Hindawi
Journal of Sensors
Volume 2023, Article ID 2261730, 18 pages
https://doi.org/10.1155/2023/2261730

Research Article

Jammer Location-Aware Method in Wireless Sensor Networks Based on Fibonacci Branch Search

Fang Yang, Nina Shu, Chenxi Hu, Jun Huang, and Zhao Niu

National University of Defense Technology, College of Electronic Engineering, Hefei, Anhui 230037, China

Correspondence should be addressed to Nina Shu; shunina@nudt.edu.cn

Received 25 July 2021; Revised 21 April 2022; Accepted 11 May 2022; Published 10 May 2023

Academic Editor: Xue-bo Jin

Due to the sharing and open-access characteristics of the wireless medium, wireless sensor networks (WSNs) can be easily attacked by jammers. To mitigate the effects of a jamming attack, one reliable solution is to locate and remove the jammer from the deployed area within the WSN. To realize the jammer's localization in the WSN, many range-free methods have been proposed. However, most of these methods are sensitive to the distribution of nodes and the parameters of the jammer. For this reason, a jammer location-aware method based on Fibonacci branch search (FBS) is proposed in this article. First, the interference region is estimated by using the interference region mapping service of sensors in wireless sensor networks. Then, the search point is selected in the jamming area and the fitness function is designed according to the average distance from the search point to the boundary sensor. According to the basic branch structure and interactive search rules, the global optimal solution is obtained in the jamming area. Finally, the position of the search point with the best fitness value is used as the estimation of the jammer position. Compared with the existing typical range-free methods, rich simulation experiments demonstrate that the FBS algorithm is superior in the location-aware method for jammers with a higher precision and a lower sensitivity to the distribution of nodes and the parameters of the jammer, respectively.

1. Introduction

Wireless sensor networks (WSNs) are seriously threatened by radio interference attacks due to the sharing and open-access characteristics of wireless mediums [1–3]. The radio interference attacks, which are also known as jamming attacks, can seriously disrupt normal communication between legitimate sensors. By occupying the wireless communication channel or disrupting the workflow of network protocols, WSN jamming attacks can be easily initiated [4]. To reduce the impact of jamming attacks on network performance and ensure the security of WSNs, various prevention jamming strategies [5] have been proposed, such as covert timing channels [6], channel hopping [7], protocol optimization [8], channel-aware decision fusion [9], and spatial retreat [10]. In addition to these strategies, jammer location awareness is an effective method that helps us remove the jammer based on the jammer's location [11].

To date, most jammer location-aware problems have been extensively investigated and several location perception algorithms have been designed. In general, existing location-aware strategies can be classified as range-free methods and range-based methods [12]. Range-based methods usually estimate the distance information by measuring some physical attributes of jammer signals [13]. The relevant physical attributes mainly include received signal strength indicators (RSSIs) [14], time difference of arrival (TDoA) [15], time of arrival (ToA) [16], and angle of arrival (AoA) [17]. The location of the jammer is calculated by the Euclidean distance or angle size between legitimate nodes and the jammer. Different calculation algorithms will greatly affect the positioning accuracy. Typical algorithms include triangulation [18], trilateration [19], and algorithms based on multidimensional scaling (MDS), such as MDS [20], MDS-MAP [21], MDS-MAP(P) [22], and SMDS [23]. However, due to the small size of sensors in WSNs, battery life is limited and the life of the sensor is directly affected by the life of the battery [24], which further affects the life of the entire network to maximize the life cycle of the sensor network. Therefore, the seniors in WSNs are usually not allowed components

that can realize the distance between the sensors and the jammer. At the same time, sensors in the jammed area have difficulty communicating with each other because of jammer interference. Even though every sensor is equipped with ranging components, the jammed sensor cannot obtain the location information of neighboring sensors to locate the jammer's position. For these reasons, range-free methods are more suitable for WSNs to achieve the localization of jammers in the network [25].

Range-free location-aware algorithms make use of the geometric relationship of the jammed area to realize the localization of the jammer. At the same time, the problem of jamming detection was briefly researched by Wood et al. [26] and further investigated by Xu et al. [27]. Wood et al. proposed a scheme that could map a jammed region, and Xu et al. presented several measurements to detect jamming attacks in WSNs. To calculate the jammer localization, Bulusu et al. proposed centroid localization (CL) [28], which estimates the coordinates of the jammer by calculating the average relative coordinates of all jammed nodes. CL is easy to realize, but the localization error is large. To improve the location-aware accuracy, Blumenthal et al. proposed weighted centroid localization (WCL) [29], which is based on the assumption that the influence degree of different nodes on the localization of the jammer is different. The closer the node is to the jammer, the greater the weight value of the node is. WCL needs RSSI to realize the computation of the weight, which is difficult in some scenarios. To further improve the location-aware accuracy, Shoari and Seyedi proposed an algorithm based on the minimum enclosing rectangle center [12]. In the algorithm, the position of the jammer is calculated as the center of the smallest enclosing rectangle covering all of the jamming nodes. Liu et al. proposed virtual force iterative localization (VFIL) to locate the jammer [30]. The concepts of pull force, push force, and joint force are defined in the VFIL. The jammed node produces a pulling force to constantly pull the jammer to itself, and the boundary sensors push the jammer away from itself through a push force. The final location awareness of the jammer is realized through the joint force produced by the jammed nodes and the boundary nodes in the jammed area. To reduce the sensitivity of the distribution of nodes when locating the jammer, Wang et al. proposed a heuristic optimization evolutionary algorithm named the gravitational search algorithm (GSA) [1]. This method merges mass interactions and Newton's law of universal gravitation. After iterations, the coordinates of the most massive particle are adopted as the coordinates of the jammer. In addition, for multijammer scenarios, Cheng et al. designed two methods to calculate the localization of jammers based on M-clusters and X-rays [31]. Wang et al. utilized the k-means clustering algorithm to estimate the location of the jammer according to the location information of neighboring nodes [32].

The distributed sensors and the parameters of the jammer can easily affect the performance of range-free methods. In this paper, to decrease the sensitivity of range-free methods and improve the location-aware accuracy, a robust location-aware algorithm based on the Fibonacci branch search (FBS) is designed. Meanwhile, the reachability and convergence of FBS are proven mathematically, which further verifies the validity of the theory for the FBS strategy. The location-aware algorithm based on FBS uses the powerful global searchability and the high convergence speed of the technology. It improves the location-aware performance by preventing the loss of the best trajectory. Aiming at the abovementioned problems in target node location, the main contribution of this paper includes three aspects.

(1) In this paper, a robust location sensing algorithm based on Fibonacci branch search (FBS) is designed to reduce the sensitivity of distance-independent methods and improve the accuracy of location sensing. At the same time, the reachability and convergence of FBS are proved mathematically, which further verifies the effectiveness of the FBS strategy theory

(2) Based on the FBS algorithm, interactive global search and local optimization rules are used alternately to realize global optimization. Finally, the coordinates of the search point with the highest fitness value are taken as the coordinates of the jammer. Compared with many existing location aware algorithms, the proposed method has higher performance in complex scenes with different parameter settings

(3) Experiments show that the proposed method can locate the target node when it is unable to range the target node, and in terms of optimization ability, due to other similar algorithms, at the same time, even if the deployment area of the wireless sensor network has the characteristics of low density and low communication distance, the positioning error based on FBS is still less than other algorithms

The rest of this paper is described as follows. Section 2 describes the network model and the jamming model. Section 3 illustrates the main principle of the FBS algorithm. The location-aware method of jammers based on FBS is presented in Section 4. Section 5 illustrates and discusses the simulation results. Section 6 presents our conclusions.

2. System Models

The scenario with a WSN and a jammer is shown in Figure 1, where the jammer is surrounded by sensors. In this section, the network model and the jamming model are outlined.

2.1. Network Model. Assume that N_S homogeneous sensors are deployed in the area to form a WSN. $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_{N_S}]^T \in N_S \times 2$ are the coordinates of all of the sensors, and $\mathbf{s}_i = [x_i^s, y_i^s], i = 1, 2, \cdots, N_S$. The sensors can be aware of their locations through GPS or other location-aware algorithms, e.g., when the distances between the sensors are obtained, algorithms based on MDS can be used to realize location awareness. Once deployed, this article considers the location of the sensor to remain unchanged. Every sensor in the network is equipped with an omnidirectional

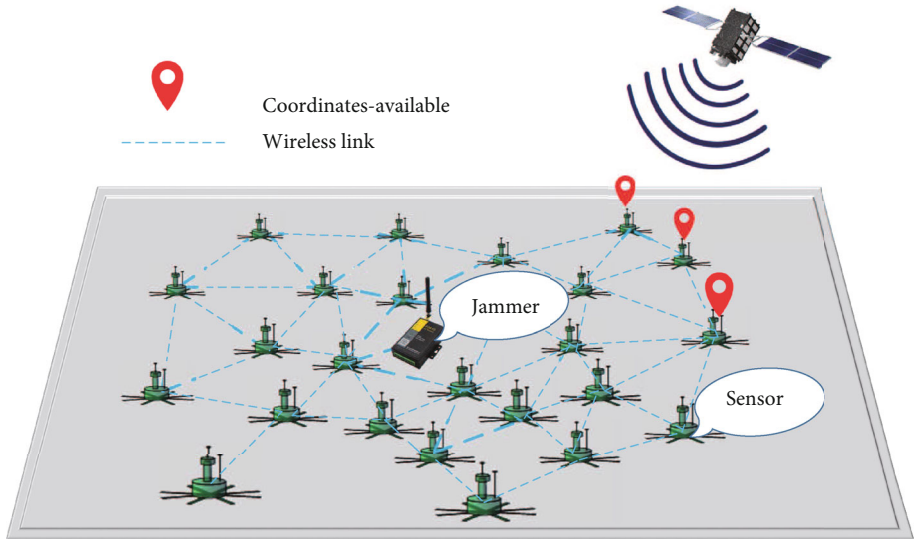


Figure 1: Scenario diagram of a WSN and a jammer.

antenna and can communicate with other sensors within a communication range R_S. There are no isolated subnets or isolated sensors in the network. For sensors that are not in the communication range, sensors can communicate with each other in a multihop manner. According to the abovementioned assumptions, the network model of the WSN in Figure 1 is shown in Figure 2.

2.2. Jamming Model. In this model, we assume that the jamming power remains stable and the jammer location stays static. This jammer continuously transmits radio signals, which can be implemented using a waveform generator that continuously transmits radio signals or a normal wireless device that keeps sending random bits to the channel without following any MAC layer protocol. Due to the large coverage of the omnidirectional antenna, the jammer is equipped with an omnidirectional antenna with the same direction effect. Under the effect of the jammer, the sensors in the WSN are divided into three types: jammed sensors, boundary sensors, and unaffected sensors. Jammed sensors are located in the jammed area and cannot communicate with any neighboring sensors. Boundary sensors are usually located at the edge of the jammed area. Although they struggle with jamming attacks, the sensors can still communicate with neighboring sensors. Unaffected sensors outside the jammed area can receive information from neighboring sensors even if the jammer appears. $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_{N_B}]^T \in N_B \times 2$ contains the coordinates of all of the boundary sensors, and $\mathbf{b}_i = [x_i^B, y_i^B]$, $i = 1, 2, \cdots, N_B$, where N_B is the number of boundary sensors in the network.

According to the abovementioned assumptions, the jamming model of the WSN is shown in Figure 3.

When a node detects itself as jammed, the node broadcasts notification messages to its neighboring nodes, as shown in Figure 4(a). Mapping is conducted by the neighboring sensors of jammed sensors who receive the interference message. Each receiver becomes a mapping member and adds nearby jammed sensors to form a local group. As shown in Figure 4(b), adjacent sensors contain mapping messages for the local exchange of group information. Adjacent groups are condensed together to form a mapped area, as shown in Figure 4(c), which can be used as an estimation of the jammed area [26]. The sensors that constitute the mapped area are all boundary nodes. The notations that will be used throughout the paper are summarized in Table 1.

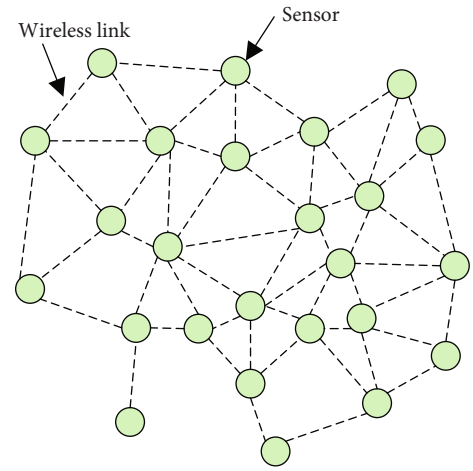


Figure 2: The network model of the WSN.

3. Fibonacci Branch Search Algorithm

The Fibonacci optimization method has proven the effectiveness and convergence of solving a series of nonlinear benchmark functions in one-dimensional space [33]. However, the method is rarely used in the properties of multidimensional space search optimization problems. In addition to the structure itself, there are few variants implemented in localized applications reported in the public literature.

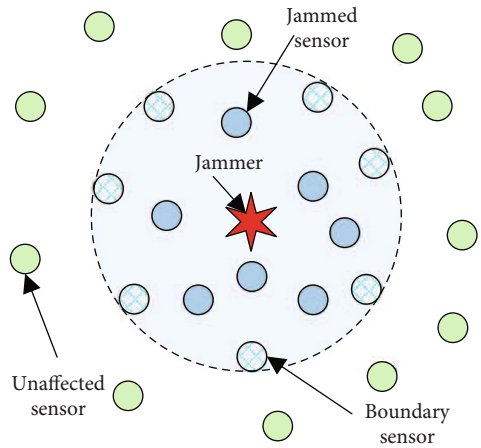


Figure 3: The jamming model of the WSN.

In this section, we will briefly introduce the traditional Fibonacci sequence method and explain the principle and in-depth understanding of the FBS-based method.

3.1. Basic Principle of the Fibonacci Method. The Fibonacci sequence, also known as the golden section sequence, was first proposed by Fibonacci [34], and its general formula is as follows [35].

$$\begin{aligned} &F(1) = F(2) = 1, \\ &F(j) = F(j-1) + F(j-2), \quad j \geq 3, \end{aligned} \tag{1}$$

where F_j indicates the jth Fibonacci sequence, which is a general term.

This paper studies the optimization method of the Fibonacci sequence to solve the minimization problem of a unimodal continuous function in the interval. The Fibonacci sequence optimization method compresses the search interval proportionally according to the Fibonacci sequence items. Then, the initial optimization point converges to the best method in a defined interval, which is considered to be the most effective solution of a one-dimensional unimodal question. We assume that there is a single-peak function on the interval. First, the technique starts to select two feasible points and performs the first iteration within a given range. Then, we need to reduce the area of the initial box to a sufficiently small box, involving the minimum value of the unimodal function $f(x)$ (after an iterative step). The smallest lie can be reduced by providing the function value known in two different ranges of points. The realization of the classic Fibonacci sequence optimization algorithm was given in [36], and no further detailed description is given here.

3.2. The Fibonacci Branch Structure and the FBS Algorithm. The basic Fibonacci strategy has difficulty effectively solving the multivariable problem, nor can it reliably evaluate the best fit of the multimodal function [35]. The FBS algorithm mainly uses the framework constructed by segmentation points and endpoints in the basic structure to search for the global optimal solution. Comparing the fitness value of each search point by calculation, the search point closest to the global optimal solution is obtained. In the next iterative calculation, we set the point with the best fitness value obtained in this optimization at the top of the search point set and the points corresponding to the suboptimal fitness value are arranged in the order from good to bad below the best. Through continuous iteration, the search point set is updated in each optimization stage. Then, the algorithm can complete the optimization of the objective function in the search space while the Fibonacci branch grows. The basic structure of the Fibonacci algorithm is shown in Figure 5.

In Figure 5, there are three points in D-dimensional Euclidean space, $\vec{x}_A$, $\vec{x}_B$, and $\vec{x}_C$. $\vec{x}_A$ and $\vec{x}_B$ indicate the coordinates corresponding to the search endpoint of the tree structure, which can be generated by the specified optimization rule, and $\vec{x}_C$ indicates the coordinates of the partition point obtained according to the given calculation criteria. In the search process, they should satisfy the following equation:

$$\frac{\left\|\vec{x}_C - \vec{x}_A\right\|}{\left\|\vec{x}_B - \vec{x}_A\right\|} = \frac{\left\|\vec{x}_B - \vec{x}_C\right\|}{\left\|\vec{x}_C - \vec{x}_A\right\|} = \frac{F_p}{F_{p+1}}, \tag{2}$$

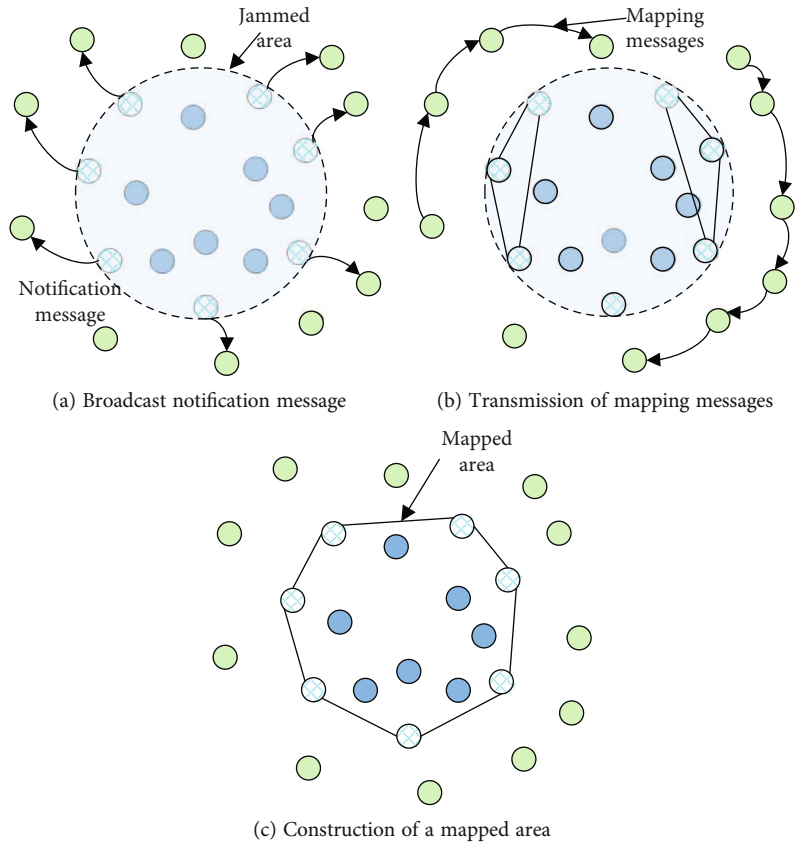
where F_p represents the pth Fibonacci number.

Considering the minimum multimodal function of multivariable $f(\vec{X})$ in the search space, the calculation formula of the split point is

$$\vec{x}_C = \begin{cases} \vec{x}_A + \dfrac{F_p}{F_{p+1}}\left(\vec{x}_B - \vec{x}_A\right), & f\left(\vec{x}_A\right) < f\left(\vec{x}_B\right), \\ \vec{x}_A + \dfrac{F_p}{F_{p+1}}\left(\vec{x}_A - \vec{x}_B\right), & f\left(\vec{x}_A\right) \geq f\left(\vec{x}_B\right). \end{cases} \tag{3}$$

In [37], a similar algorithm was described but the process and theory of the algorithm were not described in detail. This section expounds on the main part of FBS, expounds on the implementation content of FBS, and standardizes the implementation process of FBS.

Considering the basic structure of FBS, the process of obtaining the global optimal solution, including the process of building searching elements in FBS, is delimited into two stages: global search and local optimization [38]. These two stages are the corresponding rules of interaction. $\mathbf{G}_p$ is the set of objective function points to be searched in the pth iterative optimization stage; $\text{len}(\mathbf{G}_p) = F_p$ is the number of whole sets, where F_p is the intensity of the Fibonacci wave. Using the corresponding interactive optimization initial value and fitness value, the segmentation points are obtained through formula (3). After a comparison, the algorithm obtains the best fitness value corresponding to the latest best solution. In the next iteration stage, the best advantage of the adaptable value is concentrated in the corresponding front of the set and the nodes of the suboptimal adaptable value are



(a) Broadcast notification message (b) Transmission of mapping messages

(c) Construction of a mapped area

Figure 4: Overview of sensors collaboratively mapping a jammed area.

Table 1: Notations and definitions.

Notations	Definitions
N_S	The number of homogeneous sensors
$\mathbf{S}$	The coordinates of all of the sensors
$\mathbf{B}$	The coordinates of all of the boundary sensors
N_B	The number of boundary sensors
R	The branch depth
F_j	The jth Fibonacci sequences
$\mathbf{G}_p$	The set of objective function points to be searched in the pth iterative optimization stage
$\hat{\mathbf{r}}$	The estimated result of the jammer's coordinates
$\mathbf{r}$	The real location of the jammer
$\vec{x}_A, \vec{x}_B$	Search endpoint
$\vec{x}_C$	Split point

placed in the order from good to bad in the optimal selection. Through the abovementioned operations, we update points in each optimization stage and increase the Fibonacci path and community optimization search space.

In the process of FBS optimization, it is necessary to update the search endpoint according to two interactive iterative criteria and combine the calculation formula of the subdivision point to calculate the subdivision point. The two iterative updating criteria are as follows:

In rule one, the end nodes $\vec{x}_A$ and $\vec{x}_B$ of the structure are indicated as follows:

$$\begin{aligned} \left\{\vec{x}_A\right\} &= G_p = \left\{\vec{x}_q | q = [1, F_P]\right\}, \\ \left\{\vec{x}_B\right\} &= \left\{\vec{X} \middle| \vec{X} \in \prod_{f=1}^{D} \left[\vec{x}_{lb}^{f}, \vec{x}_{ub}^{f}\right]^u\right\}, \end{aligned} \tag{4}$$

$\mathbf{G}_p$ is composed of the coordinates of all the search points in the pth iteration, $\vec{x}_q$ are the search nodes in set $\mathbf{G}_p$, and q is the sequence number corresponding to the first to the pth Fibonacci sequences. $\vec{x}_A$ takes all points of $\mathbf{G}_p$ in the pth iteration. The other unselected endpoints $\vec{x}_B$ randomly take nodes, and the length of $\vec{x}_B$ is equivalent to F_p. When the dimension is f, D is the dimension of the points and the search nodes are between $\vec{x}_{lb}^{f}$ and $\vec{x}_{ub}^{f}$. Given that $\forall \vec{x} \in \{\vec{x}_B\}$, the component x in point $\vec{X}$ is a random variable uniformly distributed in the interval $[\vec{x}_{lb}, \vec{x}_{ub}]^U$, in which the normal characteristic U represents the uniform distribution of the variable, and the probability distribution of the component can be calculated as

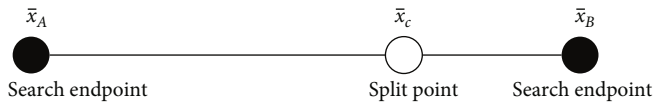
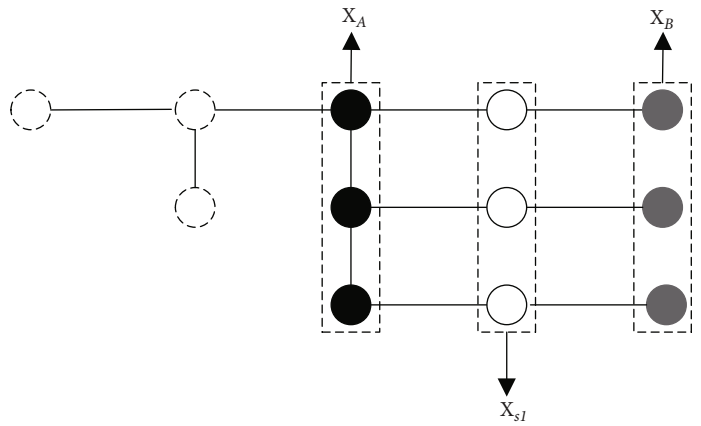
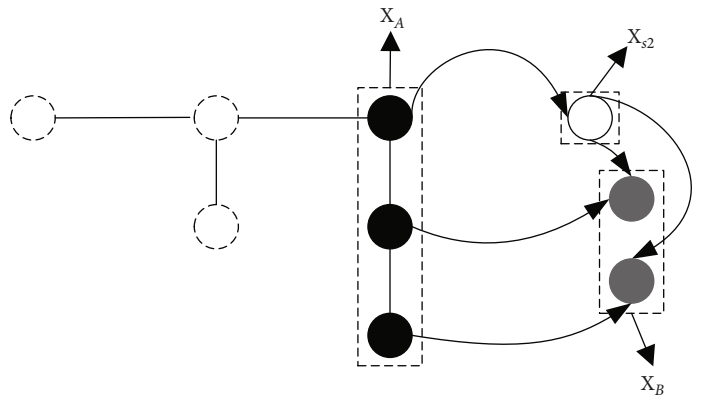


Figure 5: Basic search structure of the proposed Fibonacci branch search.



(a) Fibonacci scatter search global search phase



(b) Local optimization phase of Fibonacci scatter search

Figure 6: Schematic diagram of the Fibonacci rambler tectonic process.

$$P(x) = U\left(\vec{x}_{lb}, \vec{x}_{ub}\right) = \frac{1}{\vec{x}_{ub} - \vec{x}_{lb}}. \tag{5}$$

We can use the end nodes $\vec{x}_A$ and $\vec{x}_B$ to determine the partition points $\vec{x}_{S1}$ through equation (3).

In rule two, in the local optimization stage, assume that $\vec{x}_{\text{best}}$ represents the search point with the optimal fitness value in the iterative process of the algorithm, as follows:

$$\vec{x}_{\text{best}} = \text{best}\left(\mathbf{G}_p\right). \tag{6}$$

Best$(\cdot)$ is the optimal fitness value search node in the search node set.

Afterwards, we define the end nodes $\vec{x}_A = \vec{x}_{\text{best}}$ and find

$$f\left(\vec{x}_A\right) = \min\left\{f\left(\vec{x}_q\right), q = \left[1, F_p\right]\right\},$$
$$\vec{x}_B = \left\{\vec{x}_q \middle| \vec{x}_q \in \mathbf{G}_p \wedge \vec{x}_q \neq \vec{x}_A\right\}. \tag{7}$$

According to the end points defined in formula (7), the partition point $\vec{x}_{S2}$ can be determined according to the split point calculation formula in the second local optimization stage.

According to the abovementioned two interactive search rules, two different optimization stages generate $3F_p$ new points, involving endpoints $\vec{x}_A$ and $\vec{x}_B$ and segmentation points $\vec{x}_{S1}$ and $\vec{x}_{S2}$. By assessing the cost function of the new nodes, the fitness of the new points is determined and the new points are ranked from good to bad according to

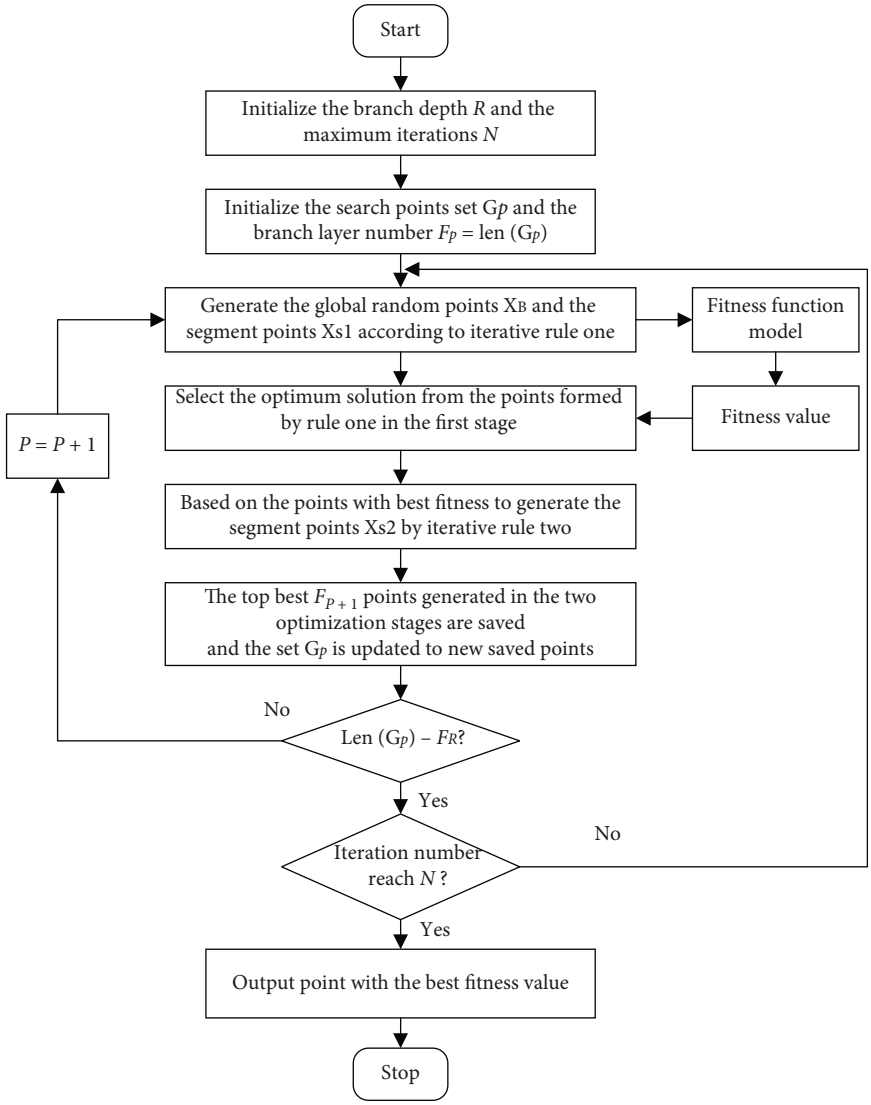
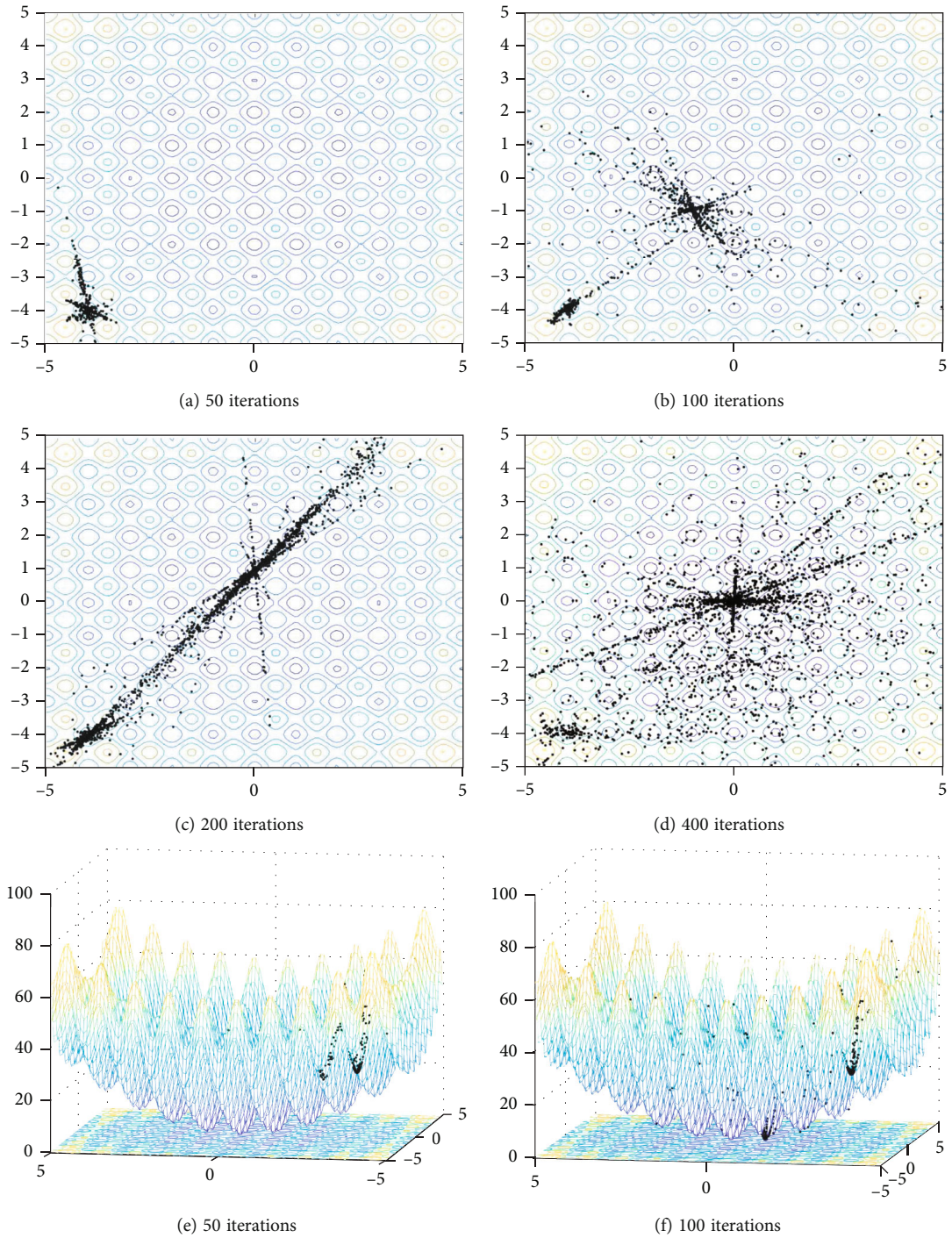


Figure 7: Flowchart of the Fibonacci branch search algorithm.

Input: **B**, R, N,$\mathbf{G}_p$
Output: $\hat{\mathbf{r}}$
$F_p \longleftarrow \text{len}(\mathbf{G}_p)$
for $i = 1 : N$ do
  for $j = p : R$ **do**
    Create the overall random nodes $\vec{x}_B$ and the split points $\vec{x}_{S1}$ by the rule one
    Calculate the fitness value for each search point based on the coordinates of the search point to all sensors in **B**
    Find the best result from the nodes formed by rule one
    Based on the points with optimum fitness value to adaptable points $\vec{x}_{S2}$ by rule two
  Sort the all-search points from good to bad according to the fitness value of every search point
  The top best F_{j+1} points are retained and the search points set $\mathbf{G}_j$ is renewed to the new retained points
  end for
  end for

Algorithm 1: Location-aware of the jammer based on FBS in WSNs.



(a) 50 iterations

(b) 100 iterations

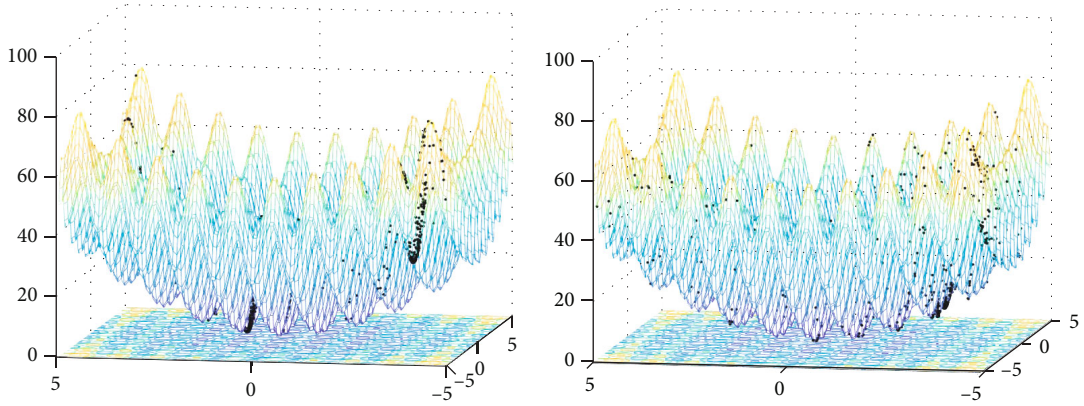
(c) 200 iterations

(d) 400 iterations

(e) 50 iterations

(f) 100 iterations

Figure 8: Continued.



(g) 200 iterations (h) 400 iterations

Figure 8: Behavior results of the point location history in FBS.

the fitness. We take the Fibonacci series as the total number of search nodes, and we need to save the most suitable F_{p+1} set of these points. Then, we need to remove other $3F_p - F_{p+1}$ nodes. The search space set in the pth iteration is updated from the saved points.

Corresponding to the global and local search stages, a schematic diagram of the Fibonacci branching process is shown in Figure 6.

The depth of the Fibonacci branching layer shown in Figure 6 is set to the expected value at the beginning, and the total number of nodes in each branching layer is stored in the Fibonacci sequence. In Figure 6, the white dotted circle is the search point set in the previous iteration process, the black solid border circle represents the endpoint of this iteration $\vec{x}_A$, and the gray real circle represents the global random endpoint $\vec{x}_B$. Figure 6(a) depicts the global search phase, which is the first stage of the whole process, in which the partition point $\vec{x}_{S1}$ represented by a white circle solid is made up based on the uniformly distributed points and $\vec{x}_A$. In Figure 6(b), other end nodes are fitted optimally in the local optimization stage $\vec{x}_A$ and $\vec{x}_B$ in the current iterative space is merged. Then, a new split node $\vec{x}_{S2}$ is obtained through the iterative rules. The adaptable values of $\vec{x}_A$, $\vec{x}_B$, $\vec{x}_{S1}$, and $\vec{x}_{S2}$ are assessed, and the optimal F_{p+1} solution of the objective function evaluation is preserved.

Figure 7 discloses a flow chart of a general process for a specific implementation of FBS.

3.3. Proof of Reachability and Convergence in FBS for Global Optimization of Multimodal Functions. In this section, according to the properties of abovementioned Fibonacci, the reachability and convergence of Fibonacci are studied. Through strict mathematical proof, it is proven that the FBS-based algorithm proposed in this paper can determine the global optimal solution and ensure that the FBS algorithm converges to the optimal solution.

3.3.1. Accessibility Investigation of the FBS Algorithm. Mathematical proof I FBS obtains the set of the solution objective function space by searching the reachable set in the space.

From the characteristics of the abovementioned algorithm, we know that after a sufficiently long iteration $n < +\infty$, rule 1 generates uniformly distributed endpoints $\vec{x}_B$ in a constrained space, such as $\forall \vec{X} \in \vec{x}_B, \vec{X} = (\vec{x}_d)_{D\times 1}$. D are dimensions of points, their probability distribution is $P(\vec{x}_d) = U(\vec{x}_{\min}, \vec{x}_{\max})$, and $\vec{x}_d$ satisfies the following relationship of the objective function field:

$$\int^{D} \cdots \int \frac{1}{\vec{x}_{\max} - \vec{x}_{\min}} d\vec{x}_d = 1. \tag{8}$$

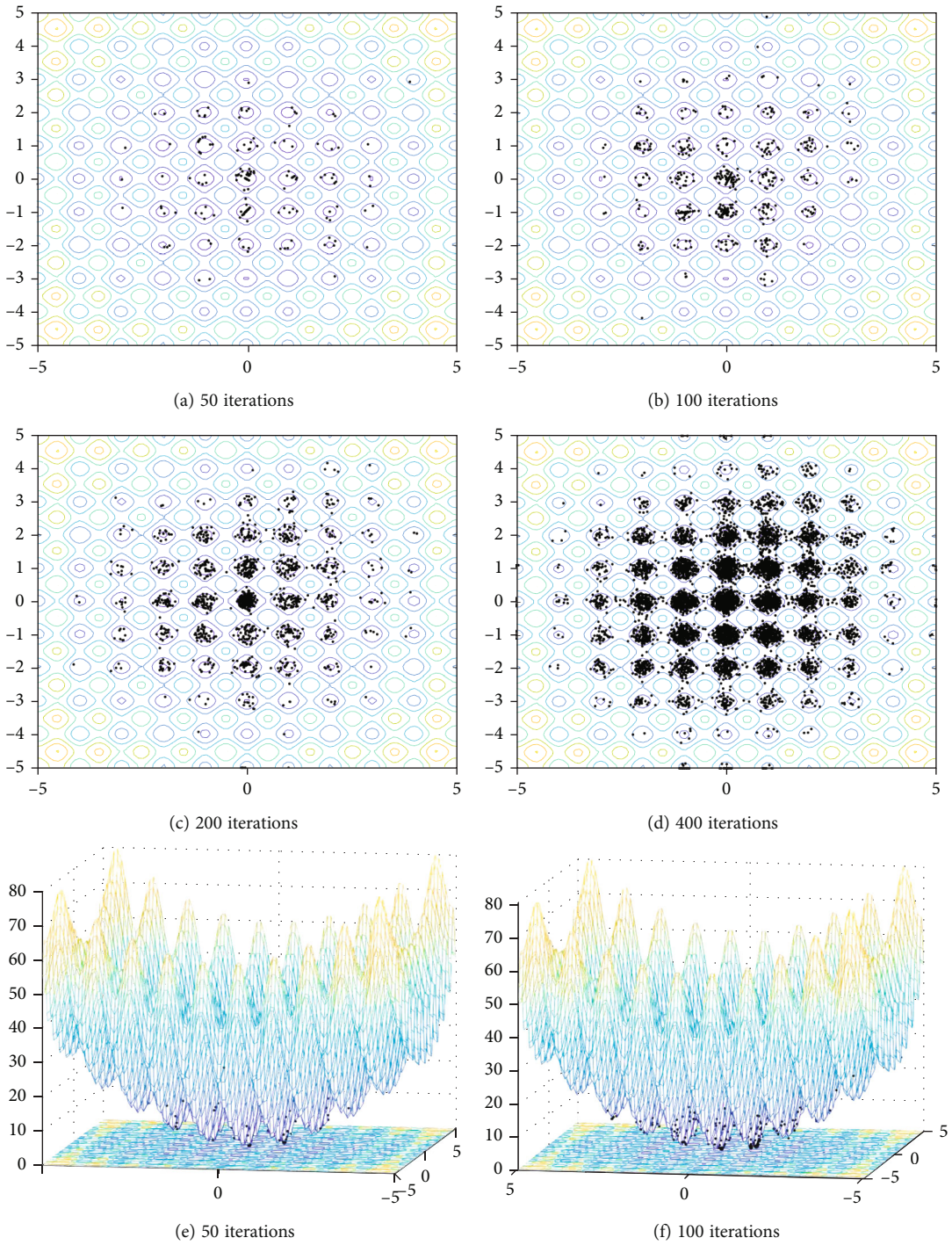
The abovementioned theoretical proof process shows that $\forall \vec{X} \in \vec{x}_B$ obeys $\vec{X} \in B$; optimization set B is the reachable set of $\vec{x}_B$ obtained by the FBS algorithm.

In mathematical proof II, for the FBS algorithm, the overall situation optimality of the objective function in the search field is feasible.

Assume that the overall optimal solution $\vec{x}^*$ of the search field is in the field B. It is proved by theory that the solution $\vec{x}^*$ is in reachable B. After that, we assume that the probability in a uniform distribution is P and then assume that the result obtained by the algorithm is $\vec{x}^L$. On the basis of the proof, after a sufficiently long iteration, the probability of FBS reaching the saddle point is $P^* \le \prod_{i=1}^{n}(1-P)$ to obtain the result

$$\lim_{n\longrightarrow+\infty} P^* \le \lim_{n\longrightarrow+\infty} \prod_{i=1}^{n}(1-P) = 0. \tag{9}$$

From this point of view, the FBS algorithm can obtain the overall best solution of the objective function in the search space.



(a) 50 iterations

(b) 100 iterations

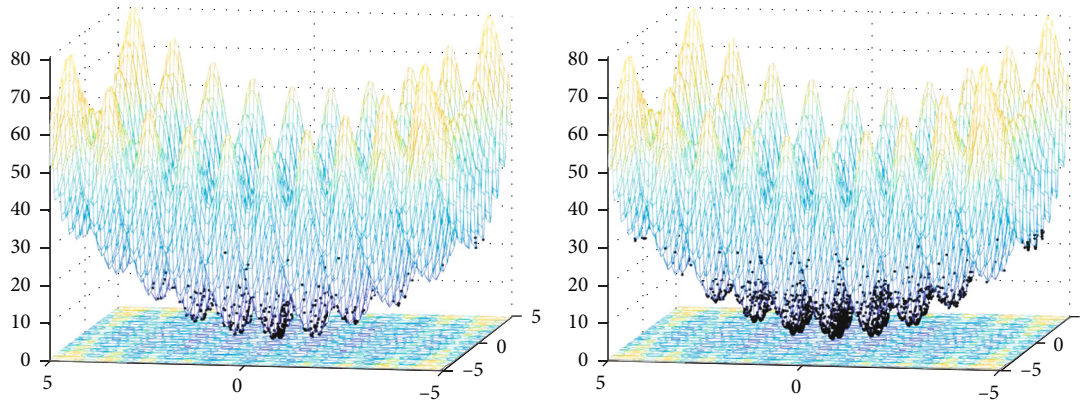
(c) 200 iterations

(d) 400 iterations

(e) 50 iterations

(f) 100 iterations

Figure 9: Continued.



(g) 200 iterations (h) 400 iterations

Figure 9: Behavior results of the point location history in GSA.

Table 2: Simulation experiment parameters.

Param.	Value
Size of area	(1 km × 1 km)
Number of sensors N_S	100
Transmitting range of sensors R_S	150 m
Jamming range of jammer R_J	150 m
Routing protocol	OSPF

Therefore, according to the abovementioned proofs, we can obtain that the overall optimal solution of the objective function field is accessible in the search field.

3.3.2. Convergence Analysis of the FBS Algorithm. Assuming that FBS achieves a sufficiently long iteration $n < +\infty$ to find the overall optimal solution, the existing G of the basic construction generated by FBS could become a gradually optimized node set $\{\vec{x}_T\} = \{\vec{x}_t | \vec{x}_{\text{best}} \in S_t\}, t = 1, 2, \cdots, n$. $\vec{x}_{\text{best}}$ is the best result of the iteration set. Afterwards, we construct the probability $P(t) = P(|\vec{X}^* - \vec{x}_t| \geq \varsigma)$ of FBS converging to the overall best result. ς indicates a small pinned variable. According to the rules $P(t) = P_{\bar{X}}(t) + P_\varsigma(t)$, $P_{\bar{X}}(t)$ is the chance of creating an equidistribution of random nodes of the field defined by rule 1, while $P_\varsigma(t)$ is the chance of creating an equidistribution of random nodes of the field defined by the radius parameter ς of rule 2. Thus, when there are t iterations of FBS, the search point does not reach the ς interval region around the global optimal solution, $\vec{X}^*$, $\tilde{P}(t) = P(|\vec{X}^* - \vec{x}_t| \geq \varsigma)$; then, $\lim_{t \longrightarrow +\infty} \tilde{P}(t) = 0$ is obtained and the following results are obtained:

$$P(n<+\infty) > P(t) = 1 - \tilde{P}(t) P(n<+\infty) > 0. \quad (10)$$

Letting $t \longrightarrow +\infty$, we can obtain

$$P(n<+\infty) = 1. \quad (11)$$

It is proven that FBS converges to the overall best result with a probability of 100%.

Through the proof in Sections 3.3.1 and 3.3.2 above, the results show that the designed FBS algorithm is feasible and convergent and finally can obtain the overall best result.

4. Jammer Location-Aware Method Based on Fibonacci Branch Search

FBS-based jammer location awareness mainly includes three steps: a selection of initial search points, a clarification of the fitness function, and a search point update.

In the process of location awareness, to narrow the search scope, the search points should be all within the mapped area. In the pth iteration, we assume that F_p search points are selected in the mapped area randomly and the coordinates of the ith search point are $\mathbf{p}_i = [x_i^P, y_i^P], i = 1, 2, \cdots, F_p$.

To evaluate the performance of the estimation result of the jammer's location quantitatively, the fitness function is designed in this section. When a jamming attack is conducted, the jammed area is approximately a circle and the distance between the jammer and the farthest boundary node is approximately equal to the jamming radius. Based on the analysis result in Section 2.2, we can obtain the coordinates of the boundary nodes. Then, the distances between the boundary nodes and the search point are calculated as the fitness function.

In the pth iteration, there are F_p search points in the mapped area and the coordinates of the ith search point are denoted as $\mathbf{p}_i = [x_i^P, y_i^P], 1 \leq i \leq F_p$. Then, the fitness function at the pth iteration for the ith search point is denoted as

$$\text{Fit}_i(p) = \frac{1}{N_B} \sum_{j=1}^{N_B} \left| d_{ij}(p) - \bar{d}_i(p) \right|, \quad i = 1, 2, \cdots, N_B, \quad (12)$$

where N_B is the number of boundary nodes, $[x_j^B, y_j^B]$ are the coordinates of the jth boundary node, and $d_{ij}(p)$ is the distance between the ith search point and the jth

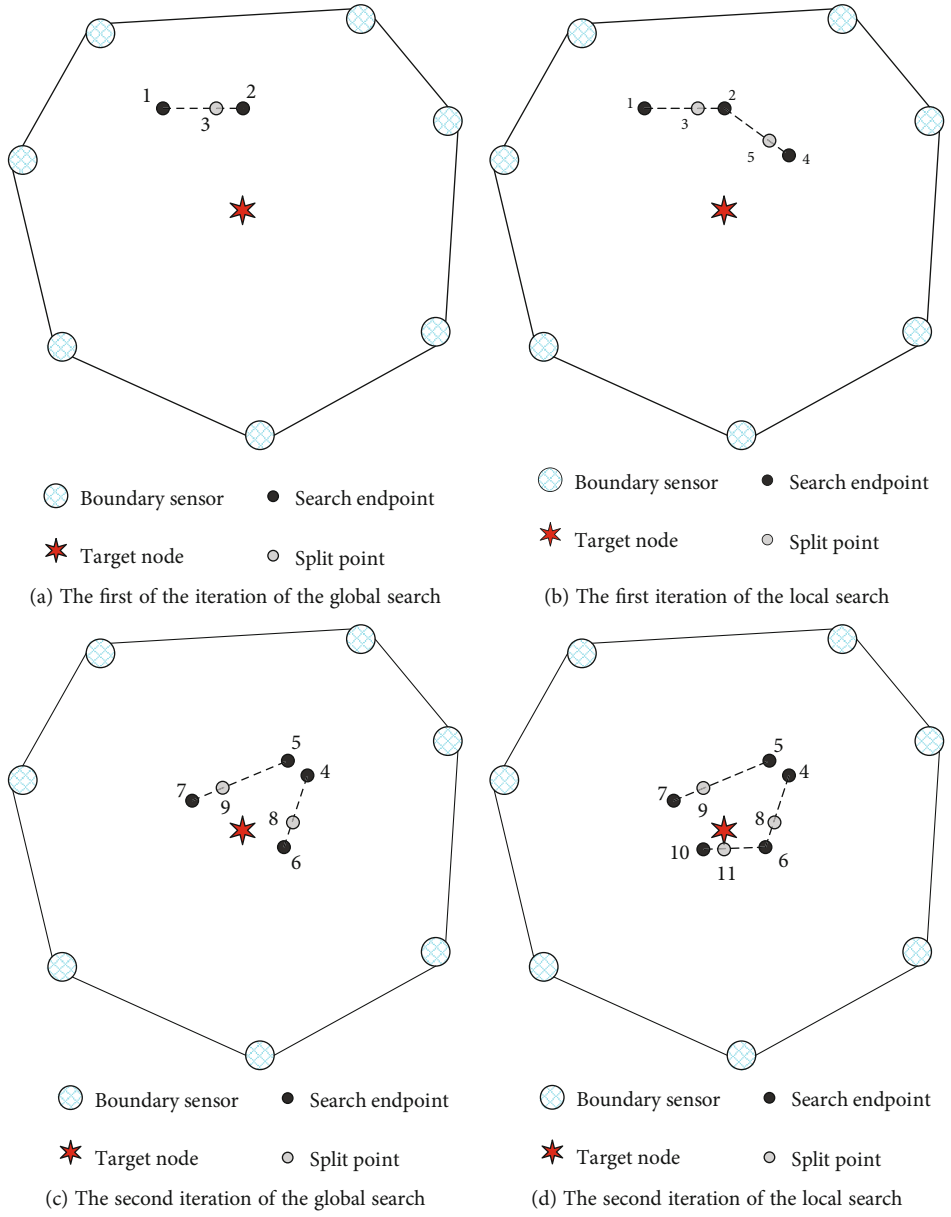


Figure 10: Schematic diagram of positioning process based on FBS.

boundary node. $\bar{d}_i(p)$ is the average distance between the i th search point and all of the boundary nodes. $d_{ij}(p)$ is calculated as

$$d_{ij}(t) = \sqrt{\left(x_i^P(p) - x_j^B\right)^2 + \left(y_i^P(p) - y_j^B\right)^2}, \quad (13)$$

and $\bar{d}_i(t)$ is calculated as

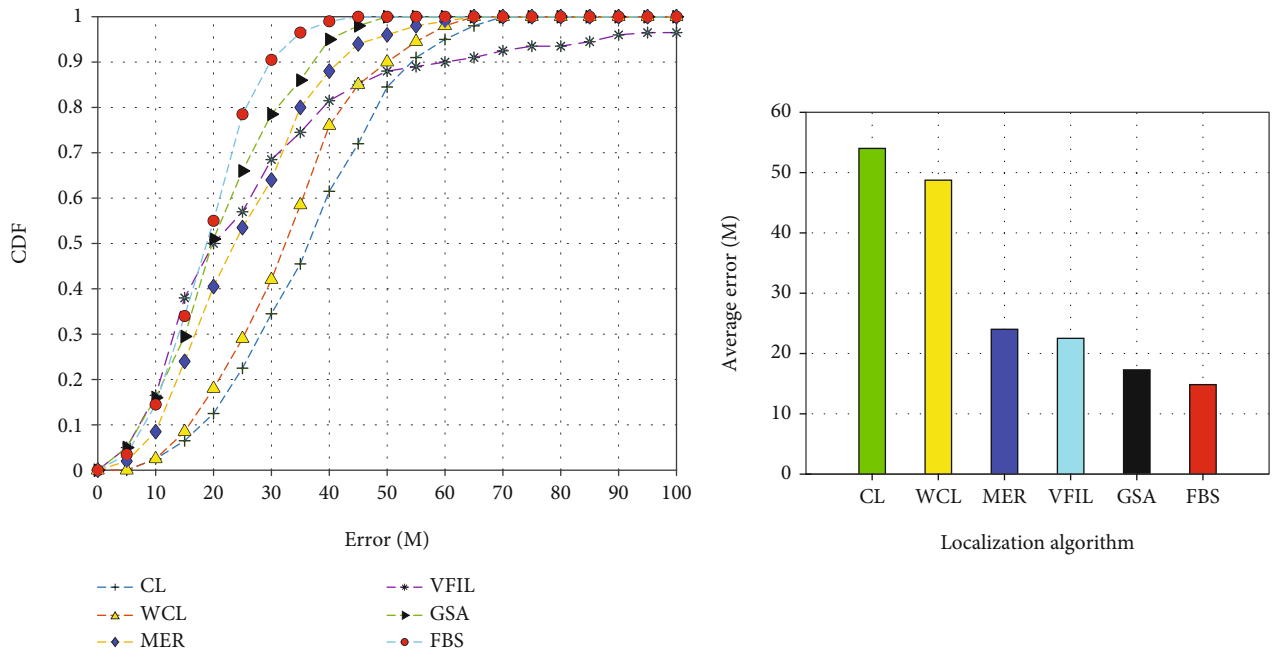
$$\bar{d}_i(t) = \frac{1}{N_B}\sum_{j=1}^{N_B}\left(d_{ij}(t)\right). \quad (14)$$

After the fitness value of each search point is calculated, all of these points are sorted in a descending order from the best to the worst; the top F_{p+1} points are saved, while the F_{p+1} search points form a new set $\mathbf{G}_{p+1}$ for the next optimization.

In the process of location awareness, the goal is to search for the search point of the smallest fitness value. The pseudocode of the FBS-based location-aware algorithm is shown in Algorithm 1. The input is the boundary sensors' coordinate set $\mathbf{B}$, the branch depth R, the maximum iterations N, and the initial search point set $\mathbf{G}_p$, and the output is the estimated result of the jammer's coordinates $\hat{\mathbf{r}}$.

The coordinates of the search node with the best adaptable value are selected as the coordinates of the jammer which is represented as $\hat{\mathbf{r}}$.

The implementation of FBS is completed by generating search elements and search branches. Assuming that the



(a) The cumulative distribution functions of location-aware errors

(b) Average location-aware errors

Figure 11: Analysis of location-aware errors when the number of sensors N_S = 100 and the jamming range R_J = 150 m.

optimization space is considered N_e when FBS is used in location awareness of the jammer, the comparison dimension in the search element is set as N_c; then, the algorithm complexity of FBS is approximate $O(C_f N_e (N_c)^2)$, where $C_f = \sum_{i=1}^{N}(F_i)$ is the final sum of the searched elements in the Fibonacci search branch under the maximum number of loop iterations. Thus, it can be seen from the abovementioned FBS complexity analysis that the computational complexity of our proposed optimization algorithm mainly depends on the dimension of the search element and the maximum number of iterations.

5. Simulation Experiments

In this section, we analyze the performance of the proposed FBS algorithms using abundant computer simulations. The simulation is realized in EXata, which is an excellent simulator for wireless networks, and the analysis of the simulation result is realized in MATLAB.

5.1. Location History of the Search Points in FBS for the Rastrigin Function. In this part, the proposed edge-back method with global optimization capabilities proves that the use of location historical search points in the optimization iteration process finds the global optimal solution instead of falling into the benchmark of local optimization and is combined with the gravity search algorithm (GSA). The benchmark function selected in this part is the Rastrigin function, which has multiple local optimal solutions and a globally optimal solution.

The performance of the proposed fullback movement trajectory search points is scattered in the best solution, and the search space for the optimal convergence point in Rastrigin is shown in Figure 8. This digital display of the fullback model can simulate the three-dimensional position of the historical search point and the trajectory profile in different iterations. To compare the performance of the FBS-based algorithm, we compared the algorithm with GSA and the result of GSA is shown in Figure 9. In FBS and GSA, the initial position of the search point is set at the extreme local optimal point.

As shown in Figures 8 and 9, the search points constantly explore the potential areas in the solution space and finally cluster around the global optimal value in a multimodal Rastrigin mode. The experiments displayed in Figure 9 show that with the increase of the number of iterations, the point cluster of the GSA algorithm is gradually in the extreme point and continues to maintain the local optimal conditions, with almost no particles searching for the global optimal extreme point. Based on a local optimum of the Rastrigin function, it is further proven that the algorithm is essentially trapped by the local optimum condition and falls into the local search space. It can be found under the same conditions from the search point trajectory and the 3D version shown in Figure 8. Although the Rastrigin function is asymmetric and multichannel and has different mountain levels, it is found that the global best challenge comes from many local variables. The minimum value is in the search space. It is worth noting that with the help of a global random search, FBS can be jumped out from the local optimal solution at the extreme point and from the notch solution at the local optimal point. Throughout the historical process of locating search points, in the two-dimensional and three-dimensional space iterations, the points converging to the global optimal condition and the

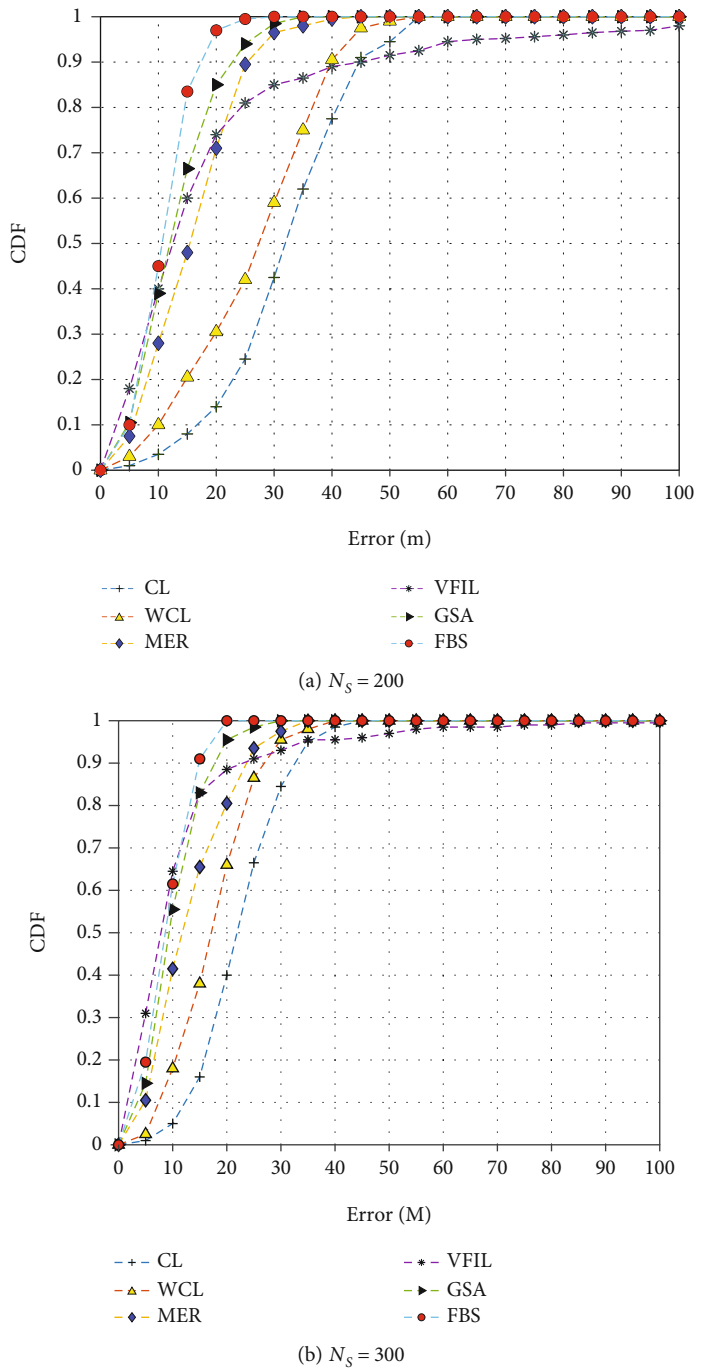


FIGURE 12: The CDF of location-aware errors when the number of sensors is different, and the jamming range $R_J = 150$ m.

regions initializing the optimal growth point are often scattered in the extreme points and gradually move to the optimal solution in the search space. After the first 50 iterations, more than half of the agents are close to the global optimal valley and begin to converge to the optimal valley. As the number of iterations increases, an increasing number of agents gather and disperse near the extreme point, especially in the global optimal target region. Finally, the search point finds the global optimum solution and converges to the global optimum, which can be investigated and reasoned by introducing the concept of global randomness into an endpoint generated by FBS rule 1. To ensure the convergence of the algorithm, local development and optimization capabilities are emphasized at other endpoints. Because the global random point performs a global search in space, it usually moves from a less suitable universe to a more suitable universe. The best universe will be saved and moved to the next search. Therefore, these capabilities and behaviors will help the FBS algorithm not fall into a local optimum and quickly converge to the optimal target point.

Simulation and discussion prove the effectiveness and convergence of the FBS algorithm. The algorithm proposed

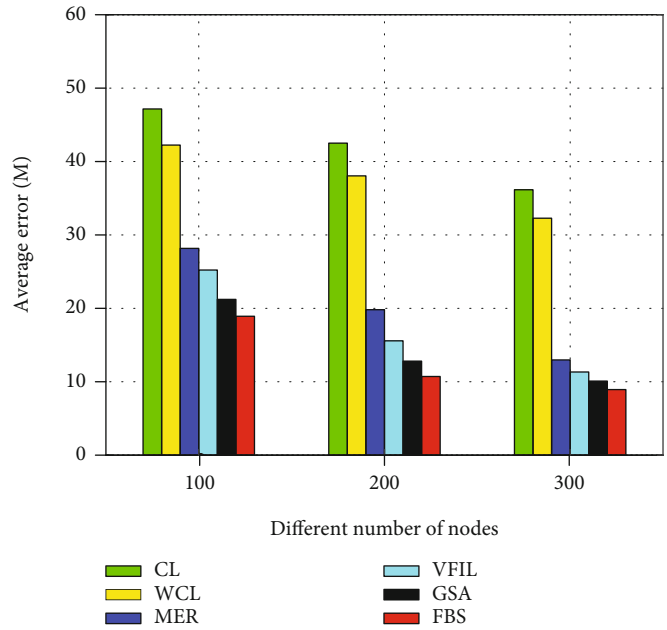


Figure 13: Influence of node density on the average location-aware errors of different algorithms.

in this article can find the global optimal solution in the solution space.

5.2. Parameters Setting and Benchmark in the Location-Aware Method. For every experiment, the sensors and the jammer are all randomly deployed in the (1 km × 1 km) area. The simulation experiment parameters of the sensors and the jammer are listed in Table 2.

To realize the quantitative analysis of location-aware results, the average location-aware error is utilized to evaluate the performance of each method, which is denoted by E and given by

$$E = \frac{\|\mathbf{r} - \hat{\mathbf{r}}\|}{N(E)}, \tag{15}$$

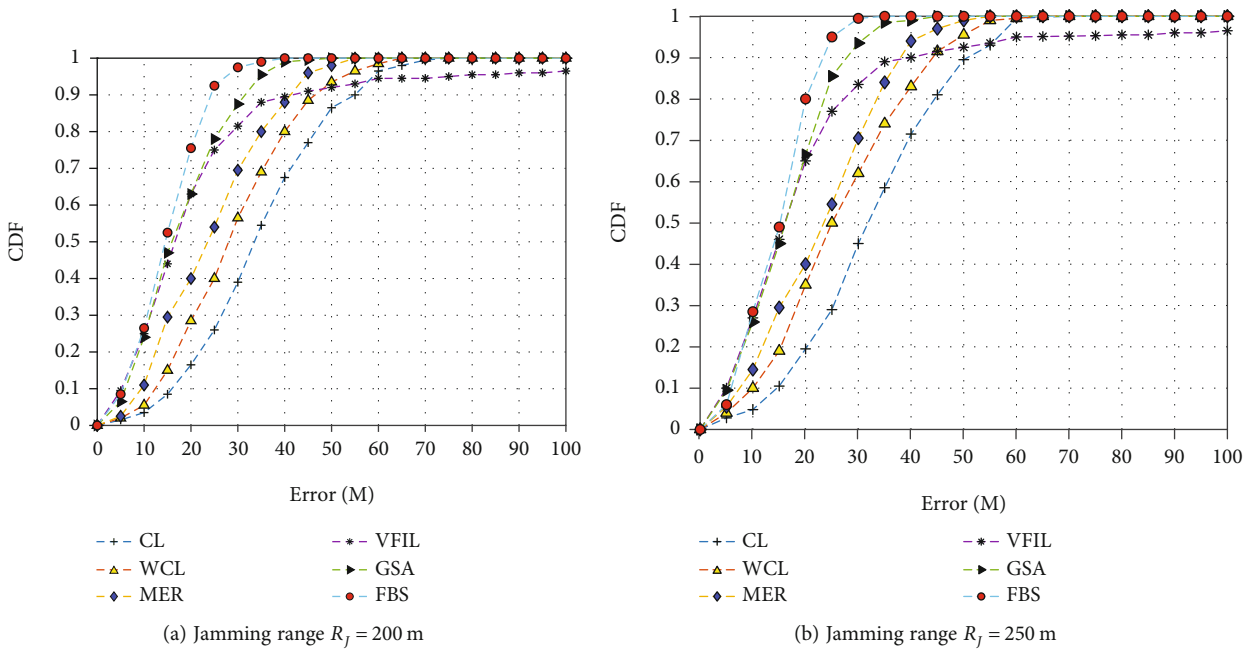
where $N(E)$ is the number of simulation experiments, $\mathbf{r}$ is the real location of the jammer, and $\hat{\mathbf{r}}$ is the jammer coordinate calculated by the algorithm. In addition, the cumulative distribution functions (CDFs) of the average location-aware error are also considered.

5.3. Performance Comparison and Result Analysis. Use the FBS-based noncooperative target node location algorithm to locate the communication node in the wireless sensor network and take Figure 10 as an example to describe the location process.

Figure 10(a) shows the global search stage during the first iteration. First, randomly select a search point (node 1) in the mapping area as the search endpoint. Then, randomly select a node (node 2) as the search endpoint and obtain the split point (node 3) according to the calculation formula of the split point. The fitness values of node 1, node 2, and node 3 are calculated and sorted, and the fitness value of node 2 is the best, which is used as the endpoint for local optimization. As shown in Figure 10(b), randomly select a search point (node 4) as the endpoint, find the split point (node 5) according to criterion 2, sort the fitness values of all nodes, and select the best two nodes (node 4 and node 5) used as the endpoints of the second iteration. In the global search process of the second iteration of Figure 10(c), two search points (node 6 and node 7) are randomly selected and the split point between node 4 and node 6 (node 8). The dividing point between node 5 and node 7 (node 9). Calculate and sort the fitness values of all nodes in the area. In the second local search stage of Figure 10(d), node 6 with the best fitness value is selected as the endpoint. Randomly select the search point (node 10) as another search point, and find the split point between node 6 and node 10 (node 11). Calculate and sort the fitness values of node 4, node 5, node 6, node 7, node 8, node 9, node 10, and node 11, and select node 6, node 10, and node 11 as the search for the third-iteration endpoint.

For every scenario, 10^3 experiments are conducted, and in every scenario, we compare the performance of FBS with CL, WCL, MER, VFIL, and GSA. The CDF of the average location-aware error when $N_S = 100$, $R_J = 150$ m is shown in Figure 11(a). Figure 11(b) shows the average location-aware errors of different location-aware algorithms. As we can see from the results, the error of the FBS-based location-aware algorithm is lower than those of the other algorithms.

Assuming that the area size is constant, the sensor number can be used to reflect the sensor density in the area. To analyze the influence of different node densities on the performance of different algorithms when the jamming range of the jammer is set as 150 m, the number of sensors is set as 200 and 300. The CDF of location-aware errors for different node densities is presented in Figure 12 after conducting 10^3 experiments independently. Figure 12(a) presents the CDF



(a) Jamming range $R_J = 200$ m (b) Jamming range $R_J = 250$ m

Figure 14: CDF of location-aware errors when the number of sensors $N_S = 100$; the jamming range is different.

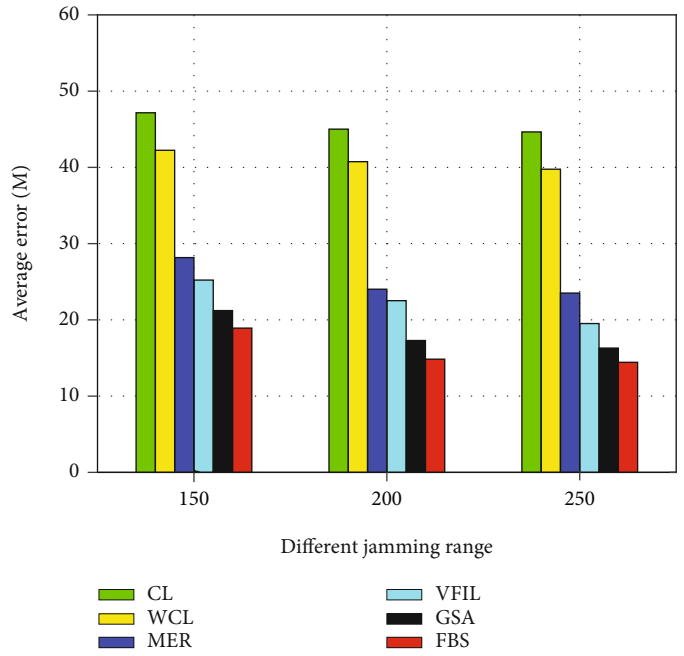


Figure 15: Influence of the jamming range on average location-aware errors of different algorithms.

of location-aware errors when $N_S = 200$, and Figure 12(b) presents the CDF of location-aware errors when $N_S = 300$. The average location-aware error change with different node densities is shown in Figure 13. As we can find in Figure 13, the average location-aware error of FBS is smaller than other algorithms, which means that the performance of FBS is better than the other algorithms. CL, WCL, MER, and VFIL are more sensitive to the distribution of sensors than GSA and FBS.

To analyze the influence of different jamming ranges on the performance of different algorithms, the number of sensors in the area is set as 100 and the jamming ranges are set as 150 m, 200 m, and 250 m. After conducting 10^3 experiments independently, the CDF of location-aware errors for different jamming ranges is presented in Figure 14. Figure 14(a) presents the CDF of location-aware errors when $R_J = 200$ m, and Figure 14(b) presents the CDF of location-aware errors when $R_J = 250$ m. The average location-aware

error change with different jamming ranges is shown in Figure 14(b). As we can see in Figure 15, the average location-aware error of FBS is smaller than that of the other algorithms, which illustrates the better performance of FBS; when $R_J = 250$ m, the average location-aware error of FBS is approximately 15 m.

6. Conclusion and Future Work

Predicting the active location of jammers and removing them can ensure the safety of WSNs. This paper proposes an evolutionary algorithm FBS based on the Fibonacci sequence method, which reduces the sensitivity of existing algorithms to the deployment and parameters of WSN jammer location awareness. Although this FBS has been studied in part of the literature, the accessibility and convergence of the FBS algorithm have not been proven. In this paper, the reachability and convergence of FBS are strictly proven, which further verifies the validity of the theory and supports the previous view. In the process of jammer location awareness, the boundary sensors in the jammed area are identified by the map service of the jammed area and the fitness function is constructed by the distance from the search point to all boundary sensors. After iteration, the position of the best fitness search point is estimated as the position of the jammer. The experimental results are compared with the Cl, WCL, mer, vfil, and GSA algorithms. The experimental results show that the location-aware algorithm based on FBS has a good performance, and the location-aware result is more accurate than that of the other algorithms.

In the real world, the jammer may be in a mobile state all the time, which increases the difficulty of positioning the jammer. At the same time, wireless sensor networks have hardware limitations on battery power due to cost; complex jammer location-aware algorithms will consume a lot of energy. Therefore, designing a simple and efficient algorithm that can get awareness of the location of mobile jammers is our future research direction.

Data Availability

No data were used to support this study.

Conflicts of Interest

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

Acknowledgments

This research is partially supported by the High-Level Talent Fund no. 22-TDRCJH-02-013, the National Natural Science Foundation of China under Grant nos. 62002377, 62072424, 61772546, 61625205, 61632010, 61751211, 61772488, and 61520106007, Key Research Program of Frontier Sciences, CAS no. QYZDY-SSW-JSC002, NSFC with nos. NSF ECCS-1247944 and NSF CNS 1526638, and in part by the National Key Research and Development Plan nos. 2017YFB0801702 and 2018YFB1004704.

References

[1] T. Wang, X. Wei, J. Fan, and T. Liang, "Jammer localization in multihop wireless networks based on gravitational search," *Security and Communication Networks*, vol. 2018, 11 pages, 2018.

[2] L. Chhaya, P. Sharma, G. Bhagwatikar, and A. Kumar, "Wireless sensor network based smart grid communications: cyber attacks, intrusion detection system and topology control," *Electronics*, vol. 6, no. 1, p. 5, 2017.

[3] A. Gaddam, T. Wilkin, M. Angelova, and J. Gaddam, "Detecting sensor faults, anomalies and outliers in the internet of things: a survey on the challenges and solutions," *Electronics*, vol. 9, no. 3, p. 511, 2020.

[4] S. Misra, R. Singh, and S. V. R. Mohan, "Information warfare-worthy jamming attack detection mechanism for wireless sensor networks using a fuzzy inference system," *Sensors*, vol. 10, no. 4, pp. 3444–3479, 2010.

[5] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: attack and defense strategies," *IEEE Network*, vol. 20, no. 3, pp. 41–47, 2006.

[6] S. Vadlamani, B. Eksioglu, H. R. Medal, and A. Nandi, "Jamming attacks on wireless networks: a taxonomic survey," *International Journal of Production Economics*, vol. 172, pp. 76–94, 2016.

[7] L. Ye, J. Fulong, L. Hao, W. Jianhui, H. Chen, and Z. Meng, "Interference robust channel hopping strategies for wireless sensor networks," *China Communications*, vol. 13, no. 3, pp. 96–104, 2016.

[8] M. S. Al-kahtani, L. Karim, and N. Khan, "Efficient opportunistic routing protocol for sensor network in emergency applications," *Electronics*, vol. 9, no. 3, p. 455, 2020.

[9] D. Ciuonzo, A. Aubry, and V. Carotenuto, "Rician MIMO channel- and jamming-aware decision fusion," *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 3866–3880, 2017.

[10] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: technical challenges, recent advances, and future trends," *Recent Advances, and Future Trends*, vol. 104, no. 9, pp. 1727–1765, 2016.

[11] T. Wang, X. Wei, J. Fan, and T. Liang, "Adaptive jammer localization in wireless networks," *Computer Networks*, vol. 141, pp. 17–30, 2018.

[12] A. Shoari and A. Seyedi, "Localization of an uncooperative target with binary observations," in *2010 IEEE 11th International Workshop on Signal Processing Advances in Wireless Communications*, pp. 1–5, Marrakech, Morocco, June 2010.

[13] X. Wei, Q. Wang, T. Wang, and J. Fan, "Jammer localization in multi-hop wireless network: a comprehensive survey," *IEEE Communication Surveys and Tutorials*, vol. 19, no. 2, pp. 765–799, 2017.

[14] F. Jameel, M. A. A. Haider, and A. A. Butt, "Robust localization in wireless sensor networks using RSSI," in *2017 13th International Conference on Emerging Technologies*, pp. 1–6, Islamabad, Pakistan, December 2017.

[15] B. Jin, X. Xu, and T. Zhang, "Robust time-difference-of-arrival (TDOA) localization using weighted least squares with cone tangent plane constraint," *Sensors*, vol. 18, no. 3, pp. 1–16, 2018.

[16] Y. Cong, X. Sun, and M. Sun, "Refinement of TOA localization with sensor position uncertainty in closed-form," *Sensors*, vol. 20, no. 2, pp. 1–19, 2020.

[17] S. C. K. Herath and P. N. Pathirana, "Optimal sensor arrangements in angle of arrival (AoA) and range based localization with linear sensor arrays," *Sensors*, vol. 13, no. 9, pp. 12277–12294, 2013.

[18] N. S. Ewa, "Localization in wireless sensor networks: classification and evaluation of techniques," *International Journal of Applied Mathematics and Computer Science*, vol. 22, no. 2, pp. 281–297, 2012.

[19] F. Thomas and L. Ros, "Revisiting trilateration for robot localization," *IEEE Education Society*, vol. 21, no. 1, pp. 93–101, 2005.

[20] I. Borg and P. J. Groenen, "Modern multidimensional scaling: theory and applications," *Journal of Educational Measurement*, vol. 40, no. 3, pp. 277–280, 2003.

[21] Y. Shang, W. Ruml, Y. Zhang, and M. P. Fromherz, "Localization from mere connectivity," in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pp. 201–212, Annapolis, Maryland, USA, June 2003.

[22] Y. Shang and W. Ruml, "Improved MDS-based localization," in *In Proceedings of the 2004 IEEE International Conference on Computer Communications (INFOCOM)*, pp. 2640–2651, Hong Kong, China, March 2004.

[23] G. T. F. de Abreu and G. S. Destino, "Super MDS: source location from distance and angle information," in *In Proceedings of the 2007 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 4430–4434, Hong Kong, China, March 2007.

[24] M. Amirinasab Nasab, S. Shamshirband, A. T. Chronopoulos, A. Mosavi, and N. Nabipour, "Energy-efficient method for wireless sensor networks low-power radio operation in Internet of things," *Electronics*, vol. 9, no. 2, p. 320, 2020.

[25] A. Shoari and A. Seyedi, "Target localization with binary observations: effect of censoring non-detecting sensors," in *In Proceedings of the 2011 IEEE International Symposium on Information Theory Proceedings*, pp. 2504–2508, St. Petersburg, Russia, August 2011.

[26] A. D. Wood, J. A. Stankovic, and S. H. J. A. M. Son, "A jammed-area mapping service for sensor networks," in *In Proceedings of the 24th IEEE International Real-Time System Symposium (RTSS)*, pp. 286–297, Cancun, Mexico, December 2003.

[27] W. Xu, W. Trappe, Y. Zhang et al., "The feasibility of launching and detecting jamming attacks in wireless networks," in *In Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 46–57, Urbana-Champaign, Illinois, USA, May 2005.

[28] N. Bulusu, V. Bychkovskiy, D. Estrin, and J. Heidemann, "Scalable, ad hoc deployable RF-based localization," in *In Proceedings of the Grace Hopper Conference on Celebration in Women Computer*, pp. 1–5, Vancouver, Canada, Octobor 2002.

[29] J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann, "Weighted centroid localization in zigbee-based sensor networks," in *In Proceedings of the 2007 IEEE International Symposium on Intelligent Signal Processing (WISP)*, pp. 1–6, Alcala de Henares, Spain, Octobor 2007.

[30] H. Liu, X. Wenyuan, Y. Chen, and Z. Liu, "Localizing jammers in wireless networks," in *In Proceedings of the 2009 IEEE International Conference on Pervasive Computing and Communications*, pp. 1–6, Galveston, TX, USA, March 2009.

[31] T. Cheng, P. Li, S. Zhu, and D. Torrieri, "M-cluster and X-ray: two methods for multi-jammer localization in wireless sensor networks," *Integrated Computer-Aided Engineering*, vol. 21, no. 1, pp. 19–34, 2014.

[32] Q. Wang, J. Fan, X. Wei, and T. Wang, "Multi-jammers localization formulti-hop wireless network," *The Journal of Communication*, vol. 37, no. 13, pp. 176–186, 2017.

[33] B. Yildiz and E. Karaduman, "On Fibonacci search method with _k_ -Lucas numbers," *Applied Mathematics and Computation*, vol. 143, no. 2-3, pp. 523–531, 2003.

[34] A. Etminaniesfahani, A. Ghanbarzadeh, and Z. Marashi, "Fibonacci indicator algorithm: a novel tool for complex optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 74, pp. 1–9, 2018.

[35] M. Subasi, N. Yildirim, and B. Yildiz, "An improvement on Fibonacci search method in optimization theory," *Applied Mathematics and Computation*, vol. 147, no. 3, pp. 893–901, 2004.

[36] J. O. Omolehin, M. A. Ibiejugba, A. E. Onachi, and D. J. Evans, "A Fibonacci search technique for a class of multivariable functions and ODEs," *International Journal of Computer Mathematics*, vol. 82, no. 12, pp. 1505–1524, 2005.

[37] X. Wang, D. J. Lyu, and Y. Dong, "Cutting parameters multi-scheme optimization based on Fibonacci tree optimization algorithm," *Control and Decision*, vol. 33, pp. 1373–1381, 2018.

[38] H. Zhang and F. Zeng, "Implementation of a novel Fibonacci branch search optimizer for the design of the low sidelobe and deep nulling adaptive beamformer," *International Journal of Microwave and Wireless Technologies*, vol. 12, no. 7, pp. 660–677, 2020.