) = \begin{cases} 1 & \text{if } j \in H(i) \\ 0 & \text{otherwise} \end{cases}. \quad (10)$$

Based on the abovementioned equations, the RMSE of localization error variance is

$$\mathrm{RMSE}_{\mathrm{CRLB}} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(\tilde{x}_i^2 + \tilde{y}_i^2\right)}. \quad (11)$$

The energy consumption of the mobile anchors depends on the path length and number of virtual anchors:

$$\text{Energy} = E_{\mathrm{va}} \times M + E_{\mathrm{len}} \times L, \quad (12)$$

where E_{va} and E_{len} are, respectively, the energy consumption of broadcasting a virtual anchor and moving per meter; M and L are, respectively, the numbers of virtual anchors and path length.

Table 2 presents a list of symbols used in this paper.

4. Proposed Methodology

4.1. DASCAN Path. As Figure 1 shows, ROI is divided into n horizontal rows, numbered $0, 1, 2, \cdots, n$ from bottom to top, and the gap between neighboring rows is h. In addition, ROI is divided into m vertical columns horizontally, numbered $0, 1, 2, \cdots, m$ from left to right, and the gap between two neighboring columns is l.

The virtual anchors on even and odd rows are, respectively, indicated by A and B. The bottom left point of ROI is taken as the coordinate origin, so the locations of virtual anchors are:

(1) $A_{i,j} = (j \times l, i \times h)$, where $i \leq n$ is an even number, and $j = 0, 1, 2, \cdots, m$.

Input: L_i, $MaxIter$, w
Output: $(\widehat{ux}_i, \widehat{uy}_i)$

$V_{i1} \longleftarrow$ the virtual anchor with the largest RSSI in L_i
$V_{i2} \longleftarrow$ the virtual anchor with the second-largest RSSI in L_i
If ($\exists V \in L_i$ satisfying $\triangle V_{i1} V_{i2} V$ is an equilateral triangle)
  $V_{i3} \longleftarrow V$
Else
  $V_{i3} \longleftarrow$ the virtual anchor with the third-largest RSSI in L_i
End if
Get $(\widehat{ux}_i', \widehat{uy}_i')$ by eq. (26)
$SP_i \longleftarrow [\widehat{ux}_i' - l, \widehat{ux}_i' + l] \times [\widehat{uy}_i' - l, \widehat{uy}_i' + l]$
Initialize a grey wolf population in SP_i
Initialize $\vec{A}$, $\vec{C}$, a and P by Eq. (14), Eq. (15), Eq. (20), and Eq. (25)
Calculate the fitness of each wolf by Eq. (3)
$\alpha \longleftarrow$ the best wolf
$\beta \longleftarrow$ the second-best wolf
$\delta \longleftarrow$ the third-best wolf
$t \longleftarrow 0$
While$(t < MaxIter)$
  For *each wolf X*
    Get $X(t+1)$ by Eq. (19)
    $r_8 \longleftarrow$ a random number in [0,1]
    If$(a > P > w \wedge r_8 \geq 0.5)$
      Get $Y(t+1)$ by Eq. (21)
      Get the final $X(t+1)$ by Eq. (22)
    Else if$(P \geq a > w \wedge r_8 \geq 0.5)$
      Get $Z(t+1)$ by Eq. (23)
      Get the final $X(t+1)$ by Eq. (24)
    End if
  End for
  Update $\vec{A}$, $\vec{C}$, a and P by Eq. (14), Eq. (15), Eq. (20), and Eq. (25)
  Calculate the fitness of each wolf by Eq. (3)
  Update α, β, and δ
  $t \longleftarrow t + 1$
End while
Return$(\widehat{ux}_i, \widehat{uy}_i) \longleftarrow \vec{X}_\alpha$

Algorithm 2: Location estimation of U_i.

Table 3: Simulation parameters.

Parameter	Value
H, L	100 m
N	100
E_{va}	1.4 J
E_{len}	0.2 J
η	3.5
MaxIter	100
Number of particles of each optimization algorithm	50
l	10 to 25 m in 5 m steps
R	0.8 l to 1.4 l in 0.2 l steps
DOI	0.05 to 0.2 in 0.05 steps
σ	4

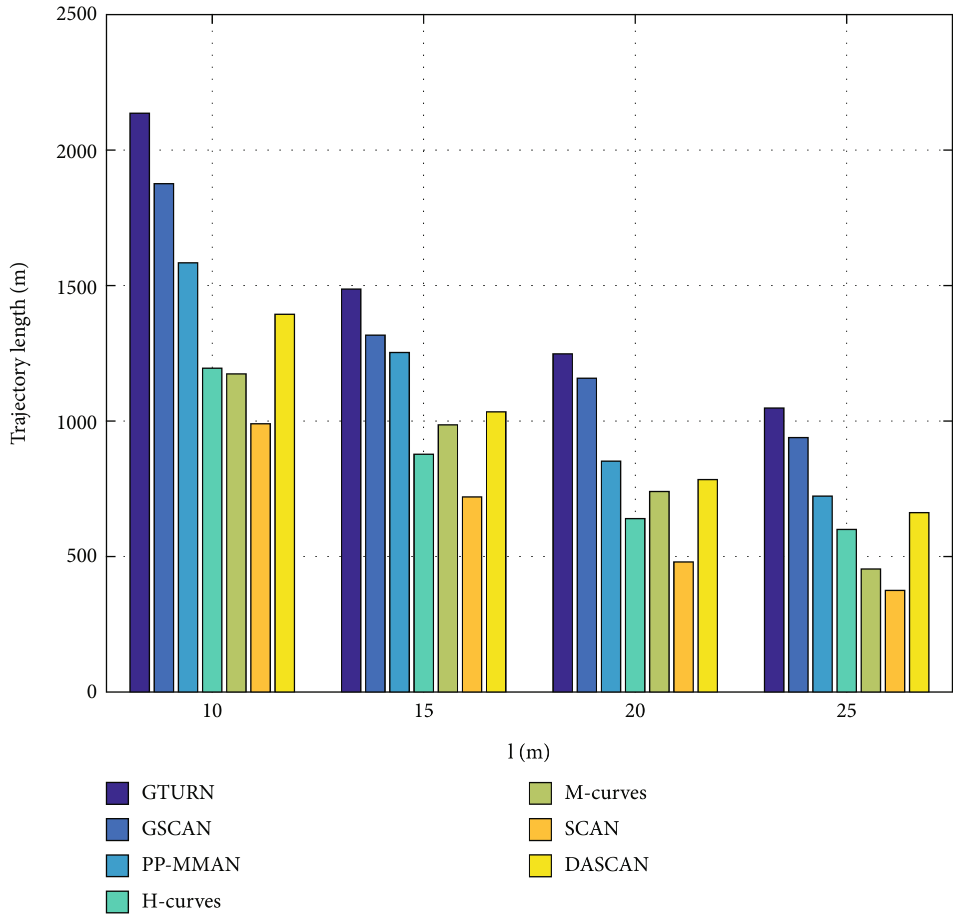


FIGURE 3: Path lengths EZoIZm.

(2) $B_{i,k} = ((k+1/2) \times l, i \times h)$, where $i \le n$ is an odd number, and $k = 0, 1, 2, \cdots, m-1$.

Because the virtual anchors deployed as equilateral triangles can achieve better localization accuracy [9, 10], DASCAN requires $\Delta A_{i,j}A_{i,j+1}B_{i+1,j}$ and $\Delta A_{i,j}A_{i,j+1}B_{i-1,j}$ to be equilateral triangles, so $h = \sqrt{3}l/2$. The points in Figure 1 are virtual anchors. Algorithm 1 shows the process of obtaining DASCAN, where the trajectory of the two mobile anchors is carefully designed to be as short as possible.

In Algorithm 1, List_1 and List_2 are, respectively, the lists of virtual anchors to be traversed by MA_1 and MA_2. Lines 1 and 2 initialize the two lists and parameters. Lines 3–14 generate the two lists when $i < n-1$. The variable flag decides the traversal direction. MA_1 and MA_2 move from left to right if flag = 1; they move from right to left if flag = 2. Lines 15–29 decide List_1 and List_2 when $i \ge n-1$. If $i = n-1$, MA_1 traverses row $n-1$ and MA_2 traverses row n. If $i = n$, MA_1 traverses the top row. Moreover, the traversal direction also depends on flag.

According to different n and flag after the *while* (line 14), DASCAN has four types of paths, as Figure 2 shows.

4.2. Location Estimation Using Improved GWO Algorithm. GWO has a higher optimization efficiency and is easier to implement than other evolutionary algorithms. Its main drawback is that, once it falls into a local optimum, it is difficult for the algorithm to jump out. Thus, we propose the MDS-GWO algorithm to estimate the positions of unknown nodes.

4.2.1. Improved GWO Algorithm. GWO is a swarm intelligence optimization algorithm that imitates a wolf group's hunting behavior. In GWO, the best, the second-best, and the third-best wolves are, respectively, denoted by α, β, and δ. The rest of the wolves are denoted by ω. When hunting, α directs ω to surround, hunt, and attack their prey. In addition, β and δ assist the work of α then capture the prey.

The mathematical model of encircling prey is as follows:

$$\begin{aligned} \vec{D} &= \left| \vec{C} \cdot \vec{X}_p(t) - \vec{X}(t) \right|, \text{and} \\ \vec{X}(t+1) &= \vec{X}_p(t) - \vec{A} \cdot \vec{D}, \end{aligned} \tag{13}$$

where t is the current iteration number, $\vec{X}_p$ is the position vector of prey, and $\vec{X}$ is the position vector of a grey wolf. The coefficient vectors $\vec{A}$ and $\vec{C}$ are

$$\vec{A} = 2\vec{a} \cdot \vec{r}_2 - \vec{a}, \text{and} \tag{14}$$

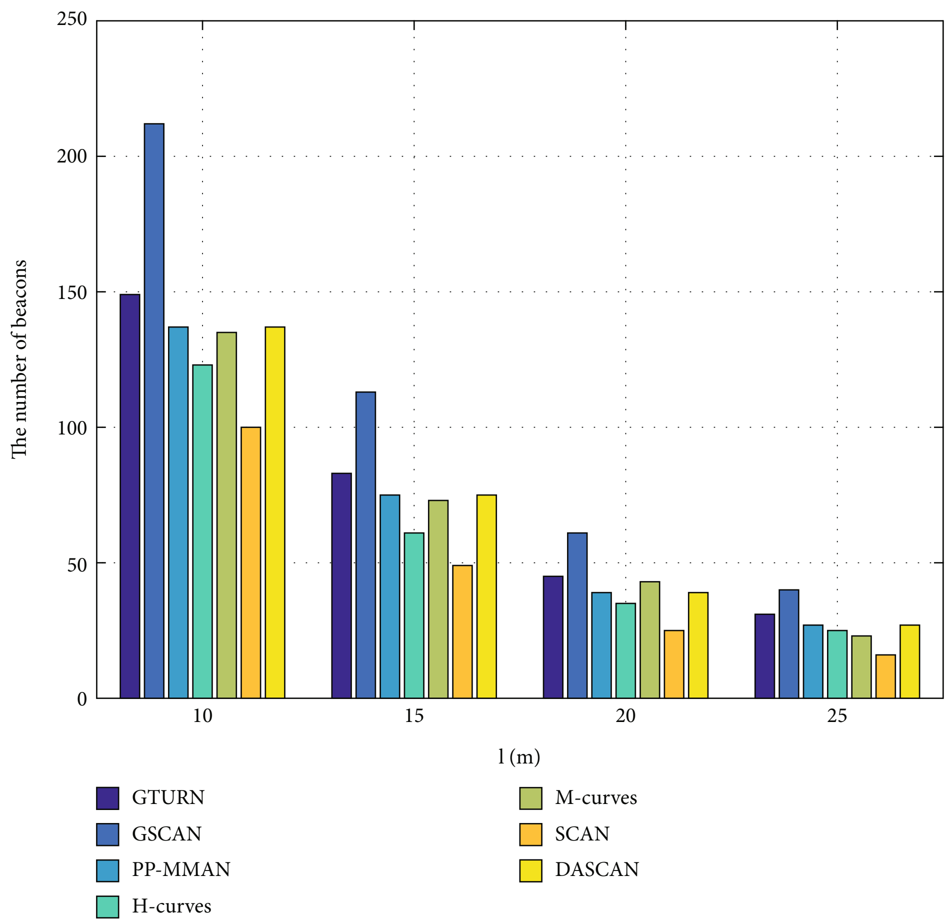


Figure 4: Number of virtual anchors.

$$\vec{C} = 2\vec{r}_3, \tag{15}$$

where $\vec{r}_2$ and $\vec{r}_3$ are random vectors in (0,1), and each element of $\vec{a}$ is

$$a = 2\left(1 - \frac{t}{\text{MaxIter}}\right), \tag{16}$$

where MaxIter is the maximum iteration number.

The mathematical model of hunting prey is as follows:

$$\begin{aligned}\vec{D}_\alpha &= \left|\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}\right|,\\ \vec{D}_\beta &= \left|\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}\right|,\\ \vec{D}_\delta &= \left|\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}\right|,\end{aligned} \tag{17}$$

$$\begin{aligned}\vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1 \cdot \left(\vec{D}_\alpha\right),\\ \vec{X}_2 &= \vec{X}_\beta - \vec{A}_2 \cdot \left(\vec{D}_\beta\right), \text{and}\\ \vec{X}_3 &= \vec{X}_\delta - \vec{A}_3 \cdot \left(\vec{D}_\delta\right),\end{aligned} \tag{18}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}, \tag{19}$$

where $\vec{X}_\alpha$, $\vec{X}_\beta$, and $\vec{X}_\delta$, respectively, represent the position vectors of α, β, and δ in the current iteration; $\vec{D}_\alpha$, $\vec{D}_\beta$, and $\vec{D}_\delta$ are the distances between the current grey wolf and the three best wolves.

Inspired by [20, 21], this paper introduces multiple disturbance strategies into GWO to conquer the drawback of GWO, and this improved GWO algorithm is referred to as MDS-GWO. The improvements are:

(1) *Nonlinear decrease of a*. The convergence speed of the GWO algorithm depends on the speed at which a decreases from 2 to 0. Using Equation (16), a decreases at the same speed, which cannot balance the global and local search, so this paper updates a by

$$a = 2\left(\left(\frac{t}{\text{MaxIter}}\right)^2 - \frac{2t}{\text{MaxIter}} + 1\right). \tag{20}$$

Based on Equation (20), a decreases quickly in the early stage to make GWO converge fast, and a decreases slowly

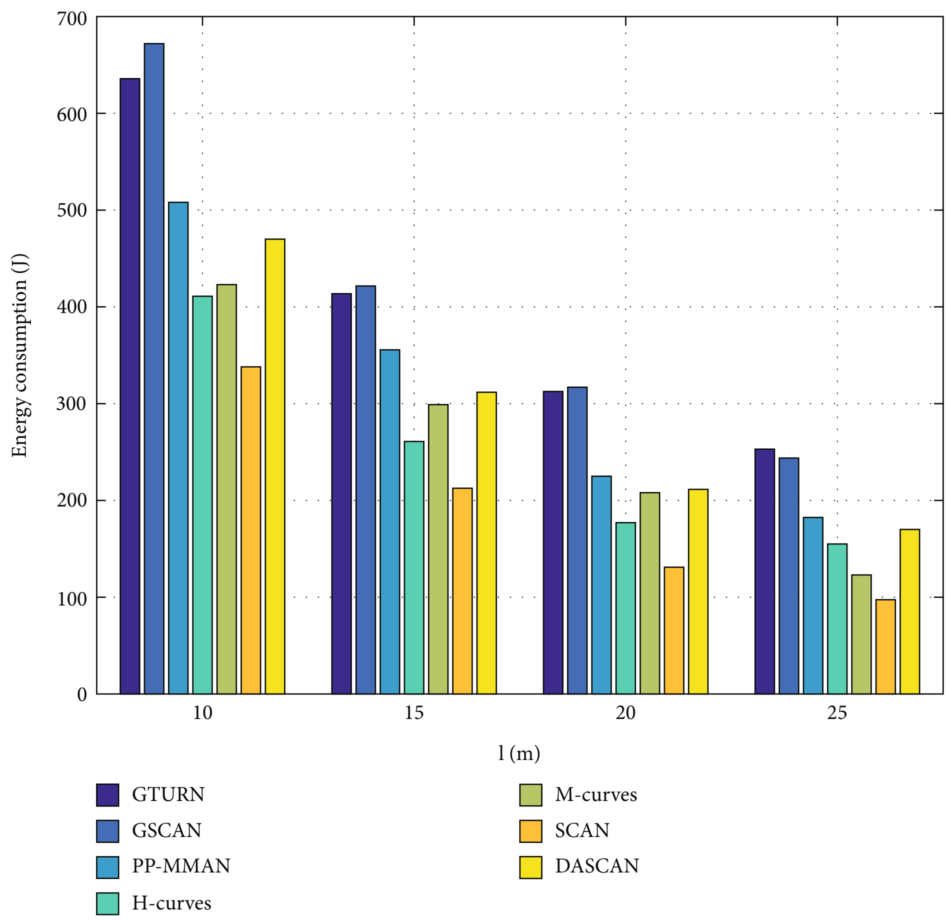


Figure 5: Energy consumption.

later to improve the ability of global search while preventing GWO from falling into a local optimum.

(2) *Random perturbation*. Furthermore, random perturbation is introduced to GWO to enhance the ability of global search. For each wolf of ω, a new individual is introduced as

$$\vec{Y}(t+1) = \vec{ub} - r_4\left(\vec{ub} - \vec{lb}\right), \tag{21}$$

where $\vec{ub}$ and $\vec{lb}$ are, respectively, the upper and lower bounds of the search space, and r_4 is a random number in [0,1], so the grey wolf is updated as

$$\vec{X}(t+1) = \begin{cases} \vec{Y}(t+1) & \text{if } f\left(\vec{Y}(t+1)\right) < f\left(\vec{X}(t+1)\right) \\ \vec{X}(t+1) & \text{otherwise} \end{cases}, \tag{22}$$

where $f(\vec{Y}(t+1))$ and $f(\vec{X}(t+1))$ are the fitness values of $\vec{Y}$ and $\vec{X}$, respectively.

(3) *Mirror grey wolf*. When a grey wolf surrounds the prey, a mirror wolf is produced at a position symmetrical to the prey, which is:

$$\vec{Z}(t+1) = \vec{X}(t) + 2\left(\vec{r_5} \bullet \vec{X_\alpha} - \vec{r_6} \bullet \vec{X}(t)\right), \tag{23}$$

where $\vec{Z}$ is the position of the mirror wolf, and $\vec{r_5}$ and $\vec{r_6}$ are random vectors in [0,1]. The grey wolf is updated as

$$\vec{X}(t+1) = \begin{cases} \vec{Z}(t+1) & \text{if } f\left(\vec{Z}(t+1)\right) < f\left(\vec{X}(t+1)\right) \\ \vec{X}(t+1) & \text{otherwise} \end{cases}. \tag{24}$$

Finally, the grey wolf is updated according to the value of P, defined by

$$P = w + r_7(2 - w), \tag{25}$$

where $0 < w < 2$ is a predefined constant, and r_7 is a random number in [0,1]. In each iteration, if $a > P > w$, the grey wolf has a 50% chance to update its position using Equation (22); If $P \geq a > w$, the grey wolf has a 50% chance to update its

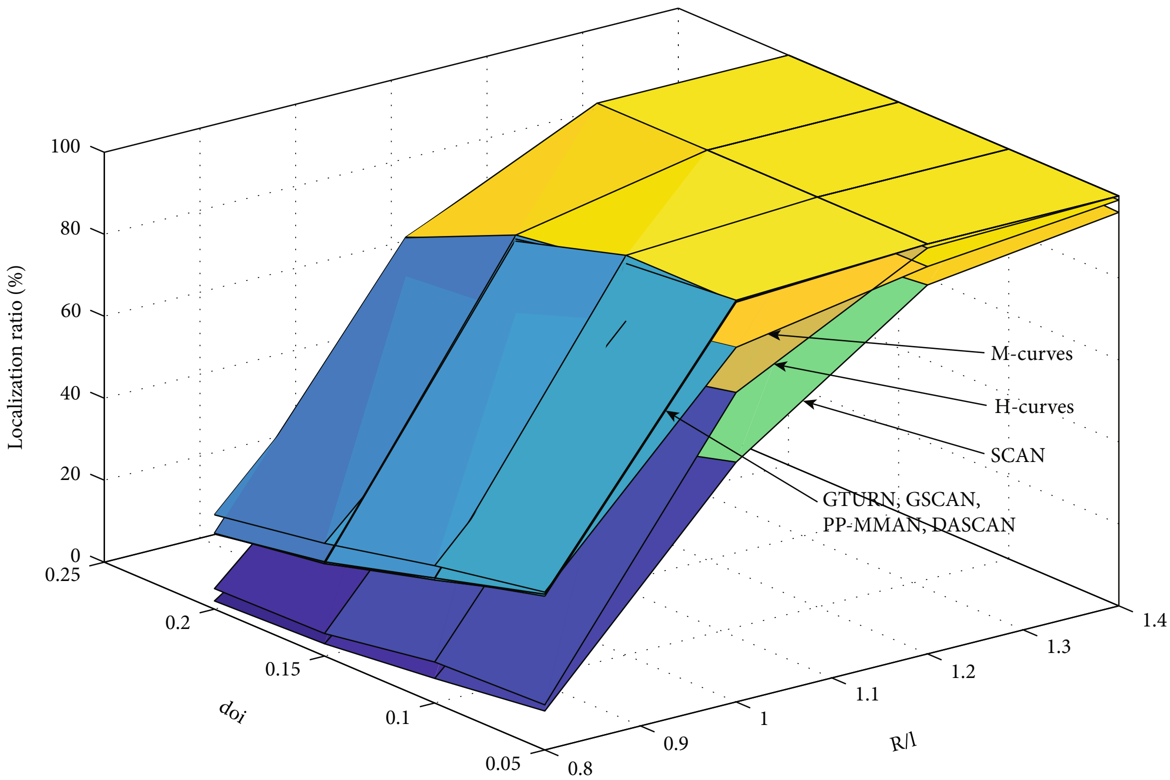


FIGURE 6: Localization rates.

TABLE 4: Localization errors of location estimation algorithms (Unit: m).

	Trilateration	PSO	WOA	GWO	MDS-GWO
GTURN	0.230	0.213	0.193	0.189	0.161
GSCAN	0.233	0.216	0.188	0.179	0.161
PP-MMAN	0.239	0.221	0.189	0.181	0.162
H-curves	0.266	0.250	0.213	0.202	0.186
M-curves	0.246	0.219	0.196	0.187	0.176
SCAN	0.370	0.350	0.329	0.306	0.273
DASCAN	0.232	0.215	0.187	0.180	0.160

position using Equation (24). In other cases, the grey wolf updates its position using Equation (19).

4.2.2. Location Estimation Using MDS-GWO. The virtual anchors of DASCAN form equilateral triangles, so the neighboring virtual anchors of an unknown node are likely to form equilateral triangles. Therefore, an unknown node can choose three neighboring virtual anchors forming an equilateral triangle to get an initial estimation. The location estimation of unknown node U_i consists of three steps:

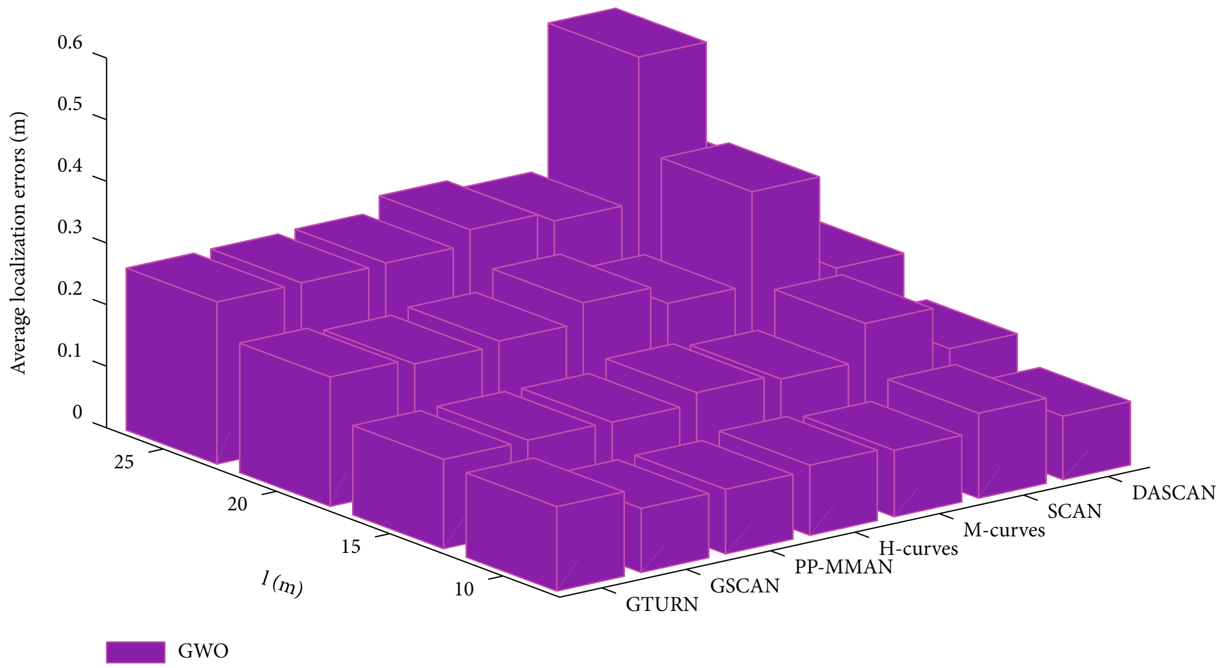
(1) Use an RSSI-weighted centroid algorithm to estimate the approximate location $(\widehat{ux}_i', \widehat{uy}_i')$

(2) Determine the search space SP_i of MDS-GWO for U_i according to $(\widehat{ux}_i', \widehat{uy}_i')$

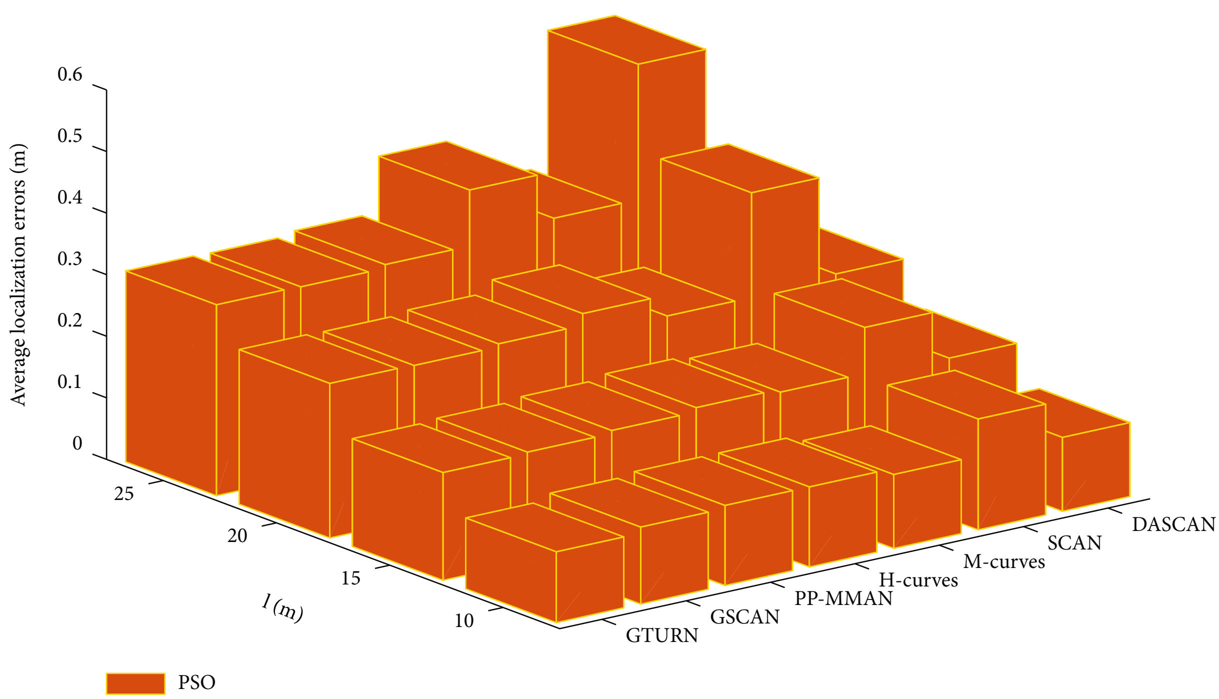
(3) Use MDS-GWO to estimate the coordinates $(\widehat{ux}_i, \widehat{uy}_i)$ of U_i

Each unknown node U_i maintains a virtual anchor list, L_i, which contains the locations and RSSIs of neighboring virtual anchors. The virtual anchors with the first- and second-largest RSSIs are, respectively, denoted by V_{i1}, and V_{i2}. If a virtual anchor in L_i can form an equilateral triangle with V_{i1} and V_{i2}, it is chosen as V_{i3}. Otherwise, the virtual anchor with the third-largest RSSI is chosen as V_{i3}. $(\widehat{ux}_i', \widehat{uy}_i')$ is calculated as

$$\left(\widehat{ux}_i', \widehat{uy}_i'\right) = \frac{\sum_{j=1}^{3}\omega_{ij}\left(vx_{ij}, vy_{ij}\right)}{\sum_{j=1}^{3}\omega_{ij}}, \tag{26}$$

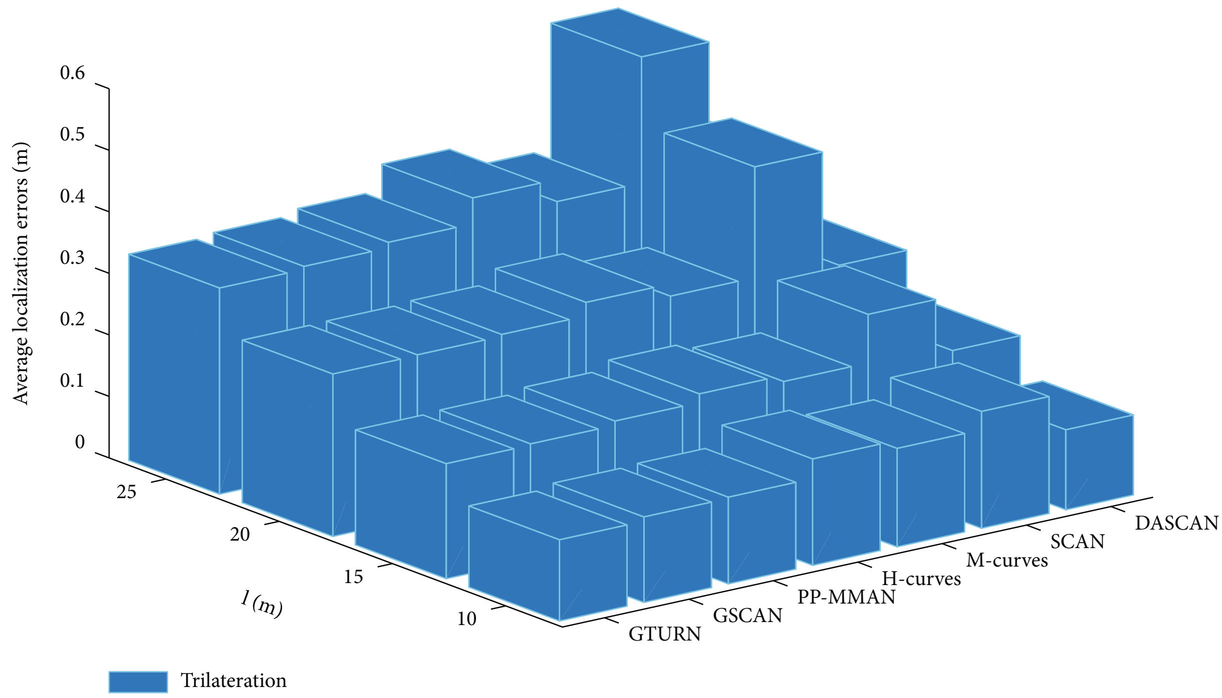


(a) GWO

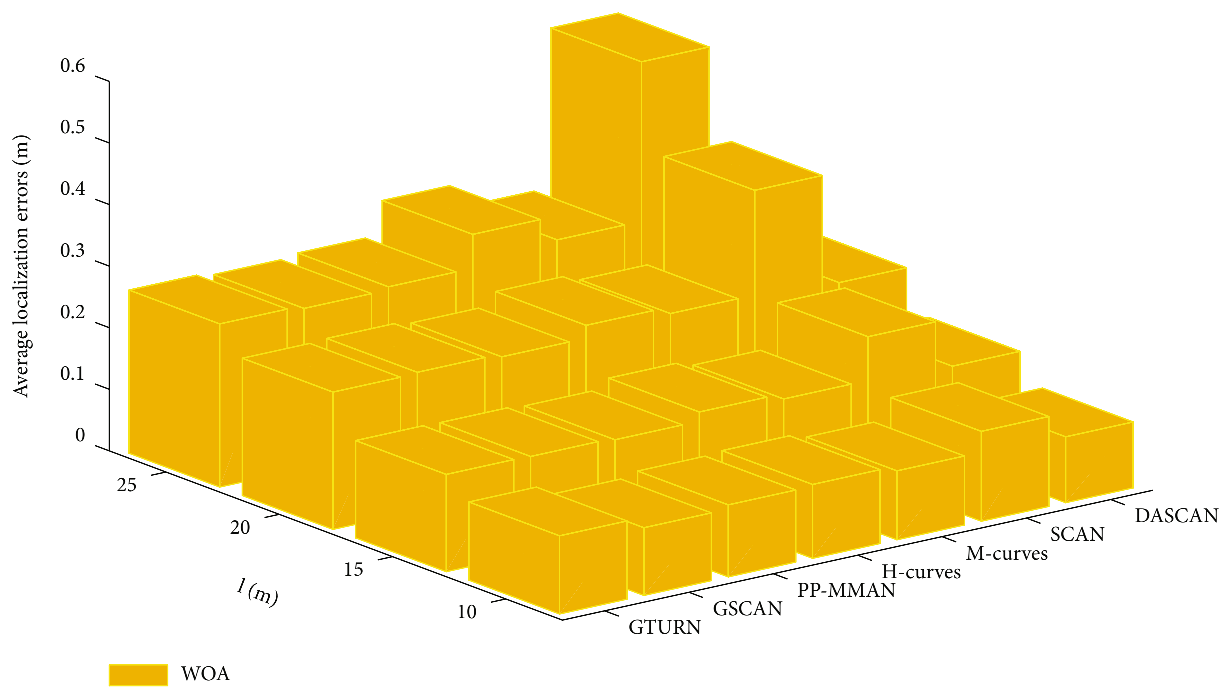


(b) PSO

Figure 7: Continued.

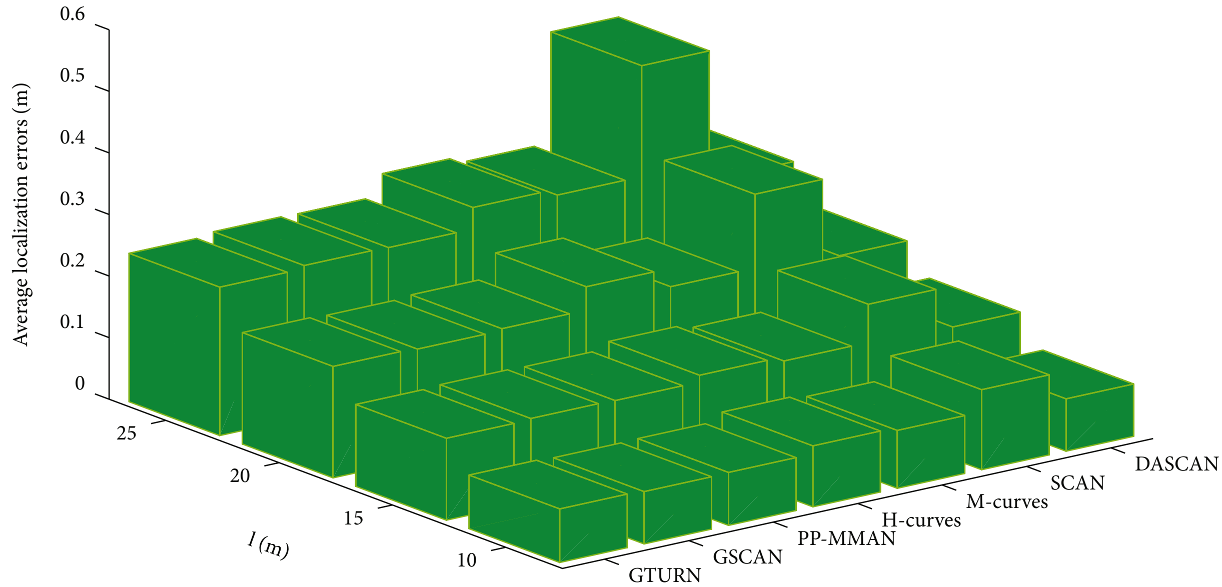


(c) Trilateration



(d) WOA

Figure 7: Continued.



(e) MDS-GWO

Figure 7: Localization errors of different location estimation algorithms.

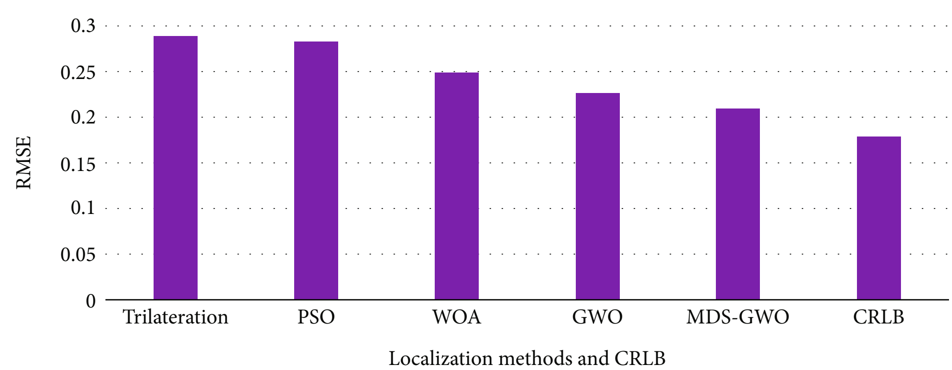


Figure 8: RMSE of localization error of five localization algorithms.

where $\omega_{ij} = \sqrt{\mathrm{RSSI}_{ij}}$. The search space is $[\widehat{ux}_i' - l, \widehat{ux}_i' + l] \times [\widehat{uy}_i' - l, \widehat{uy}_i' + l]$.

Algorithm 2 presents the details of MDS-GWO-based location estimation. Note that this algorithm is run by each unknown node simultaneously.

In this algorithm, lines 1–7 got three virtual anchors V_{i1}, V_{i2}, and V_{i3}, and lines 8 and 9 decided the search space based on these three virtual anchors. Lines 10–16 initialize the wolf population and necessary parameters of MDS-GWO, and lines 17–33 are the main process of location estimation by MDS-GWO. When it has not reached MaxIter, lines 18–28 update each grey wolf according to different conditions, and lines 29–32 update the parameters necessary to prepare for the next iteration. Finally, line 34 returns the position of wolf α as the estimated location of unknown node U_i.

5. Results and Discussion

This section reports the performance of the proposed algorithm compared with the other algorithms. It also presents the impact of parameters on localization performance in terms of energy consumption, localization rate, and localization error. The simulation parameters are shown in Table 3. The other parameters are the same as in the corresponding references.

5.1. Comparisons of Different Paths. This section compares the different paths of mobile anchors of GTURN, GSCAN, PP-MMAN, H-Curves, M-Curves, SCAN, and DASCAN.

(1) *Energy Consumption of Mobile Anchors.* The energy consumption of a mobile anchor depends on the path length and the number of virtual anchors.

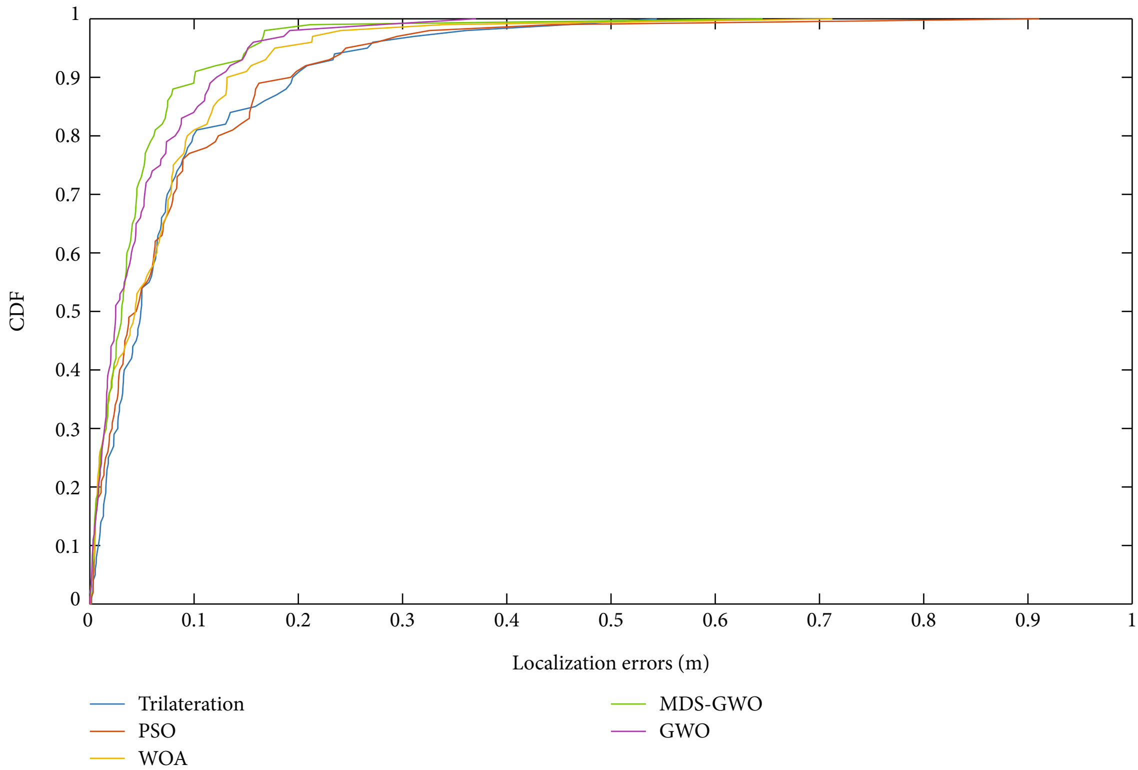


Figure 9: CDF of localization error of five localization algorithms.

Figure 3 presents the lengths of all paths, which become shorter as l increases. On average, GTURN is the longest path, followed by GSCAN, PP-MMAN, DASCAN, H-Curves, M-Curves, and SCAN. GTURN, GSCAN, and PP-MMAN use three mobile anchors, which are longer than the other paths. DASCAN is shorter than the other multianchor paths and longer than one-anchor paths. On average, DASCAN is 53%, 36.5%, and 14% shorter than GTURN, GSCAN, and PP-MMAN, respectively; it is 14%, 15.5%, and 34% longer than M-Curves, H-Curves, and SCAN, respectively.

DASCAN uses two mobile anchors with a carefully designed path, while GSCAN, GTURN, and PP-MMAN use three mobile anchors and boundary compensations, so DASCAN is shorter than GSCAN, GTURN, and PP-MMAN. By contrast, M-Curves, H-Curves, and SCAN use a single mobile anchor to traverse the ROI without repeated scans, so they are shorter than DASCAN.

Figure 4 shows the number of virtual anchors. All paths generate fewer virtual anchors as l increases. Generally, GSCAN generates the most virtual anchors, followed by GTURN, PP-MMAN, DASCAN, M-Curves, H-Curves, and SCAN. The three mobile anchors of GSCAN repeatedly broadcast beacons at the same locations, and GSCAN provides the second-longest path, so it has the most virtual anchors. However, SCAN traverses the ROI along straight lines, and its path is the shortest, so it has the fewest virtual anchors. The proposed DASCAN has fewer virtual anchors than the other multianchor paths and more virtual anchors than one-anchor paths. The number of virtual anchors of DASCAN is, respectively, 11% and 53% fewer than GTURN and GSCAN, and it is, respectively, 12.3% and 32% more than H-Curves and SCAN. The number of virtual anchors of DASCAN is slightly less than PP-MMAN and slightly more than M-Curves.

The boundary compensation methods of GSCAN, GTURN, and PP-MMAN require these paths to generate more virtual anchors, so GSCAN has redundant virtual anchors. By contrast, DASCAN has no duplicate virtual anchors or boundary compensation, so it has fewer virtual anchors than GSCAN, GTURN, and PP-MMAN. SCAN traverses ROI with straight lines and generates virtual anchors periodically, while H-Curve and M-Curve generate virtual anchors at each turn of the path. DASCAN makes slightly more turns than M-Curve, so it has more virtual anchors than SCAN, H-Curve, and M-Curve.

Figure 5 shows the energy consumptions. GSCAN consumes the most energy, followed by GTURN, PP-MMAN, DASCAN, M-Curves, H-Curves, and SCAN. Since GSCAN and GTURN are the longest paths and have the most virtual anchors, they consume the most energy. DASCAN consumes less energy than the other multianchor paths and consumes more energy than one-anchor paths. On average, the energy consumption of DASCAN is, respectively, 42.2%, 38.8%, and 9.3% less than GSCAN, GTURN, and PP-MMAN, and 9%, 14%, and 33% more than M-Curves, H-Curves, and SCAN.

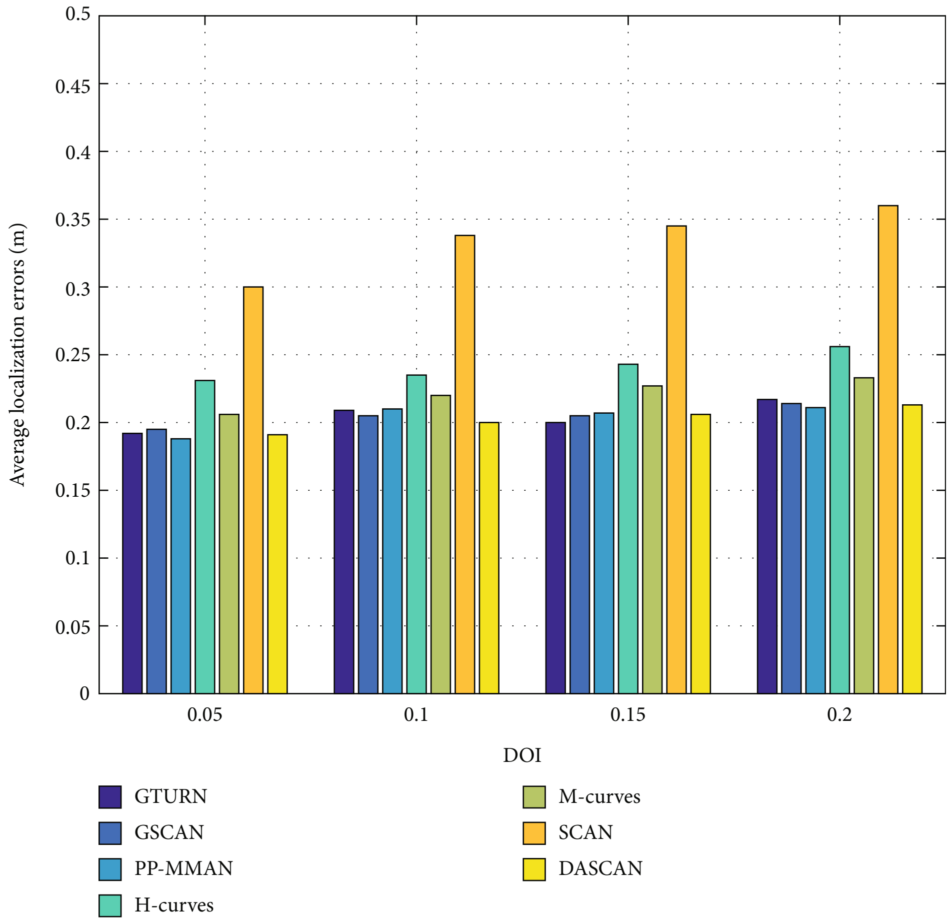


Figure 10: Localization errors under different DOI.

Although DASCAN consumes less energy than GTURN, GSCAN, and PP-MMAN, their localization rates and error levels are almost identical. The localization rate and accuracy of DASCAN are higher than H-Curves, M-Curves, and SCAN, although it consumes more energy.

(2) *Localization Rates*. In this section, $l = 15$ m. As Figure 6 shows, the localization rates of all paths decrease as DOI increases because the larger DOI causes the unknown nodes to receive fewer virtual anchors. The localization rates of all paths increase as R increases because a larger R causes the unknown node to receive more virtual anchors.

GSCAN, GTURN, PP-MMAN, and DASCAN use multiple anchors, and each anchor has a radio range, so they are less affected by DOI than other paths using a single mobile anchor. The average localization rate differences of DASCAN, GSCAN, GTURN, and PP-MMAN are less than 2%, and their average localization rates are higher than H-Curves, M-Curves, and SCAN. When $R = l$, the average difference in GSCAN, GTURN, PP-MMAN, and DASCAN is only 0.56%, but the average localization rate of DASCAN is, respectively, 13%, 38.3%, and 43.4% higher than M-Curves, H-Curves, and SCAN.

5.2. Comparisons of Location Estimation Algorithms. This section compares the location estimation algorithms in terms of the localization error. These algorithms, with different paths, included trilateration, PSO, WOA, GWO, and MDS-GWO. In this section, $R = 1.2l$ and $DOI = 0.05$ are used. The average localization error over all l is shown in Table 4. Figure 7 shows the results for different l values.

The localization errors of trilateration and PSO were the two largest. Those of GWO and WOA were the same. With each path, MDS-GWO achieved the highest localization accuracy. In all cases, trilateration with SCAN had the lowest localization accuracy. The localization errors of MDS-GWO with GTURN, GSCAN, PP-MMAN, and DASCAN were almost identical and smaller than MDS-GWO with the other paths. Considering all paths, the average localization errors of trilateration, PSO, WOA, GWO, and MDS-GWO were, respectively, 0.259, 0.240, 0.213, 0.202, and 0.183 m. The localization error of MDS-GWO was, respectively, 42%, 31.7%, 16.9%, and 11.4% less than trilateration, PSO, WOA, and GWO. Considering MDS-GWO with different paths, the localization accuracy of DASCAN was almost the same as GTURN, GSCAN, PP-MMAN, and the localization error of DASCAN was, respectively, 70.6%, 16.3%, and 10% less than SCAN, H-Curves, and M-Curves. The virtual anchors of DASCAN, GSCAN, GTURN, and PP-MMAN

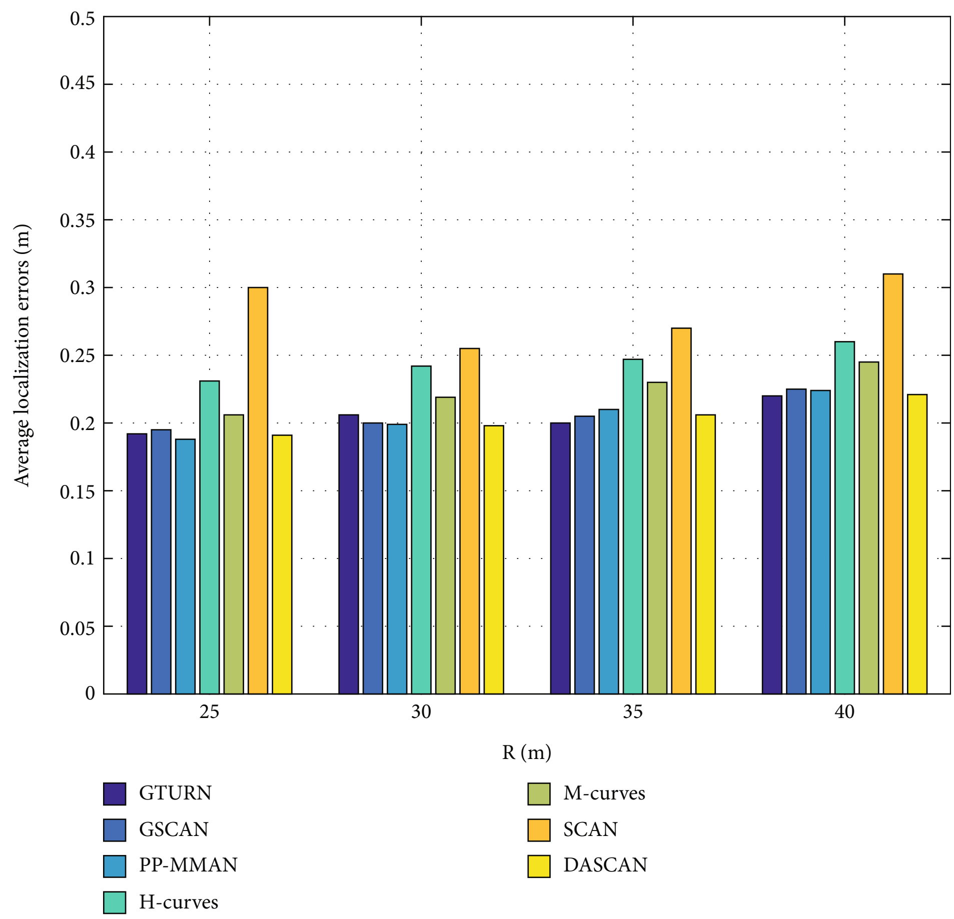


Figure 11: Localization errors under different R.

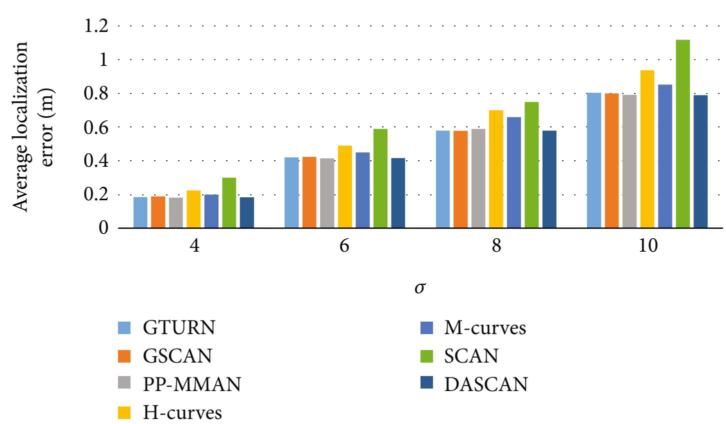


Figure 12: Localization errors under different σ.

formed equilateral triangles. The localization rates of these paths were almost the same as mentioned above, so their localization errors were almost the same. SCAN, H-Curves, and M-Curves use only one mobile anchor, which is heavily affected by DOI, so their localization errors were higher than the other paths.

Figure 8 shows the RMSE of localization algorithms when $l = 20$ m, $R = 1.2$ l, $DOI = 0.05$, and the mobile anchors use DASCAN as the moving path. MDS-GWO had the minimal RMSE, followed by GWO, WOA, PSO, and trilateration. On average, the RMSE of MDS-GWO was, respectively, 8.2%, 19%, 35.1%, and 38% less than GWO,

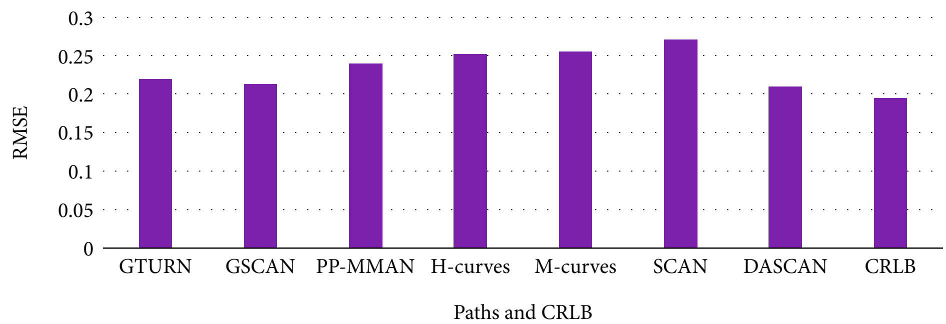


Figure 13: RMSE of localization error of seven paths.

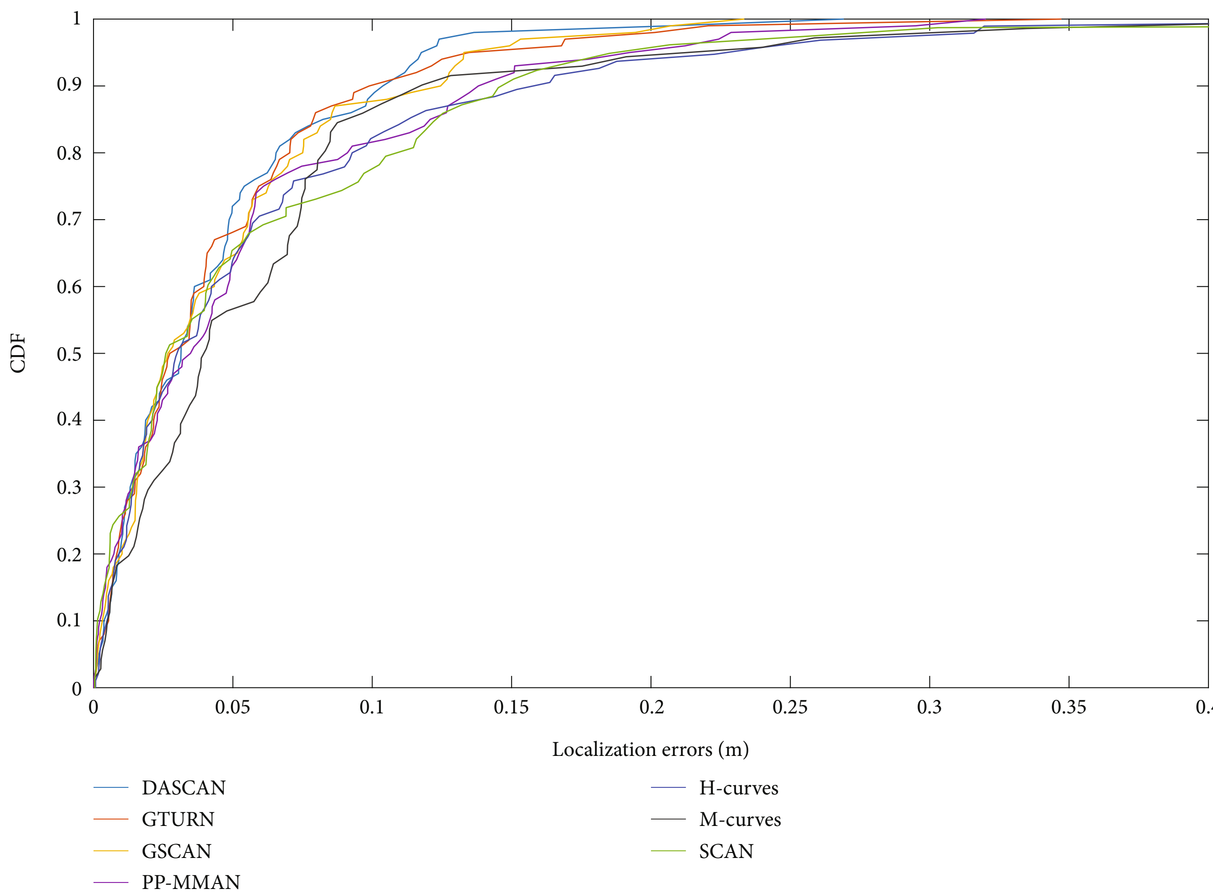


Figure 14: CDF of localization error of seven paths.

WOA, PSO, and trilateration. Figure 9 shows the CDF of localization errors. MDS-GWO localized all sensor nodes with an error of less than 0.3 m, and more than 90% of the unknown nodes were localized with an error of less than 0.1 m. Comparatively, GWO and WOA, respectively, localize about 85% and 80% of unknown nodes with localization errors less than 0.1 m. Furthermore, the RMSE of MDS-GWO was only 0.03 larger than $RMSE_{CRLB}$. MDS-GWO made the most of GWO's advantages and applied multiple strategies to jump out of the local optimum, so we concluded that MDS-GWO was superior to the other localization methods.

5.3. Impacts of Different Parameters on Localization Errors. This section analyzes the impact of DOI, R, and σ on localization errors of MDS-GWO over different paths.

(1) *Impact of* DOI. In this part, $l = 20$ m and $R = 25$ m. Figure 10 shows that SCAN had the most significant localization error under all DOI, followed by H-

Curves and M-Curves. With the increase in DOI, the localization errors of all paths increased. For every 0.05 increase in DOI, the localization error of SCAN increased by 0.02 m on average, which is the largest among all paths. DASCAN was less affected by DOI than H-Curves, M-Curves, and SCAN, and almost the same as GTURN, GSCAN, and PP-MMAN. Under all DOI, the localization error of DASCAN was the same as GTURN, GSCAN, and PP-MMAN and less than H-Curves, M-Curves, and SCAN. The localization error of DASCAN was, respectively, 0.03, 0.02, and 0.07 m less than H-Curves, M-Curves, and SCAN.

(2) *Impact of R*. In this part, $l = 20$ m, $R = 25, 30, 35,$ and 0 m, and DOI = 0.05. As Figure 11 shows localization errors of all paths increased with R. The localization error of SCAN was the largest in all cases because unknowns may not receive sufficient beacons to be localized, and many virtual anchors of SCAN were collinear. The localization errors of GTURN, GSCAN, PP-MMAN, and DASCAN were almost the same. For every 5 m increase in R, the localization errors of M-Curves and SCAN increased by 0.013 m and 0.15 m, respectively, while those of GTURN, GSCAN, PP-MMAN, DASCAN, and H-Curves grew by less than 0.01 m. Of all paths, DASCAN was the least affected by R. Under all R values, the localization error of DASCAN was slightly smaller than GTURN, GSCAN, and PP-MMAN and was significantly less than H-Curves, M-Curves, and SCAN. On average, the localization error of DASCAN was, respectively, 0.05, 0.03, and 0.08 m less than H-Curves, M-Curves, and SCAN.

(3) *Impact of σ*. In this part, $l = 20$ m, $R = 25$ m, DOI = 0.05, and $\sigma = 4, 6, 8,$ and 10. Figure 12 shows that the localization error of SCAN was the largest, followed by H-Curves and M-Curves. For every two increases of σ, the localization errors of H-Curves, M-Curves, and SCAN, respectively, increased 0.23, 0.218, and 0.27 m on average, and the localization errors of GTURN, GSCAN, PP-MMAN, and DASCAN increased by 0.2 m. DASCAN was less affected by σ than H-Curves, M-Curves, and SCAN, and it was almost the same as GTURN, GSCAN, and PP-MMAN. On average, the localization error of DASCAN was, respectively, 0.1, 0.05, and 0.2 m less than H-Curves, M-Curves, and SCAN.

(4) *Analysis of RMSE and CDF*. In this part, $l = 20$ m, $R = 25$ m, and DOI = 0.05. Figure 13 shows that the RMSE of seven paths with DASCAN had the least RMSE, followed by GSCAN, GTURN, PP-MMAN, M-Curves, H-Curves, and SCAN. The RMSE of DASCAN was, respectively, 20.3%, 21.7%, and 29% less than H-Curves, M-Curves, and SCAN, and it was only 0.015 higher than $\mathrm{RMSE}_{\mathrm{CRLB}}$. As Figure 14 shows, the CDF of DASCAN, GTURN, and GSCAN increased faster than the other paths, which means they could localize all unknown nodes with higher accuracy than the other paths. After a localization error of 0.11 m, the CDF of DASCAN was always larger than the other paths. For example, DASCAN localized more than 89% of unknown nodes with a localization error of less than 0.15 m, which was the largest among all paths.

6. Conclusions

This paper proposes a localization algorithm using two mobile anchors to balance the localization accuracy and energy consumption. The study developed a specified path, DASCAN, which uses two mobile anchors to traverse different rows after dividing the ROI into regular grids. The neighboring virtual anchors generated by the two mobile anchors form equilateral triangles to provide noncollinear beacons for each unknown node to be localized. The paper also proposes an improved GWO algorithm (MDS-GWO) to estimate the positions of unknown nodes. MDS-GWO introduces multiple disturbance strategies into GWO to improve its ability to jump out of local optima.

Data Availability

Data sets are not used in this article.

Conflicts of Interest

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

Acknowledgments

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

References

[1] G. Han, J. Jiang, C. Zhang, T. Q. Duong, M. Guizani, and G. K. Karagiannidis, "A survey on mobile anchor node assisted localization in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2220–2243, 2016.

[2] X. Song, Y. Zhao, and L. Wang, "Gauss–Markov-based mobile anchor localization (GM-MAL) algorithm based on local linear embedding optimization in internet of sensor networks," *Cognitive Systems Research*, vol. 52, pp. 138–143, 2018.

[3] D. Koutsonikolas, S. M. Das, and Y. Charlie Hu, "Path planning of mobile landmarks for localization in wireless sensor

networks," *Computer Communications*, vol. 30, no. 13, pp. 2577–2592, 2007.

[4] A. Alomari, F. Comeau, W. Phillips, and N. Aslam, "New path planning model for mobile anchor-assisted localization in wireless sensor networks," *Wireless Networks*, vol. 24, no. 7, pp. 2589–2607, 2018.

[5] G. Han, X. Yang, L. Liu, W. Zhang, and M. Guizani, "A disaster management-oriented path planning for mobile anchor node-based localization in wireless sensor networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 1, pp. 115–125, 2020.

[6] P.-H. Tsai, G.-R. Shih, W.-D. Cheng, and R.-G. Tsai, "Sigma -scan: a mobile beacon-assisted localization path-planning algorithm for wireless sensor networks," *IEEE Sensors Journal*, vol. 19, no. 23, pp. 11492–11502, 2019.

[7] K. Kannadasan, D. R. Edla, M. C. Kongara, and V. Kuppili, "M-curves path planning model for mobile anchor node and localization of sensor nodes using dolphin swarm algorithm," *Wireless Networks*, vol. 26, no. 4, pp. 2769–2783, 2020.

[8] T. Ecenaz Erdemir and E. Tuncer, "Path planning for mobile-anchor based wireless sensor network localization: static and dynamic schemes," *Ad Hoc Networks*, vol. 77, pp. 1–10, 2018.

[9] G. Han, J. Jiang, J. Chao, and X. Yang, "Path planning for a group of mobile anchor nodes based on regular triangles in wireless sensor networks," *Neurocomputing*, vol. 270, pp. 198–208, 2017.

[10] S. Sun, J. Zhao, X. Tian, and J. Zhang, "Path planning for multiple mobile anchor nodes assisted localization in wireless sensor networks," *Measurement*, vol. 141, pp. 124–136, 2019.

[11] P. Thilagavathi and J. Manickam, "ERTC: an enhanced RSSI based tree climbing mechanism for well-planned path localization in WSN using the virtual force of mobile anchor node," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 6, pp. 6665–6676, 2021.

[12] P. Singha, A. Khoslaa, A. Kumarb, and M. Khosla, "Optimized localization of target nodes using single mobile anchor node in wireless sensor network," *AEU-International Journal of Electronics and Communications*, vol. 91, pp. 55–65, 2018.

[13] L. Song, L. Zhao, and J. Ye, "DV-hop node location algorithm based on GSO in wireless sensor networks," *Journal of Sensors*, vol. 2019, Article ID 2986954, 9 pages, 2019.

[14] Y. Huang and L. Zhang, "Weighted DV-Hop localization algorithm for wireless sensor network based on differential evolution algorithm," in *Proceedings of IEEE 2nd International Conference on Information and Computer Technologies*, pp. 14–18, Kahului, USA, 2019.

[15] S. Phoemphon, C. So-In, and N. Leelathaku, "Improved distance estimation with node selection localization and particle swarm optimization for obstacle-aware wireless sensor networks," *Expert Systems with Applications*, vol. 175, article 114773, 2021.

[16] F. Lang, S. Jun, Z. W. Ye, X. X. Shi, and F. Chen, "A wireless sensor network location algorithm based on whale algorithm," in *Proceedings of 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pp. 106–110, Metz, France, 2019.

[17] Z. Guo, Y. Guo, F. Hong et al., "Perpendicular intersection: locating wireless sensors with mobile beacon," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 7, pp. 3501–3509, 2010.

[18] O. Chia-Ho and W. He, "Path planning algorithm for mobile anchor-based localization in wireless sensor networks," *IEEE Sensors Journal*, vol. 13, no. 2, pp. 466–475, 2013.

[19] H. Shi, X. Li, Y. Shang, and D. Ma, "Error analysis of quantised RSSI based sensor network localisation," *International Journal of Wireless and Mobile Computing*, vol. 4, no. 1, pp. 31–40, 2010.

[20] Z. Wei, H. Zhao, B. Han, C. Sun, and M. Li, "Grey wolf optimization algorithm with self-adaptive searching strategy," *Computer Science*, vol. 38, no. 5, pp. 1434–1437, 2021.

[21] C. Chen, R. Chellali, and Y. Yin, "Improved grey wolf optimizer algorithm using dynamic weighting and probabilistic disturbance strategy," *Journal of Computer Applications*, vol. 37, no. 12, pp. 3493–3497, 2017.