

Research Article

Localization of Large-Scale Wireless Sensor Networks Using Niching Particle Swarm Optimization and Reliable Anchor Selection

Huanqing Cui ^{1,2}, Yongquan Liang ¹, Chuanai Zhou ³, and Ning Cao ³

¹College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

²School of Computer, National University of Defence Technology, Changsha 410063, China

³College of Information Engineering, Qingdao Binhai University, Qingdao 266555, China

Correspondence should be addressed to Ning Cao; ning.cao2008@hotmail.com

Received 4 April 2018; Revised 7 October 2018; Accepted 21 October 2018; Published 2 December 2018

Academic Editor: Mauro Femminella

Copyright © 2018 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.

Due to uneven deployment of anchor nodes in large-scale wireless sensor networks, localization performance is seriously affected by two problems. The first is that some unknown nodes lack enough noncollinear neighbouring anchors to localize themselves accurately. The second is that some unknown nodes have many neighbouring anchors to bring great computing burden during localization. This paper proposes a localization algorithm which combined niching particle swarm optimization and reliable reference node selection in order to solve these problems. For the first problem, the proposed algorithm selects the most reliable neighbouring localized nodes as the reference in localization and using niching idea to cope with localization ambiguity problem resulting from collinear anchors. For the second problem, the algorithm utilizes three criteria to choose a minimum set of reliable neighbouring anchors to localize an unknown node. Three criteria are given to choose reliable neighbouring anchors or localized nodes when localizing an unknown node, including distance, angle, and localization precision. The proposed algorithm has been compared with some existing range-based and distributed algorithms, and the results show that the proposed algorithm achieves higher localization accuracy with less time complexity than the current PSO-based localization algorithms and performs well for wireless sensor networks with coverage holes.

1. Introduction

Recently, wireless sensor networks (WSNs) have been applied in various applications, where the sensor nodes' location information helps us to understand the sensed data in spatial context, and it is also useful for the other protocols including coverage, deployment, and routing [1, 2]. However, sensor nodes are usually randomly deployed by a vehicle robot or aircraft, so they have no prior knowledge of their locations. A simple solution is to equip each sensor node with a global positioning system (GPS) device, but the weakness of high cost and energy consumption makes it is infeasible for large-scale deployment. Therefore, the determination of sensor nodes' positions, which is referred to as *localization*, is one of key technologies of WSNs.

The development of microelectronic, wireless communication leads to applications of large-scale WSNs containing

hundreds even thousands of sensor nodes deployed in a prohibitively large region. For example, CitySee [3] project deployed 1,200 sensor nodes in an urban area, and LAINet [4] is deployed in a whole forest. Because distributed localization approach allows each sensor node to estimate its own location without any central control or computation unit, which alleviates large communication overhead, so they are more suitable for localization of large-scale WSNs than centralized methods. Further, because many applications require absolute positions such as longitude and latitude, the localization algorithm should be *anchors based* [5, 6], where *anchors* are the sensor nodes knowing their global positions, and the rest sensor nodes, referred to as *unknown nodes*, are localized based on beacon messages broadcasted by anchors. The localization procedure is often iterative because anchors are significantly less than unknown nodes, so some unknown nodes are localized by their neighbour localized unknown

nodes instead of anchors. Both the anchors and localized unknown nodes are referred to as *reference nodes* in this paper.

Due to random deployment of WSNs, the anchors are not evenly distributed, so localization performance is seriously affected by two problems. The first is that some unknown nodes lack enough noncollinear neighbouring anchors to localize themselves accurately. The second is that some unknown nodes have many neighbouring anchors to bring great computing burden during localization. This paper proposes an algorithm combining niching particle swarm optimization (PSO) and reliable references selection to conquer the above problems, in order to localize as many unknown nodes as possible with high localization precision. The main contributions contain the following:

- (i) It proposes a reliable reference node selection approach based on localization precision, distance to unknown node, and degree of collinearity among reference nodes, in order to obtain a minimum and sufficient set of the best neighbouring references to localize an unknown node.
- (ii) It presents a niching PSO algorithm to localize an unknown node and selects the best parameters and stop criteria of PSO according to localization precision and number of iterations, in order to conquer ambiguous localization and improve localization precision.
- (iii) It presents localization algorithms for unknown nodes with less than three neighbour reference nodes, in order to localize as many unknown nodes as possible while guaranteeing localization precision.

The organizations of the rest of this paper are as follows. Section 2 briefly reviews existing localization algorithms for WSNs. Section 3 introduces localization problem and PSO algorithm and presents basic idea of the proposed algorithm. Sections 4, 5, and 6 propose details of our localization algorithm, namely, reliable reference node selection, niching PSO-based localization, and localization algorithms for unknown nodes with less than three neighbour reference nodes. Section 7 analyses time and communication complexities of the proposed algorithm in theory and compares the proposed algorithm with the other algorithms by simulations. The conclusion and future works are given in Section 8.

2. Related Works

Lots of localization algorithms have been developed, and most of them consist of two stages: measurement of geographical information between sensor nodes and estimation of positions of unknown nodes. According to different geographical information used in localization, these algorithms can be classified into *range-free* and *range-based*.

Typical range-free algorithms include centroid, MDS (Multidimensional Scaling), and DV-hop (Distance-Vector hop). Centroid uses the centroid [7] or weighted centroid [8] of positions of neighbour reference nodes as position estimation of an unknown node. These methods rely heavily

on number of anchors, and their localization errors are large [9]. MDS [10] computes all pair-wise shortest path between sensor nodes and then applies MDS to get an initial map. Centralized computation is used to estimate positions of sensor nodes, which imposes heavy communication burden on WSNs. MDS can also be applied by each sensor nodes itself in a distributed scheme [11], but the distributed version requires a procedure of local maps combination to determine the entire layout, which increases complexity and is impossible to complete combination if there are many disjoint patches in a given WSN [10]. DV-hop [12] consists of two message flooding phases to determine average hop length between two anchors, and an unknown node estimates its position by trilateration when it receives at least three anchor messages. There is much room to improve localization accuracy of DV-hop, and many enhancements have been proposed, such as checkout DV-hop [13], selective 3-anchor DV-hop [13], DV-hop algorithm based on locally weighted linear regression [14], etc., but they also need flood messages through WSNs. Both MDS and DV-hop are based on the shortest path between sensor nodes, but this path is often zigzag especially in WSNs with coverage holes. To cope with this problem, new methods are utilized to estimate distances between sensor nodes in two-stage centralized range-free localization algorithm [9] and connectivity-based anchor-free three-dimensional localization algorithm [15], but they still need message flooding. Hence, range-free algorithms consume a considerable amount of power when they are applied to large-scale WSNs. Moreover, these algorithms perform satisfactorily when the network has a relatively even node distribution, but their performance is poor for the network of nonuniform node distribution [15].

Range-based algorithms need equip additional hardware to measure distance or angle. The popular ranging techniques include RSS (Received Signal Strength), ToA (Time of Arrival), TDoA (Time Difference of Arrival), and AoA (Angle of Arrival) [16]. After obtaining range information between an unknown node and its neighbour reference nodes, equations about the location of this unknown node can be gained, and they can be solved by an optimization method, such as maximum likelihood [17], least-square calibration [18], second-order cone programming [19], etc. Because evolutionary algorithms require less computational efforts than traditional optimization algorithms, which are more suitable for implementation on individual sensor node [20], they have been pervasively applied to localization problems, such as artificial neural network [21], genetic algorithm [22], and fuzzy logic and extreme learning machines [23].

PSO algorithm performs better in terms of localization accuracy, computational complexity and/or convergence speed compared with the other optimization methods [24–30]. PSO and BFA (Bacterial Foraging Algorithm) are applied to localize a WSN deployed by an unmanned aerial vehicle in [24], and simulation results showed that PSO is faster than BFA. The proposed method in [25] divides unknown nodes into two modes: sensor nodes with at least three neighbour reference nodes and sensor nodes with less than three neighbour reference nodes. For an unknown node of the second mode, it utilizes a similar DV-Distance [12] approach

to estimate the distance to a multihop anchor and then applies PSO to localize. Gauss-Newton algorithm (GNA) and PSO are applied to localize the sparse WSNs in [26], and simulation results showed that PSO outperforms GNA with large pedometer error. H-best PSO (HPSO) proposed by [27] divides the swarm of particle into several subswarms, and each particle updates according to global and local best fitness. The hybrid PSO proposed in [28] replaces positions of a half of particles with positions close to the best particle, leading to faster convergence than standard PSO. The fuzzy-extreme learning machine with PSO [29] uses resultant force to move approximate node location closer to actual position. PSO-based improved DV-hop algorithm [30] uses PSO to correct estimated positions after DV-hop.

Generally, localization algorithms utilize all neighbour reference nodes to estimate location of an unknown node, which may lead to high localization error, so a reliable anchor selection process is needed. The methods proposed in [31] select three anchors based on minimal hop counts to an unknown node. The selective 3-anchor DV-hop algorithm [13] selects any three noncollinear anchors to form a 3-anchor group and correspondingly generates a 3-anchor estimated position. Reliable anchor pair selection method [32] selects reliable anchor pairs by observing their average hop progresses and estimates the distances to the reliable anchors by geometric approximation of the node. These methods are DV-based, so they are not feasible for range-based localization algorithms. A concept of *confidence value* is introduced in [33] to alleviate error propagation effect in localization of underwater WSNs, and this value is defined as a decreasing function of localization error, but it does not consider other factors, such as collinearity of anchors and distances to unknown node.

3. Background and Basic Idea of Proposed Localization Algorithm

3.1. Localization Problem Statement

Definition 1 (anchors). The sensor nodes whose positions are known before localization are called *anchors*. The set of anchors of a WSN is denoted by $\mathbf{AS} = \{A_i \mid i = 1, 2, \dots, M\}$, where M is the number of anchors. The position of anchor A_i is denoted by (x_{ai}, y_{ai}) .

Definition 2 (unknown nodes, actual position). The sensor nodes that need to be located by a certain localization algorithm are called *unknown nodes*. The set of unknown nodes of a WSN is denoted by $\mathbf{US} = \{U_i \mid i = 1, 2, \dots, N-M\}$, where $N-M$ is the number of unknown nodes. The *actual positions* of U_i is denoted by (x_{ui}, y_{ui}) , which is not known in real application.

Definition 3 (neighbour). Let R be the radius of communication range of sensor nodes. Given two sensor nodes C and D , if C is located in the radio range of D , C is referred to as a *neighbour* of D . Because all sensor nodes have the same communication range, D is also a neighbour of C .

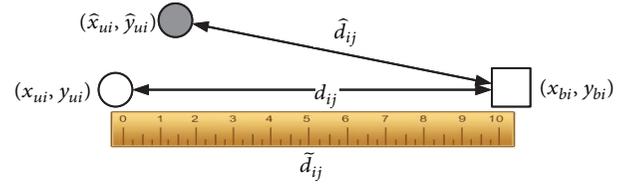


FIGURE 1: Three types of distances between sensor nodes.

Definition 4 ((range-based) localization). Give a WSN consists of N sensor nodes, including M anchors and $N-M$ unknown nodes, the distances between any pair of neighbour nodes can be measured. The (range-based) localization problem is to localize the unknown nodes using the positions of anchors and distances between neighbour nodes.

Definition 5 (estimated position). If an unknown node is localized during localization, it will get an *estimated position*, and it becomes a *localized unknown node*. The estimated position of U_i is denoted by $(\hat{x}_{ui}, \hat{y}_{ui})$.

Obviously, the purpose of localization is to make $(\hat{x}_{ui}, \hat{y}_{ui}) = (x_{ui}, y_{ui})$ for U_i .

Definition 6 (reference nodes). During localization, both anchors and localized unknown nodes are called *reference nodes*. The set of reference nodes is denoted by $\mathbf{RS} = \{B_i \mid i = 1, 2, \dots, M'\}$, $N \geq M' \geq M$. The position of reference node B_i is denoted by (x_{bi}, y_{bi}) . Obviously, if B_i is anchor A_j , $(x_{bi}, y_{bi}) = (x_{aj}, y_{aj})$; if B_i is localized unknown nodes U_k , $(x_{bi}, y_{bi}) = (\hat{x}_{uk}, \hat{y}_{uk})$.

Definition 7 (actual distance, measured distance, and estimated distance). An unknown node U_i and a reference node B_j have three types of distances:

- (i) *Actual distance*: distance between the actual positions of U_i and B_j . It is denoted by d_{ij} .
- (ii) *Measured distance*: distance measured by a certain technique. It is denoted by \tilde{d}_{ij} . Because any measurement technique has error e to some extent, we assume $\tilde{d}_{ij} = d_{ij} + N(0, d_{ij}e)$, where $N(0, d_{ij}e)$ is a Gaussian distribution.
- (iii) *Estimated distance*: distance between the estimated position of U_i and B_j . It is denoted by \hat{d}_{ij} .

Figure 1 shows an example of these types of distances between U_i and B_j .

Suppose unknown node U_i has n neighbour reference nodes B_1, B_2, \dots, B_n , we can solve the following equations to get $(\hat{x}_{ui}, \hat{y}_{ui})$:

$$\tilde{d}_{ij} = \sqrt{(x - x_{bj})^2 + (y - y_{bj})^2} \quad j = 1, 2, \dots, n \quad (1)$$

where (x, y) is unknown quantity to solve and (x_{bj}, y_{bj}) is position of B_j . Due to the distance measurement error e , $(\hat{x}_{ui}, \hat{y}_{ui}) \neq (x_{ui}, y_{ui})$, i.e., we cannot get the actual position

- (1) Initialize a swarm of particles in the given search space with random positions and velocities.
- (2) **while** stop criteria are not met **do**
- (3) Compute the fitness values of all particles.
- (4) Compute $Pbest_i$ for all particles, and $Gbest$.
- (5) Update particle i according to the following formula:

$$V_i = \omega V_i + c_1 \varphi_1 (Pbest_i - X_i) + c_2 \varphi_2 (Gbest - X_i), \quad X_i = X_i + V_i \quad (5)$$
- (6) **end while**

ALGORITHM 1: Basic procedure of PSO algorithm.

of B_j in general case. Using estimated position $(\hat{x}_{ui}, \hat{y}_{ui})$, the estimated distance \hat{d}_{ij} is

$$\hat{d}_{ij} = \sqrt{(\hat{x}_{ui} - x_{bj})^2 + (\hat{y}_{ui} - y_{bj})^2} \quad (2)$$

Since the actual distance d_{ij} is not known, the objective of localization is to minimize the difference between \hat{d}_{ij} and \tilde{d}_{ij} . Therefore, the problem of localizing U_i is formulated as

$$\begin{aligned} \min \quad & f(x, y) = \sum_{j=1}^n w_j (\hat{d}_{ij} - \tilde{d}_{ij})^2 \\ & = \sum_{j=1}^n w_j \left(\sqrt{(x - x_{bj})^2 + (y - y_{bj})^2} - \tilde{d}_{ij} \right)^2 \end{aligned} \quad (3)$$

$$\text{s.t.} \quad (1 - DOI)R \leq \sqrt{(x - x_{bj})^2 + (y - y_{bj})^2} \leq R$$

$$0 \leq x, y \leq L$$

$$j = 1, 2, \dots, n$$

where $w_j = 1/\tilde{d}_{ij} / \sum_{i=1}^n (1/\tilde{d}_{ij})$, which gives more influence to those reference nodes nearer to U_i , DOI represents the *degree of irregularity* of the communication range of sensor nodes, and L is the side length of the deployment region of WSN. In real applications, the communication range of sensor nodes is not a perfect circle due to multipath fading, unstable signal propagation, and noise, so DOI is introduced in (3). Given DOI , the communication range of a sensor node varies randomly in different directions between an upper bound R and a lower bound $(1 - DOI)R$ [34].

The first constraint guarantees the solution locates within the communication range of all neighbouring reference nodes, and the second constraint guarantees the solution locates within the deployment region. The third constraint denotes U_i has n neighbour reference nodes.

Due to the distance measurement error e , $f(x, y)$ in (3) is often a multimodal function, so we have to use optimization algorithm to solve it. It has been proved that PSO can solve this problem well in related references; this paper tries to improve its performance further.

Definition 8 (localization error). The distance between the actual and estimated positions of U_i is called its *localization error*, which is denoted by LE_i . Obviously,

$$LE_i = \frac{1}{R} \sqrt{(\hat{x}_{ui} - x_{ui})^2 + (\hat{y}_{ui} - y_{ui})^2} \quad (4)$$

Localization error is an important criterion to evaluate a localization algorithm.

3.2. Particle Swarm Optimization Algorithm. PSO is a popular multidimensional optimization algorithm, which models the behaviour of a flock of birds searching for corn [35]. In PSO, a swarm of *particles* explores in a D -dimensional hyperspace to search the global solution of an *objective/fitness function*. Three D -dimensional vectors are used to describe a particle i : (1) current location X_i ; (2) the previous location $Pbest_i$ of the best fitness; and (3) current velocity V_i . Besides, $Gbest$ denotes the position of the best particle so far. The basic procedure of PSO is shown in Algorithm 1.

In (5), c_1 and c_2 are, respectively, cognitive and social acceleration coefficients, φ_1 and φ_2 are random numbers uniformly distributed in $(0,1)$, and ω is inertia weight defined as

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{t_{max}} t \quad (6)$$

where t_{max} is maximum number of allowable iterations and ω_{max} and ω_{min} are, respectively, maximum and minimum weight. Further, velocities are bounded within $[-V_{max}, V_{max}]$ to prevent particles from exceeding the search space. The stop criteria in line 2 used in existing PSO-based localization algorithms are

$$f(X) \leq f_{opt} \vee t > t_{max} \quad (7)$$

where f_{opt} is the acceptable minimal (also the optimal) value of f .

Most of the existing PSO-based localization algorithms are based on the above PSO version, and we call it *standard PSO* (SPSO). This paper also investigates CPSO (PSO with constriction coefficient) [36], which applies *constriction coefficient* χ to prevent explosion, ensure convergence, and eliminate the arbitrary V_{max} parameter. The update formula of CPSO is

$$\begin{aligned} V_i &= \chi (V_i + c_1 \varphi_1 (Pbest_i - X_i) + c_2 \varphi_2 (Gbest - X_i)) \\ X_i &= X_i + V_i \end{aligned} \quad (8)$$

where $\chi = 2 / |2 - c - \sqrt{c^2 - 4c}|$, $c = c_1 + c_2 > 4.0$.

3.3. Basic Idea of Proposed Localization Algorithm. The localization procedure is distributed and iterative. The anchors broadcast beacon messages, including its identifier, coordinate, and information to measure distance, to its neighbour sensor nodes, and the unknown nodes receiving at least three beacon messages are localized. These localized unknown nodes become new reference nodes to provide beacon messages to their neighbours. This procedure is repeated until none of unlocalized unknown nodes can receive at least three beacon messages. If some unknown nodes are not localized yet, they request beacon messages of reference nodes in two and three hops to assist their localization. Therefore, an unknown node may appear as one of the following modes during localization:

- (i) **Mode 1:** it has at least three neighbour anchors.
- (ii) **Mode 2:** it has less than three neighbour anchors, but it has at least three neighbour reference nodes.
- (iii) **Mode 3:** it has less than three neighbour reference nodes.

The unknown node of Mode 1 and Mode 2 selects reliable reference nodes before estimating its location by niching PSO algorithm, and the unknown node of Mode 3 needs special localization algorithm. These algorithms are presented in the next three sections.

4. Reliable Reference Node Selection

Reliability of a reference node is based on its localization error and distance to an unknown node. Localization error cannot be computed in real applications due to the absence of actual positions of unknown nodes, but the sequence of iterative localization and G_{best} of PSO-based localization algorithm represent localization error to a great extent. Therefore, the reliability of reference node B_i to localize an unknown node U is defined as

$$conf_i = \alpha g(Lseq_i) + \beta g(\tilde{d}_i) + \gamma g(Fit_i) \quad (9)$$

where $Lseq_i$, \tilde{d}_i , and Fit_i are, respectively, the sequence of iterative localization of B_i , measured distance between B_i and U , and fitness of B_i . α , β , and γ are weights of different parts of $conf$. $Lseq_i$ is defined as

$$Lseq_i = \begin{cases} 1, & \text{if } B_i \text{ is an anchor} \\ \infty, & \text{if } B_i \text{ is localized by Algorithm 5} \\ 1 + \max_j \{Lseq_{C_j}\}, & \text{otherwise} \end{cases} \quad (10)$$

where C_j is a reference node to localize B_i . Fit_i is defined as

$$Fit_i = \begin{cases} f(\hat{x}_{b_i}, \hat{y}_{b_i}), & \text{if } B_i \text{ is localized by Algorithm 3} \\ 0, & \text{if } B_i \text{ is an anchor} \\ \max_j \{f(\hat{x}_{c_j}, \hat{y}_{c_j})\}, & \text{if } B_i \text{ is localized by Algorithm 4 or 5} \end{cases} \quad (11)$$

where $(\hat{x}_{c_j}, \hat{y}_{c_j})$ is the estimated location of C_j to localize B_i .

TABLE 1: $conf$ and $conf'$ and dc of B_i in Figure 1.

	B_1	B_2	B_3	B_4	B_5	B_6
$conf$	1.54	1.32	0.85	0.93	1.25	0.43
$conf'$	1.37	1.19	0.79	0.88	1.21	0.40
dc	-	-	0.05	0.07	0.06	0.02

Further, $g(x)$ of (9) is a normalization function to map variable $x \in [x_{min}, x_{max}]$ to $g(x) \in [0, 1]$, which is defined as

$$g(x) = \begin{cases} 1, & \text{if } x_{max} = x_{min} \\ 1 - \frac{x - x_{min}}{x_{max} - x_{min}}, & \text{if } x_{max} \neq x_{min} \wedge x < \infty \\ 0, & \text{if } x = \infty \end{cases} \quad (12)$$

Note that x_{max} and x_{min} are derived from the corresponding variables (i.e., $Lseq_i$, \tilde{d}_i , or Fit_i) of all neighbour reference nodes of U , instead of all reference nodes of whole WSN. Moreover, $x_{min} \geq 0$ according to the definition of each variable.

Reference node with the largest $conf$ is chosen as the first reliable one. If there are several reference nodes have the same $conf$, each of them can be selected as the first reliable reference node. The second one is chosen based on the following value:

$$conf'_i = \alpha g(Lseq_i) + \gamma g(Fit_i) \quad (13)$$

The reason of removing the second part of (9) is illustrated in Figure 2(a). Suppose U has 6 neighbour reference nodes B_i ($i = 1, 2, \dots, 6$), and $conf$ and $conf'$ are shown in Table 1. B_1 is chosen as the first one. If we still use $conf$ to select the second one, B_2 is the best choice. Whichever reference node we choose as the third one, the triangle formed by these chosen reference nodes will be too narrow and long to judge whether they are collinear or not.

Reference node with the largest $conf'$ is chosen as the second, so B_5 is chosen in Figure 2. If several reference nodes have the same $conf'$, the farthest to the first chosen reference node is selected. After selecting two reference nodes, the rest reliable reference nodes are chosen based on $conf$ and degree of collinearity among reference nodes. Any three reference nodes can form a triangle, and we utilize the minimal internal angle dc of this triangle to measure degree of collinearity of these reference nodes. If dc is smaller than a predefined threshold δ , the observed reference node is nonreliable. As shown in Figure 2(b), B_2 cannot be chosen after B_1 and B_5 if $\delta = 0.1$, although $conf$ of B_2 is the second largest. B_4 is chosen as the third one. This process is repeated until all neighbour reference nodes of U are investigated.

The detailed selection algorithm is presented in Algorithm 2. This algorithm takes the set of all neighbour reference node **RS** of an unknown node U as input and returns the set of reliable reference nodes **SRS**. Lines 1 and 2 choose the first reliable reference node RB_1 , and lines 3-5 choose the second one RB_2 . Line 9 removes reference nodes

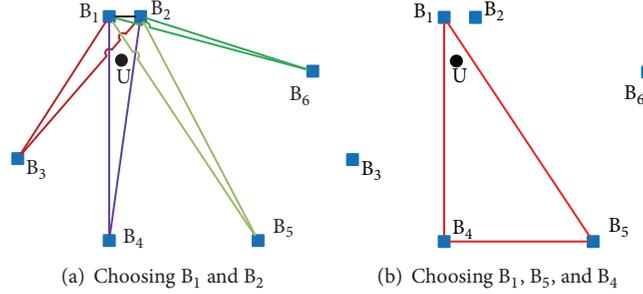


FIGURE 2: Selection of reliable reference nodes.

Input: unknown node U and its neighbour references set \mathbf{RS} , $|\mathbf{RS}| > 2$
Output: selected reference nodes set \mathbf{SRS}

- (1) Compute $conf$ for all $B_i \in \mathbf{RS}$
- (2) $RB_1 \leftarrow \arg \max_i \{conf_i \mid B_i \in \mathbf{RS}\}$
- (3) Compute $conf'$ for all $B_i \in \mathbf{RS} - \{RB_1\}$
- (4) $\mathbf{CRS} \leftarrow \arg \max_i \{conf'_i \mid B_i \in \mathbf{RS} - \{RB_1\}\}$
- (5) $RB_2 \leftarrow \arg \max_i \{Dist(RB_1, B_i) \mid B_i \in \mathbf{CRS}\}$
- (6) $\mathbf{SRS} \leftarrow \{RB_1, RB_2\}$, $\mathbf{RS} \leftarrow \mathbf{RS} - \{RB_1, RB_2\}$
- (7) **while** $|\mathbf{RS}| > 0$ **do**
- (8) $CRB \leftarrow \arg \max_i \{conf_i \mid B_i \in \mathbf{RS}\}$
- (9) **if** CRB is not collinear with any two elements of \mathbf{SRS} **then**
- (10) $dc \leftarrow$ minimal internal angle of triangles formed by CRB and any two elements in \mathbf{SRS}
- (11) **if** $dc \geq \delta$ **then**
- (12) $\mathbf{SRS} \leftarrow \mathbf{SRS} \cup \{CRB\}$
- (13) **end if**
- (14) **end if**
- (15) $\mathbf{RS} \leftarrow \mathbf{RS} - \{CRB\}$
- (16) **end while**
- (17) **return** \mathbf{SRS}

ALGORITHM 2: Reliable reference nodes selection.

being collinear with current set of reliable reference nodes \mathbf{SRS} . Line 10 computes dc . Line 12 appends reference nodes with $dc \geq \delta$ into \mathbf{SRS} . Note that \mathbf{SRS} may contain only two reference nodes according to this algorithm. If so, U is localized by Algorithm 4 with RB_1 and RB_2 as inputs.

5. Niching PSO-Based Localization Algorithm

5.1. Choice of Stop Criteria of PSO. The stop criteria of PSO determine number of iterations and consequently determine time complexity of localization algorithm. There are three types of stop criteria: a prespecified maximum number of iterations, achievement of a specified acceptable solution, and lack of change in a certain successive iteration [35]. The existing PSO-based localization methods utilize (7), which is the combination of the first two criteria.

Obviously, $\tilde{d}_{ij} = d_{ij}$ if $e = 0$, so $f_{opt} = 0$, then (7) is feasible. However, $e > 0$ in real applications, and it is difficult to give a feasible f_{opt} . If we observe the curves of $f(Gbest)$, as shown in Figure 3 taking three test cases as examples,

we can find that $f(Gbest)$ drops significantly before the first 100 iterations, and it keeps the same during the last 300 iterations. We also perform simulations with 1,200 different test cases using PSO-based localization algorithm and count the maximum number of iterations of the same continuous $Gbest$. The results are illustrated in Figure 4. There are more than 99.08% test cases keeping the same $Gbest$ during 200 continuous iterations. On the contrary, the first iteration to achieve the final $Gbest$ does not obey any rule, as shown in Figure 5.

Therefore, if t_{max} is very large, it may waste many iterations when the same $Gbest$ repeats continuously; but if t_{max} is small, it may not achieve the best fitness. We adopt the combination of above-mentioned three stop criteria instead of (7), as shown in Algorithm 3.

5.2. Niching PSO-Based Localization Algorithm. We investigate the unknown node with low localization accuracy and find that these sensor nodes are often localized at the symmetric positions of their true coordinates, which is

Input: unknown node U and the set of selected neighbour references **SRS**
Output: estimated position of $U(\hat{x}_u, \hat{y}_u)$

```

(1) Initialize the search space  $sp$ , and divide  $sp$  into  $K$  disjoint parts
(2) Evaluate the distances of  $U$  to each  $B \in \mathbf{SRS}$ 
(3) Generate a group of particles in each region
(4) Evaluate the fitness values of all particles
(5) Evaluate  $Pbest$  of all particles, and  $Sbest_j$  of particles group  $j$ 
(6)  $preSbest_j \leftarrow Sbest_j, repTimes_j \leftarrow 0, j = 1, 2, \dots, K$ 
(7)  $isContinue_j \leftarrow \mathbf{TRUE}, j = 1, 2, \dots, K$ 
(8) for  $i \leftarrow 1$  to  $t_{max}$  do
(9)   for  $j \leftarrow 1$  to  $K$  do
(10)    if  $isContinue_j$  then
(11)      Update particles of group  $j$ 
(12)      Re-evaluate the fitness of these particles
(13)      Update  $Pbest$  and  $Sbest_j$ 
(14)      if  $f(Sbest_j) < f_{opt}$  then
(15)        return  $(\hat{x}_u, \hat{y}_u) \leftarrow Sbest_j$ 
(16)      else if  $|f(Sbest_j) - f(preBest_j)| < \xi$  then
(17)         $repTimes_j \leftarrow repTimes_j + 1$ 
(18)      else
(19)         $preSbest_j \leftarrow Sbest_j, repTimes_j \leftarrow 0$ 
(20)      end if
(21)      if  $(repTimes_j > \lambda_1 \wedge f(Sbest_j) > \mu) \vee repTimes_j > \lambda_2$  then
(22)         $isContinue_j \leftarrow \mathbf{FALSE}$ 
(23)      end if
(24)    end if
(25)  end for
(26)  if  $\bigvee_{j=1}^K isContinue_j = \mathbf{FALSE}$  then
(27)    return  $(\hat{x}_u, \hat{y}_u) \leftarrow \arg \min_j \{f(Sbest_j)\}$ 
(28)  end if
(29) end for
(30) return  $(\hat{x}_u, \hat{y}_u) \leftarrow \arg \min_j \{f(Sbest_j)\}$ 

```

ALGORITHM 3: Niching PSO-based localization algorithm.

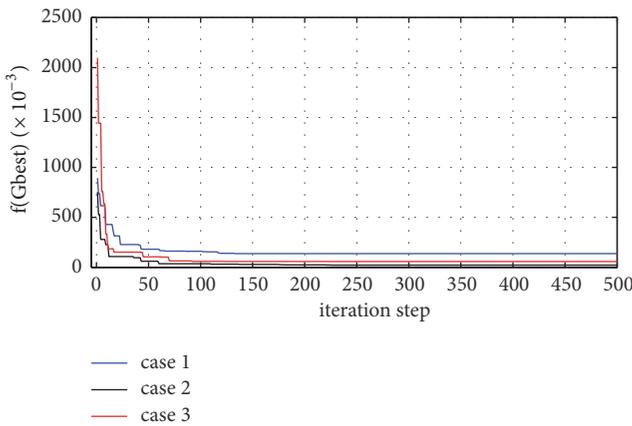


FIGURE 3: $f(Gbest)$ of each iteration.

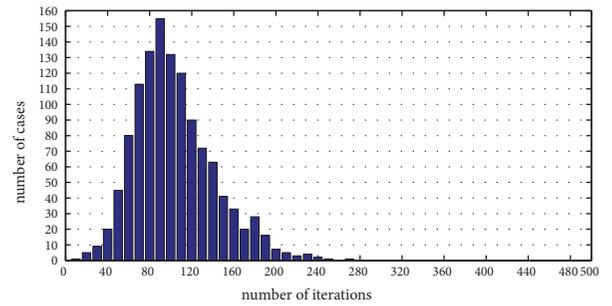


FIGURE 4: The maximum number of iterations of the same continuous $Gbest$.

referred to as *flexible ambiguity* in rigid theory [37]. Although localization algorithms based on rigid theory can cope with this problem, they need identify global-rigid patches [38, 39]. In this paper, niching technique is utilized to cope with this

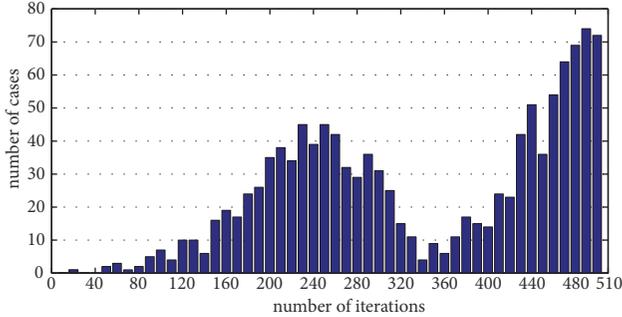
problem. Niching is an important technique for multimodal optimization, and lots of existing niching PSO algorithms require a niching parameter based on Euclidean distance or topology among particles [40, 41], where each particle still searches for the optima in entire space.

In the proposed algorithm, particles are divided into several subswarms, and each group searches for different

Input: unknown node U and its neighbour references B_1, B_2
Output: estimated position of $U(\hat{x}_u, \hat{y}_u)$ if U can be localized

- (1) $\bar{d}_i \leftarrow$ measured distances of U to $B_i (i = 1, 2)$
- (2) $(x_{bi}, y_{bi}) \leftarrow$ positions of B_i
- (3) $(x_j, y_j) \leftarrow$ roots of $\bar{d}_i^2 = (x - x_{bi})^2 + (y - y_{bi})^2, i, j = 1, 2$
- (4) **if** $(x_1, y_1) = (x_2, y_2)$ **then**
- (5) **return** $(\hat{x}_u, \hat{y}_u) \leftarrow (x_1, y_1)$
- (6) **else**
- (7) **TwoRS** \leftarrow {neighbour references of B_1 and B_2 } - $\{B_1, B_2\}$
- (8) **if** **TwoRS** = Φ **then**
- (9) **return** U cannot be localized
- (10) **end if**
- (11) $Sc_j \leftarrow \sum_k \{conf_{bk} \mid B_k \in \text{TwoRS} \wedge \text{Dist}((x_j, y_j), (x_{bk}, y_{bk})) > R\}$
- (12) **if** $Sc_1 \neq Sc_2$ **then**
- (13) **return** $(\hat{x}_u, \hat{y}_u) \leftarrow (x_j, y_j)$ **if** $Sc_j > Sc_{3-j}$
- (14) **else**
- (15) **CTwoRS** \leftarrow {common neighbour reference nodes of B_1 and B_2 } \cap **TwoRS**
- (16) **if** **CTwoRS** $\neq \Phi$ **then**
- (17) $B_3 \leftarrow \arg \max_B \{conf_B \mid B \in \text{CTwoRS}\}$
- (18) **else**
- (19) $B_3 \leftarrow \arg \max_B \{conf_B \mid B \in \text{TwoRS}\}$
- (20) **end if**
- (21) $(x_3, y_3) \leftarrow$ solution of Algorithm 3 using B_1, B_2, B_3 as inputs
- (22) **return** $(\hat{x}_u, \hat{y}_u) \leftarrow (x_j, y_j)$ **if** (x_j, y_j) is nearer to (x_3, y_3) than (x_{3-j}, y_{3-j})
- (23) **end if**
- (24) **end if**

ALGORITHM 4: Localize the unknown node with two neighbour references.

FIGURE 5: The first iteration achieving the final G_{best} .

subregions of the whole search space of PSO. The final result is the best particle of all subswarms. Because the reference nodes chosen by Algorithm 2 are noncollinear, this method can achieve a unique and best estimation. Moreover, the particles are constricted in a fixed subregion, so it does not need a niching parameter based on the calculation of Euclidean distance. The detailed localization algorithm is shown in Algorithm 3.

In Algorithm 3, the radio range of sensor nodes is represented by its circumscribed square. For an anchor, the side length of the square is R . For a localized unknown node, the side length is $2R$, because the localization error of

unknown node is less than R , according to simulation results of Section 7.4. In first, the algorithm generates search space sp of particles based on the following method (line 1):

- (i) **Case 1.** If U has at least two neighbour references and $SRS = \{B_i \mid i = 1, 2, \dots, n, n > 1\}$, sp is defined as

$$g(x) = \begin{cases} sp.left = \max_i \{0, x_{bi} - kR\} \\ sp.right = \min_i \{L, x_{bi} + kR\} \\ sp.bott = \max_i \{0, y_{bi} - kR\} \\ sp.top = \min_i \{L, y_{bi} + kR\} \end{cases} \quad (14)$$

where $(sp.left, sp.top)$ and $(sp.right, sp.bott)$ are, respectively, coordinates of the top left and bottom right corners of sp , $k = 1$ if $B_i \in \text{AS}$, otherwise $k = 2$.

- (ii) **Case 2,** If U has only one neighbour reference node B_1 , sp is the circumscribed square of the communication range of B_1 .

sp is divided into $K \geq 4$ equal and disjoint subregions further. The distance \bar{d}_i of U to B_i is evaluated in line 2. It is calculated as follows:

- (i) **Case 1.** If B_i is a neighbour reference node, $\bar{d}_i = \bar{d}_i$.

Input: unknown node U and its neighbour reference B_1
Output: estimated position of $U(\hat{x}_u, \hat{y}_u)$ if U can be localized

- (1) $\tilde{d}_1 \leftarrow$ measured distances of U to B_1
- (2) $(x_{b_1}, y_{b_1}) \leftarrow$ positions of B_1
- (3) **TwoRS** \leftarrow {neighbour references of B_1 }
- (4) **if** **TwoRS** = Φ **then**
- (5) **return** U cannot be localized
- (6) **end if**
- (7) **if** |**TwoRS**| > 1 **then**
- (8) $\{B_2, B_3\} \leftarrow \arg \max_B \{conf_B \mid B \in \text{TwoRS}\}$
- (9) **else**
- (10) $B_2 \leftarrow B \in \text{TwoRS}$
- (11) **ThrRS** \leftarrow {neighbour references of B_2 } - $\{B_1\}$
- (12) **if** |**ThrRS**| = Φ **then**
- (13) **return** U cannot be localized
- (14) **end if**
- (15) $B_3 \leftarrow \arg \max_B \{conf_B \mid B \in \text{ThrRS}\}$
- (16) **end if**
- (17) **return** $(\hat{x}_u, \hat{y}_u) \leftarrow$ solution of Algorithm 3 using B_1, B_2, B_3 as inputs

ALGORITHM 5: Localize the unknown node with one neighbour references.

- (ii) **Case 2.** If B_i is B_3 selected by line 17 in Algorithm 4, the distance is

$$\bar{d}_3 = \frac{1}{2} (\tilde{d}_1 + \tilde{d}_2 + \hat{d}_{b_{13}} + \hat{d}_{b_{23}}) \quad (15)$$

where $\tilde{d}_1(\tilde{d}_2)$ is measured distance between U and $B_1(B_2)$ and $\hat{d}_{b_{13}}(\hat{d}_{b_{23}})$ is estimated distance from $B_1(B_2)$ to B_3 .

- (iii) **Case 3.** If B_i is B_3 selected by line 19 in Algorithm 4, the distance is

$$\bar{d}_3 = \begin{cases} \tilde{d}_1 + \hat{d}_{b_{13}} & \text{if } B_3 \text{ is a neighbour node of } B_1 \\ \tilde{d}_2 + \hat{d}_{b_{23}} & \text{if } B_3 \text{ is a neighbour node of } B_2 \end{cases} \quad (16)$$

- (iv) **Case 4.** If B_i is B_2 or B_3 generated by line 8 or 10 in Algorithm 5, the distance is

$$\bar{d}_i = \tilde{d}_1 + \hat{d}_{b_{1i}}, \quad i = 2, 3 \quad (17)$$

- (v) **Case 5.** If B_i is B_3 obtained by line 15 in Algorithm 5, the distance is

$$\bar{d}_3 = \tilde{d}_1 + \hat{d}_{b_{13}} \quad (18)$$

Lines 3-7 generate a group of particles in each search region and initialize some variables. $Sbest_j$ is the local best position of particle group j , $repTimes_j$ records the number of successive iterations with same $Sbest_j$, and $isContinue_j$ decides whether particle group j continues to update or not.

Lines 8-29 are main iteration process of niching PSO. There are four thresholds:

- (i) ξ judges the difference between current and previous $Sbest_j$.
- (ii) λ_1 and λ_2 decide the maximum repeated times of the same $Sbest_j$, and $\lambda_1 < \lambda_2$.
- (iii) μ determines the degree of acceptance of $Sbest_j$. For particle group j , if an unacceptable $f(Sbest_j) > \mu$ repeats more than λ_1 times, or a nearly acceptable $f(Sbest_j) \leq \mu$ repeats more than λ_2 times, the particles of this group will stop evolution.

This algorithm returns results under one of the following conditions:

- (i) It reaches the maximum number of iterations t_{max} (line 8).
- (ii) One group of particles achieves an acceptable fitness f_{opt} (line 15).
- (iii) All groups of particles stop updating (line 26).

6. Localization of Unknown Nodes of Mode 3

The unknown node belonging to Mode 3 is localized in a way similar to DV-Distance [12] in [21], which utilizes multihop anchors to avoid error accumulation. However, if the multihop anchor is far from the unknown node to be localized, the estimated distance between them is highly unreliable [42]. Therefore, the reference nodes chosen to localize an unknown node U should be as near as possible.

6.1. Localization of Unknown Node with Two Neighbour References. Unknown node U with only two neighbour reference nodes B_1 and B_2 is localized by Algorithm 4. After getting positions and measured distances of these two references (lines 1-2), it solves equations in line 3. If there is only one solution, line 5 returns it as a result. Otherwise, line 7 decides its position by two-hop neighbour references **TwoRS**. U cannot be localized if **TwoRS** = Φ (line 9). Otherwise, line 11 computes distances between each solutions $(x_j, y_j)(j = 1, 2)$ and each $B_k \in \mathbf{TwoRS}$ by function $Dist()$ and calculates two sums Sc_j , which is the sum of $conf$ (see (9)) of all B_k whose distance to (x_j, y_j) is larger than R . Line 13 returns the solution with the largest Sc as a result. If these two sums are the same, line 15 obtains the common neighbours **CTwoRS** of B_1 and B_2 . If **CTwoRS** $\neq \Phi$, line 17 chooses $B_3 \in \mathbf{CTwoRS}$ with the largest $conf$; otherwise, line 19 chooses $B_3 \in \mathbf{TwoRS}$ with the largest $conf$. Next, line 21 estimates a position (x_3, y_3) using Algorithm 3, and line 22 returns the coordinate which is nearer to (x_3, y_3) than the other as a result.

6.2. Localization of Unknown Node with One Neighbour Reference. If U has only one neighbour reference B_1 , it is localized by Algorithm 5. The first two lines are similar to those of Algorithm 4. Line 3 obtains two-hop reference nodes **TwoRS**. If **TwoRS** has at least two members, line 8 selects two of them B_2 and B_3 which have the largest $conf$. If **TwoRS** has only one element, line 10 chooses this element as B_2 . Line 11 obtains three-hop neighbour reference nodes **ThrRS**, and line 15 selects $B_3 \in \mathbf{ThrRS}$ of the largest $conf$ as B_3 . Finally, line 17 computes position of U by Algorithm 3 with B_1, B_2, B_3 as inputs. If **TwoRS** or **ThrRS** is empty, U cannot be localized (line 5 and 13).

If an unknown node with less than three neighbour reference nodes cannot be localized by Algorithms 4 and 5, it is abandoned localization because its estimated location is highly unreliable if it utilizes reference nodes in more than three hops to localize itself [38].

7. Performance Analysis and Comparisons

Because our algorithm is range-based, anchor-based, and distributed, we compare it with the algorithms of same type, which are iterative multilateration, DV-Distance [9], SPSO-based localization [21, 22], HPSO [23], and hybrid PSO [24].

7.1. Complexity Analysis. We analyse the time complexity of algorithms in first. Suppose all the input data have the same number of binary digits, then the time complexity can be calculated in terms of times of basic arithmetic operations in an algorithm, such as addition, subtraction, multiplication, and division.

Both multilateration and DV-Distance use standard least-square approach to estimate position of an unknown node. Based on (1), the estimation is

$$(\hat{x}_{ui}, \hat{y}_{ui}) = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{B} \quad (19)$$

Input: two matrices $\mathbf{A}_{p \times q}$ and $\mathbf{B}_{q \times r}$
Output: $\mathbf{C} = \mathbf{A} \times \mathbf{B}$

```

(1) for  $i \leftarrow 1$  to  $p$ 
(2)   for  $j \leftarrow 1$  to  $r$ 
(3)      $C_{ij} \leftarrow 0$ 
(4)     for  $k \leftarrow 1$  to  $q$ 
(5)        $C_{ij} \leftarrow C_{ij} + A_{ik} \times B_{kj}$ 
(6)     end for
(7)   end for
(8) end for

```

ALGORITHM 6: Matrix multiplication.

where

$$\mathbf{A} = -2 \begin{pmatrix} x_{b1} - x_{bn} & y_{b1} - y_{bn} \\ \vdots & \vdots \\ x_{b,n-1} - x_{bn} & x_{b,n-1} - x_{bn} \end{pmatrix} \quad (20)$$

$$\mathbf{B} = \begin{pmatrix} \bar{d}_{1j}^2 - \bar{d}_{nj}^2 - x_{b1}^2 + x_{bn}^2 - y_{b1}^2 + y_{bn}^2 \\ \vdots \\ \bar{d}_{n-1,j}^2 - \bar{d}_{nj}^2 - x_{b,n-1}^2 + x_{bn}^2 - y_{b,n-1}^2 + y_{bn}^2 \end{pmatrix}$$

In (20), matrix \mathbf{A} needs $2(n-1)$ subtractions and $2(n-1)$ multiplications; matrix \mathbf{B} needs $5(n-1)$ multiplications, $3(n-1)$ subtractions, and $2(n-1)$ additions.

Equation (19) needs 3 matrix multiplications and 1 matrix inversion. If Algorithm 6 is applied to calculate matrix multiplication, it needs pqr additions and pqr multiplications. Further, matrices \mathbf{A} and \mathbf{B} are, respectively, $(n-1) \times 2$ and $(n-1) \times 1$, so $\mathbf{C} = \mathbf{A}^T \mathbf{A}$ needs $4(n-1)^2$ additions and $4(n-1)^2$ multiplications, $\mathbf{D} = \mathbf{C} \mathbf{A}^T$ needs $8(n-1)$ additions and $8(n-1)$ multiplications, and $\mathbf{E} = \mathbf{D} \mathbf{B}$ needs $2(n-1)^2$ additions and $2(n-1)^2$ multiplications. Suppose $\mathbf{C} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ and $ad - bc \neq 0$, $\mathbf{C}^{-1} = (\mathbf{A}^T \mathbf{A})^{-1} = (1/(ad - bc)) \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$, so it needs 6 multiplications, 1 division, and 1 subtraction.

In total, (19) and (20) need $6(n-1)^2 + 15(n-1) + 6 = 6n^2 + 3n - 3$ multiplications, 1 division, $6(n-1)^2 + 10(n-1) = 6n^2 - 2n - 4$ additions, and $5(n-1) + 1 = 5n - 4$ subtractions. Hence, the time complexity of multilateration and DV-Distance is $O(n^2)$.

The time complexities of PSO algorithms stem from the evaluation of fitness and update of particles. For each iteration, HPSO updates each particle with the local best fitness, global best fitness, and previous individual best fitness, so its complexity is higher than SPSO. Reference [24] has already showed that the complexity of each iteration of hybrid PSO is higher than SPSO. Based on (3), (5), and (8), CPSO and SPSO have the same time complexity in each iteration.

Let m be the number of particles in CPSO. During each iteration, CPSO uses (3) to calculate the particles' fitness, which needs $n + 2$ divisions and $n - 1$ additions to get w_j ; and mn extractions of square root, $4mn$ multiplications,

TABLE 2: Parameter selection of PSO-based localization algorithm.

parameter	candidate scope	best selection	
		SPSO	CPSO
V_{max}	0.1R to 2R at step of 0.1R	0.1R	-
Swarm Size	5 to 100 at step of 5	30	10
ω_{max}	1.4 to 0.9 at step of -0.1	0.9	-
ω_{min}	0.4 to 0 at step of -0.1	0	-
c_1, c_2	1.5 to 2.5 at step of 0.05	1.7/1.8	2.45/2.5

$3mn$ subtractions, and $2mn - m$ additions to get $f(x, y)$. Equation (8) needs $5m$ multiplications, $3m$ additions, and $2m$ subtractions to update particles. During each iteration of Algorithm 3, it needs 5 comparisons in lines 14, 16, and 21, and it also needs $2m - K$ comparisons to get $Pbest$ of each particle and $Sbest$ of each group of particles (line 13). It also needs n comparisons to get sp . Let Itr be the number of iterations, Algorithm 3 needs $(n + 2)Itr$ divisions, $(4mn + 5m)Itr$ multiplications, $(2mn + 2m + n - 1)Itr$ additions, $(3mn + 2m)Itr$ subtractions, $(mn)Itr$ extractions of square root, and $(2m - K + 5)Itr + n$ comparisons. Hence, the time complexity of Algorithm 3 is $O(mn \times Itr)$. Generally, $m, n \ll Itr$, so it is important to decrease Itr .

For multilateration and Algorithm 3, n is the number of neighbouring reference nodes, so $n^2 < mn \times Itr$. DV-Distance uses all anchors to localize each unknown node, so its time complexity $O(M^2)$. Generally, $M < m \times Itr$. Therefore, the time complexity sequence is multilateration $<$ DV-Distance $<$ CPSO = SPSO $<$ (hybrid PSO, HPSO).

As for communication complexity, multilateration requires each sensor node to broadcast one package, including its positions and information to assist the receiver to measure distance, to its neighbour nodes, so its communication cost is $O(N)$. DV-Distance needs three message flooding stages in whole network, so its communication complexity is $O(N^3)$. The communication cost of existing PSO-based localization algorithm is the same as multilateration. The proposed localization algorithm needs transmit a few more packages than multilateration, because our algorithm requires unknown nodes localized by Algorithms 4 and 5 to obtain information of two- and three-hop neighbour reference nodes.

7.2. Parameter Selection of PSO Algorithm. The performance of PSO-based localization algorithms is greatly impacted by the PSO's parameters. Given the possible scopes of these parameters, a comprehensive performance comparison under each combination of parameters is presented in [43]. Table 2 presents the best parameters achieving the optimal performance in [43].

7.3. Choice of δ in Algorithm 2. In Algorithm 2, δ decides the degree of collinearity of reference nodes, so we perform simulations to choose a feasible value. As shown in Figure 6, we deploy 3 reference nodes on a circle of radius R , and an unknown node U at the centre of the circle. B_1 and B_2 are fixed at $(-R\cos(\pi/3), \pm R\cos(\pi/3))$, and B_3 changes its position in

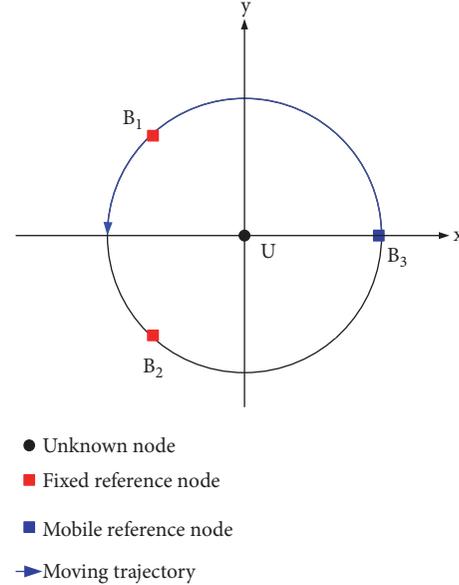


FIGURE 6: Positions of reference nodes to analyse the collinearity.

$\theta \in [0, \pi]$ along the blue arc, where θ is the angle between line $\overline{UB_3}$ and positive direction of axis x . The other setups are the same as Section 7.2. The results are shown in Figure 7. We can see that LE increases as θ increases from 0 to $2\pi/3$, and it decreases as θ decreases from $2\pi/3$ to π . Moreover, different e generates different varying trends. In the proposed algorithm, we utilize $\delta = (100e + 5)\pi/180$ as the threshold to decide the collinearity of anchors.

7.4. Comparison of Different Localization Algorithms. The simulations are performed on MATLAB 2012a, and PSO is implemented in C language. The configurations of WSNs are as follows: N is 2000 to 3000 at step of 200, M/N is 0.1 to 0.4 at step of 0.05, DOI is 0 to 0.2 at step of 0.05, R is 100 to 200 at step of 20, and e is 0 to 0.2 at step of 0.05. Both anchors and unknown nodes are deployed in the given region randomly.

The parameters of SPSO and CPSO are chosen based on Table 2. According to corresponding references, parameters of HPSO are $\omega = 0.7$, $c_1 = c_2 = 1.494$; parameters of hybrid PSO are $\omega = 1$, $c_1 = c_2 = 2.0$. CPSO has $K = 4$ groups of particles and each group has 10 particles. For Algorithms 2 and 3, $\alpha = \beta = \gamma = 1$ to give the same weight for each part of $conf$ (see (9)). $\delta = (100e + 5)\pi/180$ according to Section 7.3. $\lambda_1 = 200$, $\lambda_2 = 300$ according to the analysis in Section 5.1.

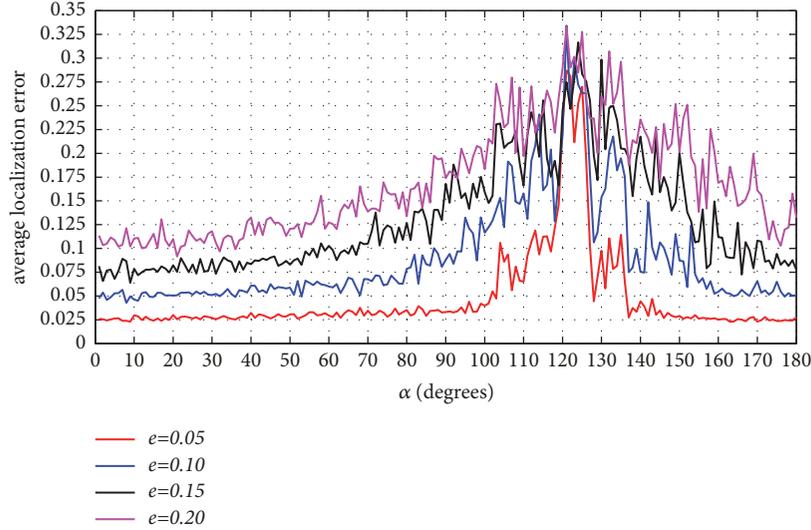


FIGURE 7: Average localization error of different angle.

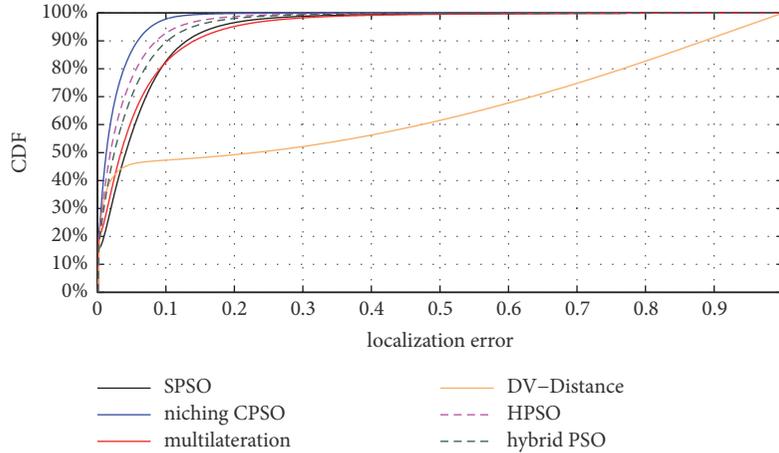


FIGURE 8: Localization error of four localization algorithms.

$\xi = 10^{-4}$, $\mu = 1$ to make the algorithm iterate as many times as possible. Algorithm 3 utilizes niching CPSO to localize an unknown node.

The average and standard deviation of LE are, respectively, denoted by \overline{LE} and σ_{LE} ; and those of Itr are, respectively, denoted by \overline{Itr} and σ_{Itr} . Moreover, the number of unknown nodes with $LE > 1$ is denoted by NFL (NFL represents “number of fail localized unknown nodes”), and the LE and Itr of these nodes are excluded when analysing LE and Itr . Simulation results are shown in Table 3, and we can see that

- (i) \overline{LE} , σ_{LE} , and NFL of DV-Distance are the largest among all localization methods. DV-Distance utilizes all anchors of a WSN to localize each unknown node, and some anchors may be collinear, so matrix \mathbf{A} of (19) may be singular, which leads to high localization error.
- (ii) \overline{LE} and σ_{LE} of SPSO and multilateration are almost the same, but SPSO localizes more unknown nodes than multilateration.

(iii) Both HPSO and hybrid PSO outperform SPSO in terms of LE and Itr , but HPSO needs more iterations than hybrid PSO. On the other hand, NFL of HPSO and hybrid PSO is larger than SPSO. Further, \overline{LE} and NFL of hybrid PSO are larger than HPSO, because the replacement strategy of hybrid PSO leads to premature convergence. However, hybrid PSO uses less iterations than HPSO and SPSO.

(iv) Niching CPSO performs better than the other algorithms in terms of \overline{LE} , σ_{LE} , and \overline{Itr} . In particular, \overline{LE} of niching CPSO is 71.75%, 65.98%, 38.24%, 38.04%, and 4.19% of that of hybrid PSO, HPSO, SPSO, multilateration, and DV-Distance, respectively. Moreover, only seldom unknown nodes cannot be localized by niching CPSO.

We further compare cumulative distribution functions (CDF) of LE and Itr of localization algorithms, as shown in Figures 8 and 9, from which we can draw the following conclusions:

TABLE 3: Comparison of four localization algorithms.

Algorithm	LE	σ_{LE}	\overline{Itr}	σ_{Itr}	NFL
SPSO [21, 22]	0.0581	0.0542	475.8757	13.6372	8.1257
Niching CPSO	0.0221	0.0189	208.3633	33.6333	0.2206
HPSO [23]	0.0335	0.0511	281.6785	21.5868	17.6117
hybrid PSO [24]	0.0408	0.0510	223.8209	14.1587	20.4287
multilateration	0.0578	0.0530	-	-	21.5019
DV-Distance	0.5272	0.1912	-	-	90.1234

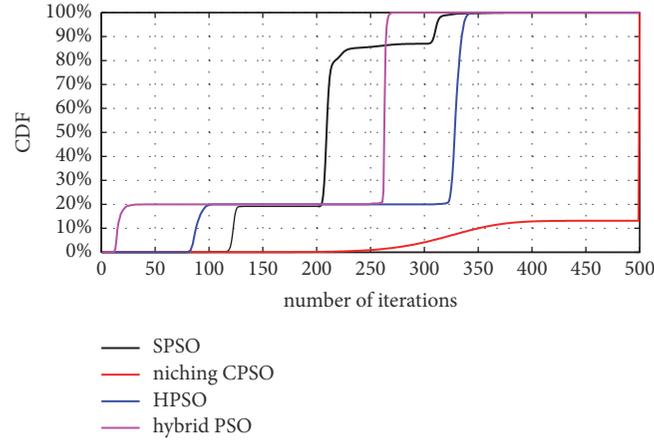


FIGURE 9: Number of iterations of PSO-based localization algorithms.

- (i) LE of DV-Distance is the worst. Its LE distributes in 0.01-1 uniformly, and only 33.4% unknown nodes have $LE < 0.01$.
- (ii) For niching CPSO localization algorithm, LE of more than 90% unknown nodes are less than 0.06, which is the best among all localization algorithms. Moreover, its largest Itr is 416, and Itr of more than 87% unknown nodes is less than 300. Therefore, niching CPSO achieves the highest localization precision with the smallest number of iterations among all algorithms.
- (iii) There are about 63.4% and 67.4% unknown nodes whose $LE \leq 0.06$ using SPSO and multilateration. Although their CDFs are almost the same, the number of unknown nodes with $LE > 0.01$ of SPSO is less than multilateration, so \overline{LE} of multilateration is larger than SPSO.
- (iv) LE of HPSO and hybrid PSO are between SPSO and niching CPSO, but their Itr are smaller than SPSO.

7.5. Impacts of Different Setups on Niching PSO-Based Localization. Based on the best parameters of SPSO and CPSO presented in Section 7.2, we compare the performance of Algorithm 3 using SPSO and CPSO with different network setups.

7.5.1. Impacts of N and M . Generally, the more anchors in a WSN obtain the more accurate localization. Figure 10

shows that niching PSO-based methods obey this rule, but the impacts of N and M on niching CPSO are less than those on SPSO, because the variation of niching SPSO is more obvious than niching CPSO. This figure also denotes that LE of niching SPSO is larger than niching CPSO. \overline{LE} of niching CPSO and niching SPSO, respectively, belong to 0.0196-0.0280 and 0.0488-0.0772, and their σ_{LE} , respectively, belong to 0.0152-0.0280 and 0.0424-0.0806.

The impacts of N and M on Itr of both niching PSO methods are little, as shown in Figure 11. \overline{Itr} of niching SPSO is 469.68-485.33, and that of niching CPSO is 206.96-210.98. On the contrary, σ_{Itr} of niching SPSO is 11.99-15.22, and that of niching CPSO is 132.52-137.24, so the varying range of Itr of niching CPSO is larger than niching SPSO.

7.5.2. Impacts of DOI and e . Figure 12 shows that DOI has little impacts on LE , while LE becomes larger as e increases. When $e = 0$, \overline{LE} of niching CPSO and niching SPSO is, respectively, 0.001 and 0.0131. When $e > 0$, \overline{LE} of niching CPSO is about 26.9%-45.2% of niching SPSO, and σ_{LE} of niching CPSO is about 25.2%-44.1% of niching SPSO.

Figure 13 shows that Itr is almost the same under different DOI and same e . Both niching CPSO and niching SPSO have the minimal Itr under $e = 0$, which is, respectively, 133.33 and 379.38. However, niching CPSO and niching SPSO exhibit different variation when $e > 0$. \overline{Itr} of niching SPSO is 500 for all $e > 0$. \overline{Itr} of niching CPSO is 254.13 when $e = 0.05$, which is the largest, and it is, respectively, 221.93, 216.43, and 216

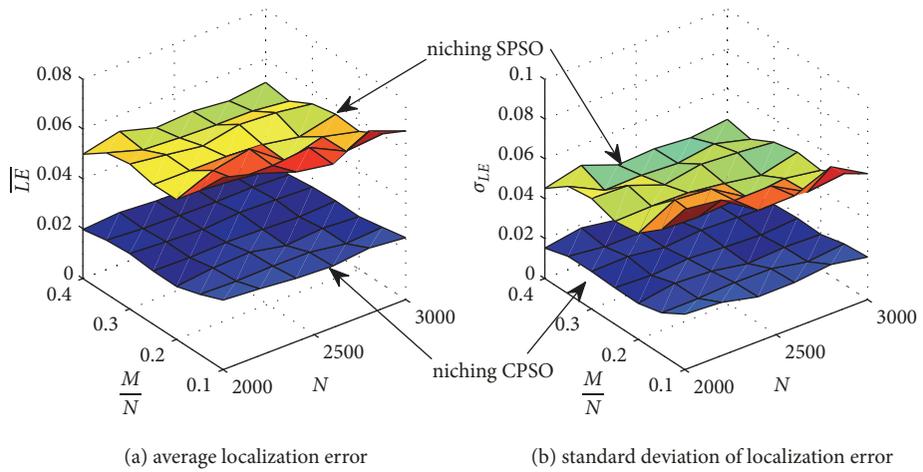


FIGURE 10: Impacts of N and M on LE of niching PSO-based methods.

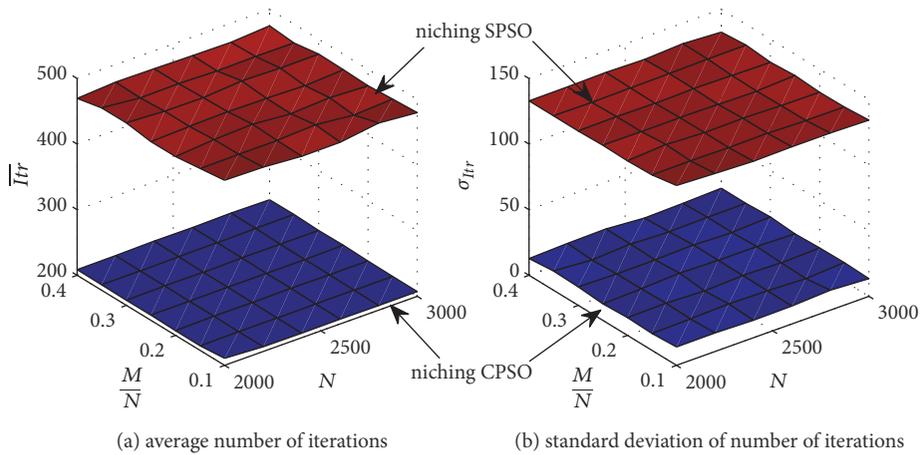


FIGURE 11: Impacts of N and M on Itr of niching PSO-based methods.

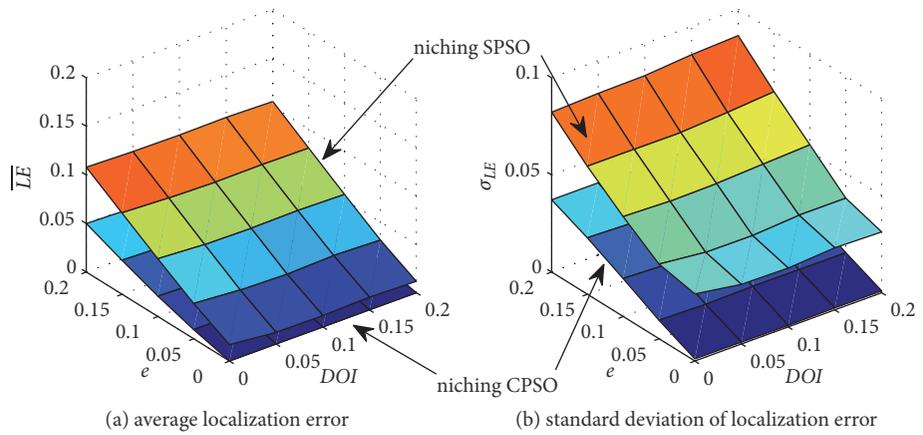


FIGURE 12: Impacts of DOI and e on LE of niching PSO-based methods.

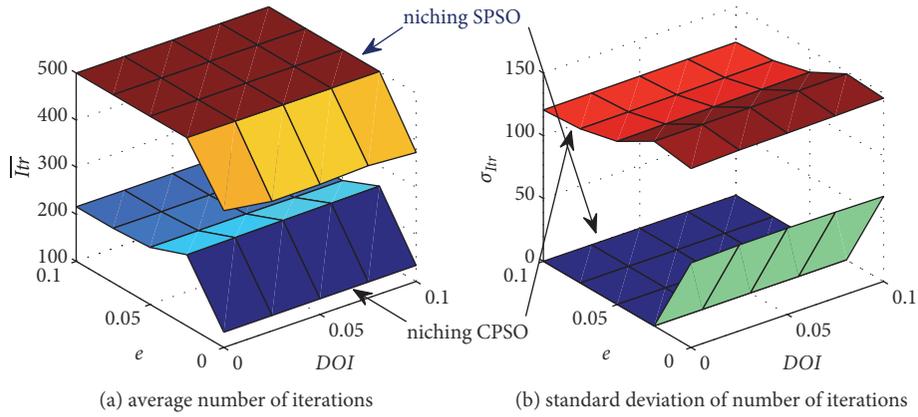


FIGURE 13: Impacts of DOI and e on Itr of niching PSO-based methods.

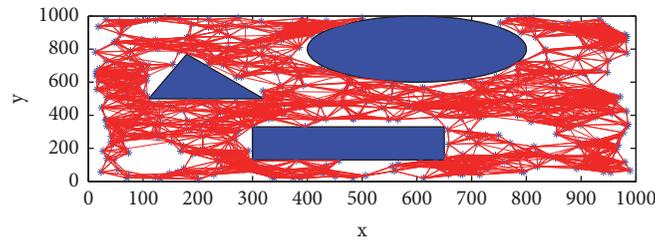


FIGURE 14: A WSN with coverage holes. The blue blocks, stars (“*”), and red lines are, respectively, coverage holes, sensor nodes, and connectivity relations.

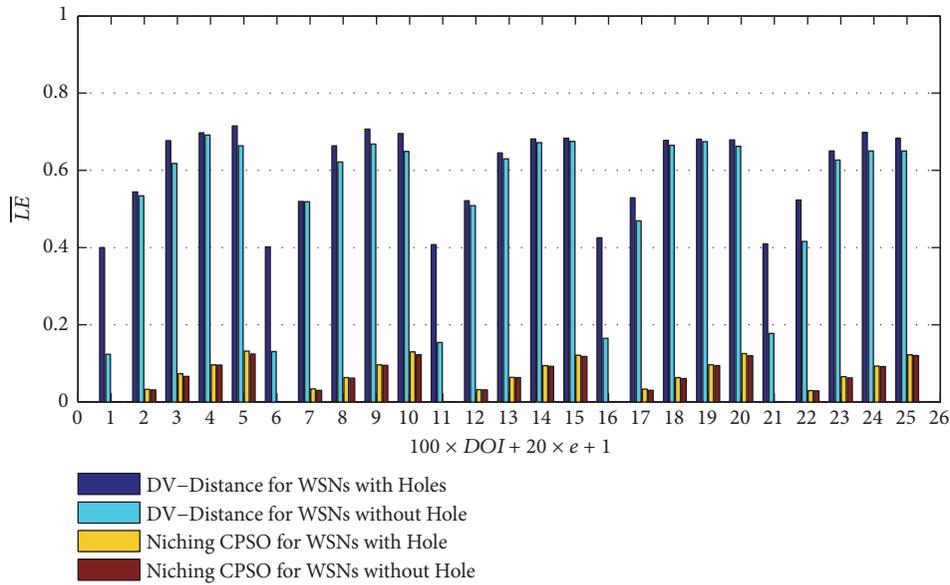


FIGURE 15: Impacts of coverage hole on \overline{LE} .

when $e = 0.1, 0.15,$ and 0.2 . On the contrary, σ_{Itr} of niching SPSO is about 68.63 when $e = 0$, and it is 0 when $e > 0$. σ_{Itr} of niching CPSO is 120.69-148.08, and it is the largest when $e = 0.05$.

7.5.3. Impacts of Coverage Holes. We compare the impacts of coverage holes of WSNs on niching CPSO-based localization

algorithm and DV-Distance, because the former is the best one among all PSO-based localization algorithms according to the above simulations, and the latter relies on connectivity information among sensor nodes.

Figure 14 illustrates the simulation region with three coverage holes, and the other setups as the same as above. Figures 15 and 16 show that the impacts of coverage holes

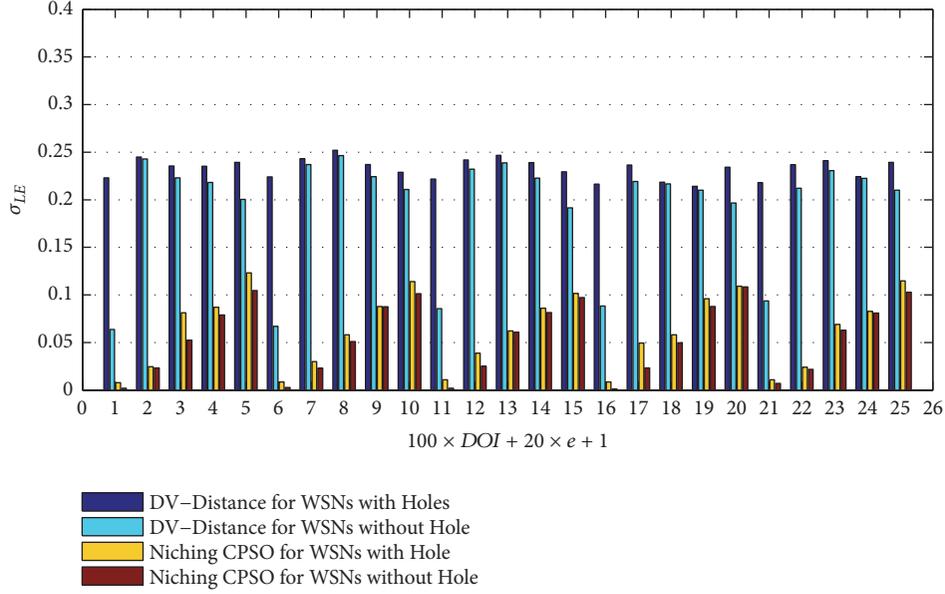


FIGURE 16: Impacts of coverage hole on σ_{LE} .

on DV-Distance are more obvious than those on niching CPSO. The difference of \overline{LE} (σ_{LE}) between WSNs with holes and without hole is located in $[0, 0.2763]$ ($[0.0018, 0.1593]$) using DV-Distance, while that of niching CPSO is located in $[0, 0.0074]$ ($[0, 0.0286]$). Moreover, all the above differences become larger and larger with the increase of e and DOI .

8. Conclusions

Due to random deployment of large-scale WSNs, anchors are not evenly distributed, so some unknown nodes may have not enough neighbouring anchors to localize itself, while some others may have overmuch neighbouring anchors to give a great burden to localize. Therefore, this paper presents a distributed, range-based, anchor-based localization algorithm using niching PSO with constriction coefficient and reliable references selection. This algorithm consists of three phases:

- (i) Reliable references selection based on distance to unknown node, angle between reference nodes, and localization precision of reference nodes.
- (ii) Generation and division of search space.
- (iii) Position estimation using niching PSO.

This algorithm uses a new stop criterion to decrease number of iterations, and a comprehensive method to choose the most reliable references to improve localization accuracy. It also presents localization algorithm for the unknown nodes with less than three neighbour reference nodes. Theoretical analysis shows that the proposed method has low complexity compared with the other PSO-based localization algorithms. Simulation results show that niching CPSO utilizes less iteration to achieve higher localization precision, and it is impacted less by number of sensor nodes, number of anchors, and degree of irregularity compared to the other algorithms.

Nowadays, the large-scale WSNs are applied to three-dimensional (3D) region more and more, and new PSO variants such as Enhanced leader PSO (ELPSO) [44] are developed recently, so we will study the application of recent PSO variants in 3D WSNs in future.

Data Availability

Data are obtained from real experiments.

Conflicts of Interest

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

Acknowledgments

This article is partly supported by National Key Research and Development Plan of China (Grants nos. 2017YFC0804406 and 2018YFC1406203). This paper is also supported by Shandong Education Department Grant J16LN73.

References

- [1] L. Cheng, C. Wu, Y. Zhang et al., "A survey of localization in wireless sensor network," *International Journal of Distributed Sensor Networks*, vol. 8, no. 12, Article ID 962523, 2012.
- [2] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, pp. 13–32, 2009.
- [3] Y. Liu, X. Mao, Y. He, K. Liu, W. Gong, and J. Wang, "Citysee: not only a wireless sensor network," *IEEE Network*, vol. 27, no. 5, pp. 42–47, 2013.
- [4] Y. Qu, W. Han, L. Fu et al., "LAINet - A wireless sensor network for coniferous forest leaf area index measurement:

- Design, algorithm and validation,” *Computers and Electronics in Agriculture*, vol. 108, pp. 200–208, 2014.
- [5] M. Femminella and G. Reali, “A zero-configuration tracking system for first responders networks,” *IEEE Systems Journal*, vol. 11, no. 4, pp. 2917–2928, 2017.
 - [6] M. Femminella and G. Reali, “Low satellite visibility areas: Extension of the GPS capabilities to deploy location-based services,” *IEEE Vehicular Technology Magazine*, vol. 7, no. 4, pp. 55–65, 2012.
 - [7] N. Bulusu, J. Heidemann, and D. Estrin, “GPS-less low-cost outdoor localization for very small devices,” *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, 2000.
 - [8] J. Blumenthal, R. Grossmann, F. Golasowski, and D. Timmermann, “Weighted centroid localization in Zigbee-based sensor networks,” in *Proceedings of the IEEE International Symposium on Intelligent Signal Processing (WISP ’07)*, pp. 1–6, October 2007.
 - [9] Y. Wen, Z. Li, Y. Meng et al., “A two-stage range-free localization method for wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 908417, 12 pages, 2015.
 - [10] V. K. Chaurasiya, N. Jain, and G. C. Nandi, “A novel distance estimation approach for 3D localization in wireless sensor network using multi dimensional scaling,” *Information Fusion*, vol. 15, pp. 5–18, 2014.
 - [11] J. Fan, Y. Hu, T. H. Luan, and M. Dong, “DisLoc: A Convex Partitioning Based Approach for Distributed 3-D Localization in Wireless Sensor Networks,” *IEEE Sensors Journal*, vol. 17, no. 24, pp. 8412–8423, 2017.
 - [12] D. Niculescu and B. Nath, “DV based positioning in ad hoc networks,” *Telecommunication Systems*, vol. 22, no. 1, pp. 267–280, 2003.
 - [13] L. Gui, T. Val, A. Wei, and R. Dalce, “Improvement of range-free localization technology by a novel DV-hop protocol in wireless sensor networks,” *Ad Hoc Networks*, vol. 24, pp. 55–73, 2015.
 - [14] W. Zhao, S. Su, and F. Shao, “Improved DV-Hop algorithm using locally weighted linear regression in anisotropic wireless sensor networks,” *Wireless Personal Communications*, vol. 98, no. 4, pp. 3335–3353, 2018.
 - [15] G. Tan, H. Jiang, S. Zhang, Z. Yin, and A.-M. Kermarrec, “Connectivity-based and anchor-free localization in large-scale 2D/3D sensor networks,” *ACM Transactions on Sensor Networks*, vol. 10, no. 1, article no. 6, 21 pages, 2013.
 - [16] S. Mazuelas, A. Bahillo, R. M. Lorenzo et al., “Robust indoor positioning provided by real-time rssi values in unmodified WLAN networks,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 5, pp. 821–831, 2009.
 - [17] A. Simonetto and G. Leus, “Distributed maximum likelihood for simultaneous self-localization and tracking in sensor networks,” *IEEE Transactions on Signal Processing*, vol. 62, no. 10, pp. 1424–1437, 2014.
 - [18] X. Wu, G. Wang, D. Dai, and M. Tong, “Accurate acoustic energy-based localization with beacon position uncertainty in wireless sensor networks,” *Journal of Network and Computer Applications*, vol. 43, pp. 76–83, 2014.
 - [19] G. Naddafzadeh-Shirazi, M. B. Shenouda, and L. Lampe, “Second order cone programming for sensor network localization with anchor position uncertainty,” *IEEE Transactions on Wireless Communications*, vol. 13, no. 2, pp. 749–763, 2014.
 - [20] R. V. Kulkarni, A. Förster, and G. K. Venayagamoorthy, “Computational intelligence in wireless sensor networks: a survey,” *IEEE Communications Surveys & Tutorials*, vol. 13, no. 1, pp. 68–96, 2011.
 - [21] P.-J. Chuang and Y.-J. Jiang, “Effective neural network-based node localisation scheme for wireless sensor networks,” *IET Wireless Sensor Systems*, vol. 4, no. 2, pp. 97–103, 2014.
 - [22] T. Najeh, H. Sassi, and N. Liouane, “A Novel Range Free Localization Algorithm in Wireless Sensor Networks Based on Connectivity and Genetic Algorithms,” *International Journal of Wireless Information Networks*, vol. 25, no. 1, pp. 88–97, 2018.
 - [23] S. Phoemphon, C. So-In, and T. G. Nguyen, “An enhanced wireless sensor network localization scheme for radio irregularity models using hybrid fuzzy deep extreme learning machines,” *Wireless Networks*, vol. 24, no. 3, pp. 799–819, 2018.
 - [24] R. V. Kulkarni and G. K. Venayagamoorthy, “Bio-inspired algorithms for autonomous deployment and localization of sensor nodes,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40, no. 6, pp. 663–675, 2010.
 - [25] P.-J. Chuang and C.-P. Wu, “Employing PSO to enhance RSS range-based node localization for wireless sensor networks,” *Journal of Information Science and Engineering*, vol. 27, no. 5, pp. 1597–1611, 2011.
 - [26] H. Guo, K.-S. Low, and H.-A. Nguyen, “Optimizing the localization of a wireless sensor network in real time based on a low-cost microcontroller,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 3, pp. 741–749, 2011.
 - [27] A. Kumar, A. Khosla, J. S. Saini, and S. Singh, “Computational intelligence based algorithm for node localization in Wireless Sensor Networks,” in *Proceedings of the 6th IEEE International Conference on Intelligent Systems (IS ’12)*, pp. 431–438, September 2012.
 - [28] S. Monica and G. Ferrari, “Swarm intelligent approaches to auto-localization of nodes in static UWB networks,” *Applied Soft Computing*, vol. 25, pp. 426–434, 2014.
 - [29] S. Phoemphon, C. So-In, and D. Niyato, “A hybrid model using fuzzy logic and an extreme learning machine with vector particle swarm optimization for wireless sensor network localization,” *Applied Soft Computing*, vol. 65, pp. 101–120, 2018.
 - [30] S. P. Singh and S. C. Sharma, “A PSO Based Improved Localization Algorithm for Wireless Sensor Network,” *Wireless Personal Communications*, vol. 98, no. 1, pp. 487–503, 2018.
 - [31] H. Woo and C. Lee, “Geometric Range-Free Localization Algorithm Based on Optimal Anchor Node Selection in Wireless Sensor Networks,” *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 509892, 10 pages, 2014.
 - [32] S. Lee, B. Koo, and S. Kim, “RAPS: reliable anchor pair selection for range-free localization in anisotropic networks,” *IEEE Communications Letters*, vol. 18, no. 8, pp. 1403–1406, 2014.
 - [33] Z. Zhou, J.-H. Cui, and S. Zhou, “Efficient localization for large-scale underwater sensor networks,” *Ad Hoc Networks*, vol. 8, no. 3, pp. 267–279, 2010.
 - [34] H. Bao, B. Zhang, C. Li, and Z. Yao, “Mobile anchor assisted particle swarm optimization (PSO) based localization algorithms for wireless sensor networks,” *Wireless Communications and Mobile Computing*, vol. 12, no. 15, pp. 1313–1325, 2012.
 - [35] A. R. Jordehi and J. Jasni, “Parameter selection in particle swarm optimisation: a survey,” *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 25, no. 4, pp. 527–542, 2013.
 - [36] M. Clerc and J. Kennedy, “The particle swarm-explosion, stability, and convergence in a multidimensional complex space,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

- [37] Y. Zhang, S. Liu, X. Zhao, and Z. Jia, "Theoretic analysis of unique localization for wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 3, pp. 623–634, 2012.
- [38] L. Zhang, L. Liu, C. Gotsman, and S. J. Gortler, "An as-rigid-as-possible approach to sensor network localization," *ACM Transactions on Sensor Networks*, vol. 6, no. 4, pp. 1–21, 2010.
- [39] X. Wang, J. Luo, Y. Liu, S. Li, and D. Dong, "Component-based localization in sparse wireless networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 2, pp. 540–548, 2011.
- [40] X. Li, "Niching without niching parameters: particle swarm optimization using a ring topology," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 1, pp. 150–169, 2010.
- [41] B. Y. Qu, J. J. Liang, and P. N. Suganthan, "Niching particle swarm optimization with local search for multi-modal optimization," *Information Sciences*, vol. 197, pp. 131–143, 2012.
- [42] B. Huang, C. Yu, and B. D. O. Anderson, "Understanding error propagation in multihop sensor network localization," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 12, pp. 5811–5819, 2013.
- [43] H. Cui, M. Shu, M. Song, and Y. Wang, "Parameter Selection and Performance Comparison of Particle Swarm Optimization in Sensor Networks Localization," *Sensors*, vol. 17, no. 3, article No. 487, 2017.
- [44] A. R. Jordehi, "Enhanced leader PSO (ELPSO): a new PSO variant for solving global optimisation problems," *Applied Soft Computing*, vol. 26, pp. 401–417, 2015.

