



**Michigan
Technological
University**

Michigan Technological University
Digital Commons @ Michigan Tech

Michigan Tech Publications

3-1-2017

Parameter selection and performance comparison of particle swarm optimization in sensor networks localization

Huanqing Cui
Shandong University of Science and Technology

Minglei Shu
Shandong Computer Science Center

Min Song
Michigan Technological University

Yinglong Wang
Shandong Computer Science Center

Follow this and additional works at: <https://digitalcommons.mtu.edu/michigantech-p>



Part of the [Computer Sciences Commons](#)

Recommended Citation

Cui, H., Shu, M., Song, M., & Wang, Y. (2017). Parameter selection and performance comparison of particle swarm optimization in sensor networks localization. *Sensors*, 17(3). <http://dx.doi.org/10.3390/s17030487>

Retrieved from: <https://digitalcommons.mtu.edu/michigantech-p/735>

Follow this and additional works at: <https://digitalcommons.mtu.edu/michigantech-p>



Part of the [Computer Sciences Commons](#)

Article

Parameter Selection and Performance Comparison of Particle Swarm Optimization in Sensor Networks Localization

Huanqing Cui ^{1,2,*}, Minglei Shu ², Min Song ³ and Yinglong Wang ²

¹ Shandong Province Key Laboratory of Wisdom Mine Information Technology, Shandong University of Science and Technology, Qingdao 266590, China

² Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputer Center in Jinan), Jinan 250101, China; shuml@sdas.org (M.S.); wangyl@sdas.org (Y.W.)

³ Computer Science Department, Michigan Technological University, Houghton, MI 49931, USA; mins@mtu.edu

* Correspondence: smart0193@163.com

Academic Editor: Mohamed F. Younis

Received: 24 December 2016; Accepted: 28 February 2017; Published: 1 March 2017

Abstract: Localization is a key technology in wireless sensor networks. Faced with the challenges of the sensors' memory, computational constraints, and limited energy, particle swarm optimization has been widely applied in the localization of wireless sensor networks, demonstrating better performance than other optimization methods. In particle swarm optimization-based localization algorithms, the variants and parameters should be chosen elaborately to achieve the best performance. However, there is a lack of guidance on how to choose these variants and parameters. Further, there is no comprehensive performance comparison among particle swarm optimization algorithms. The main contribution of this paper is three-fold. First, it surveys the popular particle swarm optimization variants and particle swarm optimization-based localization algorithms for wireless sensor networks. Secondly, it presents parameter selection of nine particle swarm optimization variants and six types of swarm topologies by extensive simulations. Thirdly, it comprehensively compares the performance of these algorithms. The results show that the particle swarm optimization with constriction coefficient using ring topology outperforms other variants and swarm topologies, and it performs better than the second-order cone programming algorithm.

Keywords: wireless sensor networks; particle swarm optimization; localization; parameter selection; performance comparison

1. Introduction

Wireless sensor networks (WSNs) are an important infrastructure of the Internet of Things used for sensing the surrounding information, whose applications can be classified into monitoring and tracking in the fields of military and public [1]. In these applications, spatial information is one of the most important contexts of the sensed data, and the location information can support the coverage, routing, and many other operations of a WSN. However, the sensor nodes of a WSN are usually deployed in an ad hoc manner without any prior knowledge of their locations, so it is essential to determine the node's location, which is referred to as *localization*.

A possible solution is to equip each sensor node with a global positioning system (GPS) device, but it is not suitable for large-scale deployment due to the constraints of cost and energy. Hence, only a part of sensor nodes (named *anchors*) are equipped with GPS devices. These anchors serve as references to the other nodes (named *unknown nodes*), which are to be localized. There exist well-organized

overviews of sensor localization algorithms [2,3]. Localization consists of ranging and estimation phases. In the ranging phase, the nodes measure their distances from anchors using received signal strength, time of arrival, time difference of arrival, link quality indicator, or angle of arrival. In the estimation phase, the node's position is estimated based on the ranging information.

A popular estimation approach is to formulate the localization problem as an optimization problem, and then use an optimization algorithm to solve the problem. Traditional optimization algorithms are widely used in localization [4–7], such as least square, maximum-likelihood, semi-determined programming, and second-order cone programming. Recently, soft computing algorithms have been widely applied to solve this problem [8,9], such as Cuckoo search algorithm [10], artificial neural network [11], bacterial foraging algorithm [12], bat algorithm [13], and biogeography-based optimization [9,14].

Particle swarm optimization (PSO) is also an important soft computing algorithm which models the behavior of a flock of birds. It utilizes a population of *particles* to represent candidate solutions in a search space, and optimizes the problem by iteration to move these particles to the best solutions with regard to a given measure of quality. Compared with the above algorithms, the advantages of particle swarm optimization are the following [15–17].

1. Ease of implementation on hardware or software.
2. High-quality solutions because of its ability to escape from local optima.
3. Quick convergence.

Recently, PSO has been used in many issues of WSNs [17,18]. This paper focuses on the PSO-based localization algorithms for static WSNs, where all sensors are static after deployment. Some PSO-based localization algorithms with different population topologies are compared in Cao et al. [19]. However, they do not consider the recently-proposed PSO-based localization algorithms, nor do they give parameter selections. The existing PSO's parameter selection guidelines [16] are not based on the objective function in localization problem of WSN, so these parameters cannot achieve the optimal localization performance.

The main contributions of this paper are as follows.

1. It surveys the popular particle swarm optimization variants and particle swarm optimization-based localization algorithms for wireless sensor networks.
2. It presents parameter selection of nine particle swarm optimization variants and six types of swarm topologies by extensive simulations.
3. It comprehensively compares the performance of these algorithms.

The rest of the paper is organized as follows. The localization problem and PSO are introduced in Section 2. Section 3 surveys the PSO-based localization algorithms, and Section 4 presents their parameter selections. Section 5 compares the performance of PSO-based localization algorithms. Section 6 concludes the paper and presents the future work.

2. Statements of Localization Problem and PSO

2.1. Localization Problem for Static WSNs

A WSN is a network of N sensor nodes, including N_A anchors and N_U unknown nodes, where $N_A \ll N_U$. The WSN is deployed in a two-dimensional region of interest at random. The region is often assumed as a square of side-length L . Suppose all the sensor nodes have the same communication range, which is a circle of radius R . Two sensor nodes are called *neighbors* if one of them lays in the communication range of the other, and the distance between them can be measured. Obviously, an unknown node can be localized if it has at least three neighboring anchors.

Since $N_A \ll N_U$, the localization of static WSNs is an iterative procedure. The unknown nodes with at least three neighboring anchors are localized first, and then the other unknown nodes are

localized based on the information of neighboring anchors and localized unknown nodes iteratively. Both the anchors and localized unknown nodes are called *reference nodes* in this paper.

During localization, suppose an unknown node U_j ($j = 1, 2, \dots, N_U$) has n neighboring reference nodes. Let \hat{d}_{kj} be the measured distance between U_j and reference node B_k ($k = 1, 2, \dots, n$), $(\hat{x}_{u_j}, \hat{y}_{u_j})$, and (x_{u_j}, y_{u_j}) be the estimated and actual positions of U_j , and (x_{b_k}, y_{b_k}) be the position of B_k . Then, the localization result should satisfy

$$\hat{d}_{kj} = \sqrt{(\hat{x}_{u_j} - x_{b_k})^2 + (\hat{y}_{u_j} - y_{b_k})^2} \quad (1)$$

The distances measured by any ranging method are inaccurate, so it is impossible to find an accurate solution to (1). Let \tilde{d}_{kj} be the right part of (1), and be referred to as *estimated distance*. Obviously,

$$\tilde{d}_{kj} \neq \hat{d}_{kj} \quad (2)$$

because of the inaccurate measured distances. Then, the purpose of localization is to minimize the difference between \hat{d}_{kj} and \tilde{d}_{kj} , which is

$$f_1 = \sum_{k=1}^n (\hat{d}_{kj} - \tilde{d}_{kj})^2 \quad (3)$$

For most ranging techniques, the measurement error is related to the distance between the two sensor nodes, and larger distances cause larger error [20]. Hence, weight w_k is introduced so that the nearer neighboring reference nodes play a greater role in localization, as shown in (4).

$$f_2 = \sum_{k=1}^n w_k (\hat{d}_{kj} - \tilde{d}_{kj})^2, \quad (4)$$

where w_k is the weight of the neighboring reference node B_k of U_j , defined as

$$w_k = \frac{\frac{1}{\hat{d}_{kj}}}{\sum_{k=1}^n \frac{1}{\hat{d}_{kj}}} \quad (5)$$

Besides, the following equations are also discussed in related literature:

$$f_3 = \frac{1}{n} f_1 \quad (6)$$

$$f_4 = \frac{1}{n} f_2 \quad (7)$$

which are, respectively, the average of f_1 and f_2 on number of neighboring reference nodes n .

Equations (3), (4), (6) and (7) are called *objective functions*.

2.2. Particle Swarm Optimization

The PSO-based localization algorithm uses PSO to solve one of the above objective functions. Because the WSN considered in this paper is in a two-dimensional region, the search space of the PSO is constrained to two-dimensions.

Let M be the number of particles of the PSO. Particle i occupies three two-dimensional vectors P_i , Q_i , and V_i , representing its current location, previous best position, and current velocity. Besides, Gb denotes the position of the best particle so far. In each iteration, particle i updates its position and velocity according to the following equations.

$$\begin{aligned} V_{ij} &= \omega V_{ij} + c_1 \varphi_{1j}(Q_{ij} - P_{ij}) + c_2 \varphi_{2j}(Gb_j - P_{ij}) \\ P_{ij} &= P_{ij} + V_{ij} \end{aligned} \quad j = 1, 2, \quad (8)$$

where c_1 and c_2 are cognitive and social acceleration coefficients, respectively, φ_{1j} and φ_{2j} are random numbers uniformly distributed in $(0, 1)$, and ω is the inertia weight. c_1 propels the particle towards the position where it had the best fitness, while c_2 propels the particle towards the current best particle. The stochastic V_i may become too high to keep all particles in the search space. Hence, V_{max} is introduced [16] to bound V_{ij} within the range $[-V_{max}, V_{max}]$.

ω is an important parameter. Linearly decreasing and simulated annealing types are the best ones of all adjustment methods [21]. Due to the computational and memory constraints of sensor nodes, the linearly decreasing method is adopted in many PSO-based localization algorithms, which is

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{t_{max}} \times t \quad (9)$$

where t_{max} is the maximum number of allowable iterations, ω_{max} and ω_{min} are maximum and minimum weights, respectively, and t is the current iteration.

As we can see, PSO needs each particle to communicate/connect with the other particles to obtain Gb . The connections among particles are called *topology*. There are two kinds of topologies: *global-best* and *local-best*. The former allows each particle to access the information of all other particles, and the latter only allows each particle to access the information of its neighbors according to different local-best topology [22,23]. Because each particle has a different swarm of neighboring particles in its local-best topology, this topology ensures that the particles have full diversity. Local-best topology uses Lb_i instead of Gb to represent the best position of the neighboring particles of particle i , and its update function is:

$$\begin{aligned} V_{ij} &= \omega V_{ij} + c_1 \varphi_{1j}(Q_{ij} - P_{ij}) + c_2 \varphi_{2j}(Lb_{ij} - P_{ij}) \\ P_{ij} &= P_{ij} + V_{ij} \end{aligned} \quad j = 1, 2. \quad (10)$$

Here, c_2 propels the particle towards the current best particle within the corresponding sub-swarm of this particle. The most popular local-best swarm topologies include:

1. *Ring topology*: Each particle is affected by its k immediate neighbors.
2. *Star/Wheel topology*: Only one particle is in the center of the swarm, and this particle is influenced by all other particles. However, each of the other particles can only access the information of the central particle.
3. *Pyramid topology*: The swarm of particles are divided into several levels, and there are 4^l particles in level l ($l \geq 0$), which form a $2^l \times 2^l$ mesh.
4. *Von Neumann topology*: All particles are connected as a toroidal, and each particle has four neighbors, which are above, below, left, and right particles.
5. *Random topology*: Each particle chooses neighbors randomly at each iteration. We utilize the second algorithm proposed in [24] to generate the random topology.

2.3. Evaluation Criteria of PSO-Based Localization Algorithms

1. *Localization error*. The localization error of unknown node U_j is defined as $E_j = \frac{1}{R} \sqrt{(\hat{x}_{u_j} - x_{u_j})^2 + (\hat{y}_{u_j} - y_{u_j})^2}$. The mean and standard deviation of localization error are denoted by \bar{E} and σ_E , respectively.
2. *Number of iterations*. This is the number of iterations of PSO to achieve the best fitness. The mean and standard deviation of the number of iterations are denoted by \overline{Itr} and σ_{Itr} , respectively.

In WSN applications, the localization error depends on ranging error, GPS error, localization error accumulation, and the localization algorithm. The ranging error results from the distance measurement technique, and the GPS error determines the errors of anchors' positions. Both ranging and GPS errors are assumed to obey Gaussian distribution, and they are denoted by e . The accumulated localization error comes from the iterative localization procedure: some unknown nodes may utilize localized unknown nodes to localize themselves, while the positions of these localized unknown nodes already have localization error.

3. A Survey of PSO-Based Localization Algorithms

3.1. Basic Procedure of PSO-Based Localization Algorithms

After measuring the distance between sensor nodes, each unknown node estimates its location by Algorithm 1. In this algorithm, Line 1 determines the particles' search space, which is the intersection region of the radio range of all neighboring reference nodes of this sensor node. After initialization in Line 2, it uses an iterative process to estimate the position (Lines 3 to 7). Note that the update process of Line 6 is different according to different PSO-based localization algorithms.

Algorithm 1 PSO-Based Localization Algorithm

- 1: Determine the particles' search space.
 - 2: Initialize a swarm of particles in the search space with random positions and velocities.
 - 3: **while** stop criteria are not met **do**
 - 4: Compute the fitness values of all particles.
 - 5: Compute Q_i , Lb_i , and/or Gb .
 - 6: Update each particle.
 - 7: **end while**
-

3.2. PSO-Based Localization Algorithms

The basic PSO algorithm using (8) is the most popular one among all PSO-based localization algorithms, and we use WPSO (weighted-PSO) to represent it for convenience.

The first WPSO-based localization algorithm was proposed in [25], which uses (6) as the objective function. WPSO is also applied to the localization problem of ultra-wide band sensor networks in [26,27]. In a static sensor network, some unknown nodes may not have enough neighbor anchors to localize themselves, so [28] applies DV (Distance Vector)-Distance to make all unknown nodes have distances to at least three anchors, and then it uses WPSO to localize. In order to make WPSO's convergence rate fast, [29] introduces a threshold to constrain a change of fitness function. Due to the inaccurate distance measurements, flip ambiguity is popular during localization, but this problem is not considered by the aforementioned algorithms. In [30], WPSO in conjunction with two types of constraints is used to cope with this problem. Besides two-dimensional sensor networks, WPSO is also applied to localize three-dimensional WSNs [27,31,32] and underwater WSNs [33]. Different from the above algorithms, a mobile anchor-assisted WPSO-based localization algorithm is proposed in [20,34], which only uses one mobile anchor to provide distance range to all unknown nodes while it traverses the sensor network. Besides localization, [35] also applies WPSO to the real-time autonomous deployment of sensor nodes (including anchors and unknown nodes) from an unmanned aerial vehicle.

Besides WPSO, many variants of PSO algorithms have been proposed to improve the performance. The most representative algorithms are listed below.

1. *Constricted PSO (CPSO)*. WPSO has the disadvantages of early convergence and swarm explosion, so CPSO [36] introduces the constriction coefficient χ to conquer these disadvantages:

$$\begin{aligned} V_{ij} &= \chi(V_{ij} + \varphi_{1j}(Q_{ij} - P_{ij}) + \varphi_{2j}(Gb_j - P_{ij})) \\ P_{ij} &= P_{ij} + V_{ij} \end{aligned} \quad j = 1, 2, \quad (11)$$

where $\chi = \frac{2}{|2-c-\sqrt{c^2-4c}|}$, $c = c_1 + c_2 > 4.0$. CPSO eliminates the V_{max} in WPSO. It performs better than WPSO in many problems [36].

2. *H-Best PSO (HPSO)*. Global- and local-best PSO algorithms have their own advantages and disadvantages. Combining these two algorithms, HPSO [9,14] divides the particles into several groups, and particle i is updated based on Gb , Lb_i , and Q_i , per the following equation:

$$\begin{aligned} V_{ij} &= \omega V_{ij} + c_1 \varphi_{1j}(Q_{ij} - P_{ij}) + c_2 \varphi_{2j}(Gb_j - P_{ij}) + c_3 \varphi_{3j}(Lb_{ij} - P_{ij}) \\ P_{ij} &= P_{ij} + V_{ij} \end{aligned} \quad j = 1, 2. \quad (12)$$

Here, c_3 is the same as c_2 of (10), and φ_{3j} is a random number uniformly distributed in $(0, 1)$. HPSO provides fast convergence and swarm diversity, but it utilizes more parameters than WPSO.

3. *PSO with particle permutation (PPSO)*. In order to speed up the convergence, PPSO [37] sorts all particles such that $f(P_i) \leq f(P_j)$, if $i \leq j$, and replaces the positions of particles $\lfloor \frac{M}{2} \rfloor + 1$ to M with positions close to P_1 . The rule of replacement is:

$$P_{\lfloor \frac{M}{2} \rfloor + k, j} = P_{1j} + \rho_{kj} \quad j = 1, 2; \quad k = 1, 2, \dots, \lfloor \frac{M}{2} \rfloor, \quad (13)$$

where ρ_{kj} is a random number uniformly distributed in $(-0.5, 0.5)$.

4. *Extremum disturbed and simple PSO (EPSO)*. Sometimes, PSO easily fall into local extrema. EPSO [38] uses two preset thresholds T_0 and T_g to randomly churn Q_i and Gb to overcome this shortcoming. The operators of extremal perturbation are:

$$Q'_{ij} = \varphi_{4j} Q_{ij}, \quad Gb'_j = \varphi_{5j} Gb_j \quad j = 1, 2. \quad (14)$$

Let t_0 and t_g be evolutionary stagnation iterations of Q_i and Gb , respectively. In (14), if $t_0 \leq T_0$ ($t_g \leq T_g$), φ_{4j} (φ_{5j}) is 1; Otherwise, φ_{4j} (φ_{5j}) is a random number uniformly distributed in $[0, 1]$. For particle i , the update function of EPSO is

$$P_{ij} = \eta_{ij} \omega P_{ij} + c_1 \varphi_{1j}(Q'_{ij} - P_{ij}) + c_2 \varphi_{2j}(Gb'_j - P_{ij}) \quad j = 1, 2, \quad (15)$$

where η_{ij} is used to control the movement direction of particle i to make the algorithm convergence fast, which is defined as

$$\eta_{ij} = \frac{P_{ij} - Gb_j}{\sqrt{(P_{i1} - Gb_1)^2 + (P_{i2} - Gb_2)^2}} \quad j = 1, 2. \quad (16)$$

5. *Dynamic PSO (DPSO)*. Each particle in DPSO [19] pays full attention to the historical information of all neighboring particles, instead of only focusing on the particle which gets the optimum position in the neighborhood. For particle i , the update function of DPSO is

$$P_{ij} = P_{ij} + c_1(P_{ij} - P'_{ij}) + \frac{c_2}{K} \sum_{k=1}^K (Q_{kj} - P_{ij}) + \frac{c_3 \varphi_{3j}}{K} \sum_{k=1}^K |Q_{kj} - Q_{ij}| \quad j = 1, 2, \quad (17)$$

where K is the number of neighboring particles of the i th particle. P'_i is the previous position of the i th particle. Note that c_1 , c_2 , and c_3 are just weights without physical meaning.

6. *Binary PSO (BPSO)*. BPSO [39] is used in binary discrete search space, which applies a sigmoid transformation to the speed attribute in update function, so its update function of particle i is

$$\begin{aligned} V_{ij} &= \omega V_{ij} + c_1 \varphi_{1j} (Q_{ij} - \text{sigmoid}(P_{ij})) + c_2 \varphi_{2j} (Gb_j - \text{sigmoid}(P_{ij})) \\ P_{ij} &= P_{ij} + V_{ij} \end{aligned} \quad j = 1, 2, \quad (18)$$

where $\text{sigmoid}(P_{ij})$ is defined as

$$\text{sigmoid}(P_{ij}) = \begin{cases} 1 & \text{if } \varphi_6 < 2 \left| \frac{1}{1+e^{-V_{ij}}} - 0.5 \right| \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

φ_6 is a random number.

7. *PSO with time variant ω , c_1 , and c_2 (TPSO)*. TPSO [19] employs time-varying c_1 , c_2 , and ω (see Equation (9)) to achieve proper balance between global and local exploitation, where

$$c_1 = (c_{1f} - c_{1s}) \frac{t}{t_{max}} + c_{1s}, \quad c_2 = (c_{2f} - c_{2s}) \frac{t}{t_{max}} + c_{2s}, \quad (20)$$

where c_{1s} , c_{1f} , c_{2s} , and c_{2f} are *initial* and *final* values of c_1 and c_2 , respectively.

8. *PSO with particle migration (MPSO)* [40]. MPSO enhances the diversity of particles and avoids premature convergence. MPSO randomly partitions particles into several sub-swarms, each of which evolves based on TPSO, and some particles migrate from one sub-swarm to another during evolution.

In one word, the above-mentioned PSO variants aim to overcome one or more drawbacks of WPSO, but they also introduce additional operations.

3.3. Comparison between PSO and Other Optimization Algorithms

The above-mentioned algorithms are also compared with the other algorithms in corresponding references, and Table 1 summarizes the comparison results.

Table 1. Advantages of particle swarm optimization (PSO)-based localization algorithms. HPSO: H-Best PSO; PPSO: PSO with particle permutation; WPSO: weighted-PSO.

| Algorithm | References | Comparative | Advantages of PSO |
|-----------|------------|--|-----------------------|
| WPSO | [12,35] | bacterial foraging algorithm | faster |
| WPSO | [25] | simulated annealing | more accurate |
| WPSO | [41] | Gauss–Newton algorithm | more accurate |
| WPSO | [26,31] | least square | more accurate |
| WPSO | [42] | simulated annealing, semi-definite programming | faster, more accurate |
| WPSO | [28] | artificial neural network | more accurate |
| WPSO | [32] | least square | faster, more accurate |
| HPSO | [9,14] | biogeography-based optimization | faster |
| PPSO | [37] | two-stage maximum-likelihood, plane intersection | faster, more accurate |

The bacterial foraging algorithm models the foraging behavior of bacteria that thrive to find nutrient-rich locations. Each bacterium moves using a combination of tumbling and swimming. Tumbling refers to a random change in the direction of movement, and swimming refers to moving in a straight line in a given direction.

The simulated annealing algorithm originated from the formation of crystals from liquids. Initially, the simulated annealing algorithm is in a high energy state. At each step, it considers some neighbouring state of the current state, and probabilistically decides between moving the system to one neighbouring state or staying in the current state. These probabilities ultimately lead the system to move to states of lower energy.

Artificial neural networks model the human brain in performing an intelligent task. It integrates computational units (neurons) in multi-layers, and these layers are connected by adjustable weights. Three traditional layers are input, hidden, and output.

The biogeography-based optimization algorithm is motivated by the science of biogeography, which investigates the species distribution and its dynamic properties from past to present spatially and temporally. In this algorithm, the candidate solutions and their features are considered as islands and species, respectively. Species migrate among islands, which is analogous to candidate solutions' interaction.

4. Parameter Selections of PSO-Based Localization Algorithms

Given a problem to be solved, the performance of a PSO depends on its parameters. Although theoretical analysis can guide the parameter selection, this analysis can occupy large space, such as [43], which exceeds the limit of this paper. On the other hand, extensive experimentation has been used widely in the parameter selection or performance analysis of PSO [15,21,23]. Therefore, we try to choose the best parameter by experiments instead of theoretical analysis. Because some parameters of HPSO and PPSO have been calibrated in corresponding references, we only calibrate parameters of WPSO, CPSO, MPSO, TPSO, BPSO, EPSO, and DPSO, and M and V_{max} of PPSO and HPSO.

4.1. Simulation Setup

The PSO-based localization algorithms are implemented in C language, and the results are analyzed by Matlab. Because our aim is to decide the best parameters, we only perform the localization procedure on one unknown node, denoted by U_1 . Suppose the actual position of U_1 is $(x_{u_1}, y_{u_1}) = (0, 0)$, and N_B reference nodes are deployed within the communication range of U_1 . Considering the error e during distance measurements, \hat{d}_{k1} ($k = 1, 2, \dots, N_B$) is defined as

$$\hat{d}_{k1} = d_{k1} + \alpha, \quad (21)$$

where d_{k1} is the real distance between U_1 and B_k , α is a random number that follows a normal distribution with mean 0 and variance $d_{k1}e$.

The simulation setup is shown in Table 2. For convenience, we use $a:b:c$ to represent the set of $[a, a + b, \dots, c]$ in this paper. The parameter selection procedure is:

1. Using $V_{max} = 0.5R$ and $M = 20, 30, 40, 50$ to find out the best c_1, c_2, c_3 , and ω .
2. Using the best c_1, c_2, c_3 and ω to choose M .
3. Using the best c_1, c_2, c_3, ω and M to choose V_{max} .
4. Using the best c_1, c_2, c_3, ω, M and V_{max} to compare fitness functions.

Table 2. Simulations setup. CPSO: Constricted PSO; DPSO: Dynamic PSO; EPSO: Extremum disturbed and simple PSO; MPSO: PSO with particle migration; TPSO: PSO with time variant ω, c_1 , and c_2 .

| Type | Values |
|----------------------|---|
| network | $R = 5:10:45, e = 0.05:0.05:0.2, N_B = 3:20$ |
| General ^a | $V_{max} = 0.1R:0.1R:2R, M = 5:5:100, t_{max} = 500$ |
| WPSO, EPSO | $\omega_{max} = 1.4:-0.1:0.9, \omega_{min} = 0.4:-0.1:0, c_1 = 1.7:0.1:2.5, c_2 = 1.7:0.1:2.5$ |
| CPSO | $c_1 = 2.05:0.05:2.5, c_2 = 2.05:0.05:2.5$ |
| MPSO, TPSO | $\omega_{max} = 1.4:-0.1:0.9, \omega_{min} = 0.4:-0.1:0, c_{1i} = c_{2f} = 3:-0.25:0.25, c_{2i} = c_{1f} = c_1 - 0.25:-0.25:0.25$ |
| HPSO | $\omega = 0.7, c_1 = c_2 = c_3 = 1.494$ |
| PPSO | $\omega = 1, c_1 = c_2 = 2.0$ |
| BPSO | $\omega = 1.0:-0.1:0, c_1 = 1.7:0.1:2.5, c_2 = 1.7:0.1:2.5$ |
| DPSO | $c_1 = 1.0:0.1:0.1, c_2 = 2.5:-0.1:1.5, c_3 = 0:0.1:1.0$ |

^a The parameters used by all PSO variants.

We use f_3 as the objective function of PSO to choose the best parameters, because it is the most popular fitness used in PSO-based localization algorithms. The stopping criteria is

$$f_3(X) \leq 10^{-4} \vee t > t_{max}, \quad (22)$$

which means the fitness value $f_3(X)$ achieves an allowable precision 10^{-4} or the number of iteration t exceeds pre-defined threshold t_{max} .

We generate 100 tests with each combination of different e , R , and N_B , and we utilize each algorithm to estimate the position of U_1 by 100 runs for each test case, under each group of PSO's parameters. The results are the average of these runs. A localization algorithm should have the best performance regardless of N_B , because N_B is different to each unknown node in real applications of WSN. Hence, the impacts of N_B on parameter selection are not analyzed.

4.2. Best Parameters of PSO-Based Localization Algorithms

Taking the selection of c_1 and c_2 of WPSO with global-best model as an example, the selection approach is introduced. We first calibrate c_1 and c_2 without considering the specific value of e , because we may not know e during localization. The results are the averages of all e . Figure 1a,b show that the impacts of c_1 and c_2 on localization errors are very little: the gap between the minimum and maximum \bar{E} (σ_E) is about 4.42×10^{-5} (1.79×10^{-4}). On the contrary, Figure 1c,d show that \bar{Itr} and σ_{Itr} reduce with the decrease of c_1 and c_2 . With different c_1 and c_2 , the difference between the maximum and minimum \bar{Itr} (σ_{Itr}) is about 31.03 (11.70). The best choices are $c_1 = 1.7$ – 1.8 and $c_2 = 1.7$ – 1.8 , which occupy 96.2% optimal number of iterations for all cases. Further, we investigate the impacts of c_1 and c_2 on localization performance under each e , and we find that $c_1 = 1.7$ – 1.8 and $c_2 = 1.7$ – 1.8 are still the best choice for each e .

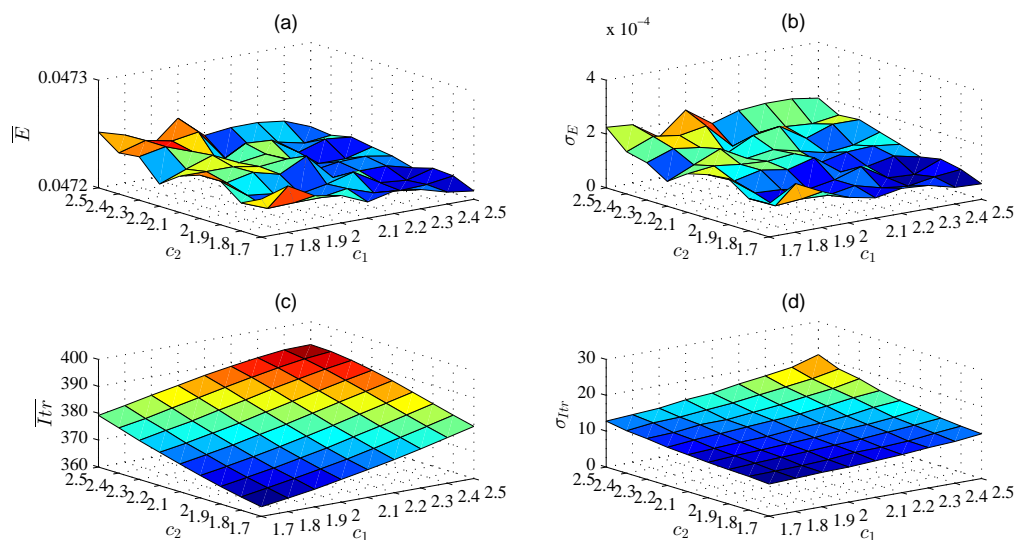


Figure 1. Localization performance of different c_1 and c_2 using WPSO with global-best model. (a) \bar{E} ; (b) σ_E ; (c) \bar{Itr} ; (d) σ_{Itr} .

Similar approaches have been applied to other parameters and algorithms, and the resulting best parameters are shown in Table 3. Because MPSO and HPSO already divide the swarm of particles into several sub-swarms which are similar with local-best models, we only analyze their global-best models.

Table 3. Parameter selections of PSO-based localization algorithms. BPSO: Binary PSO.

| Variant | Topology | $[c_1, c_2, c_3]$ | $[\omega_{max}, \omega_{min}]$ | M | V_{max} |
|---------|-------------|-----------------------------------|--------------------------------|-------|-----------|
| WPSO | global-best | [1.7–1.8,1.7–1.8,—] | [0.9,0] | 30 | 0.1R |
| | pyramid | [1.7,1.7–1.8,—],[1.8,1.7,—] | [0.9,0] | 21 | 0.1R |
| | random | [1.7,1.7–1.8,—] | [0.9,0] | 30 | 0.1R |
| | Von Neumann | [1.7,1.7–1.8,—],[1.8,1.7] | [0.9,0] | 25 | 0.1R |
| | ring | [1.7,1.7–1.8,—],[1.8,1.7,—] | [0.9,0] | 25 | 0.1R |
| | star | [1.7,1.7–1.8,—],[1.8,1.7,—] | [0.9,0] | 25 | 0.1R |
| CPSO | global-best | [2.4,2.5,—],[2.45,2.45–2.5,—] | — | 10 | — |
| | pyramid | [2.45,2.5,—],[2.5,2.45–2.5,—] | — | 21 | — |
| | random | [2.5,2.5,—] | — | 10 | — |
| | Von Neumann | [2.45,2.45–2.5,—],[2.5,2.4–2.5,—] | — | 10 | — |
| | ring | [2.45,2.5,—],[2.5,2.4–2.5,—] | — | 10 | — |
| | star | [2.4,2.5,—],[2.45,2.45–2.5,—] | — | 10 | — |
| TPSO | global-best | [0.5,0.25,—],[0.75–1,0.25–0.5,—] | [0.9,0] | 25 | 0.1R |
| | pyramid | [0.5–1,0.25,—],[0.75,0.25–0.5,—] | [0.9,0] | 21 | 0.1R |
| | random | [0.5,0.25,—],[0.75–1,0.25–0.5,—] | [0.9,0] | 30 | 0.1R |
| | Von Neumann | [0.5–1.25,0.25,—] | [0.9,0] | 25 | 0.1R |
| | ring | [0.75–1.25,0.25–0.5,—] | [0.9,0] | 25 | 0.1R |
| | star | [0.5–1,0.25–0.25,—] | [0.9,0] | 15 | 0.1R |
| PPSO | all | [2.0,2.0,—] | $\omega = 1.0$ | 20–40 | 0.1R |
| EPSO | global-best | [2.5,1.7,—] | [1.4,0.4] | 45 | 0.1R |
| | pyramid | [2.5,1.8,—] | [1.4,0.4] | 21 | 0.1R |
| | random | [2.4,2.5,—] | [0.9,0] | 20 | 0.1R |
| | Von Neumann | [2.4,2.5,—] | [0.9,0] | 30 | 0.1R |
| | ring | [2.4,2.5,—] | [0.9,0] | 25 | 0.1R |
| | star | [2.4,2.5,—] | [0.9,0] | 20 | 0.1R |
| DPSO | global-best | [0.7/0.8,2.1,0.4/0.5] | — | 75 | 0.1R |
| | pyramid | [0.7/0.8/0.9,2.3,0.4] | — | 21 | 0.1R |
| | random | [0.9,2.3,0.4] | — | 30 | 0.1R |
| | Von Neumann | [0.7/0.8/0.9,2.3,0.4] | — | 35 | 0.1R |
| | ring | [0.7/0.8/0.9,2.3,0.4] | — | 35 | 0.1R |
| | star | [0.7/0.8,2.1,0.4/0.5] | — | 45 | 0.1R |
| BPSO | global-best | [2.1/2.2,1.7,—] | $\omega = 1.4/1.5$ | 75 | 0.1R |
| | pyramid | [2.3/2.4,1.7,—] | $\omega = 1.4$ | 21 | 0.1R |
| | random | [1.9,2.0,—] | $\omega = 1.3/1.4/1.5$ | 75 | 0.1R |
| | Von Neumann | [2.0–2.3,1.7,—] | $\omega = 1.5$ | 80 | 0.1R |
| | ring | [2.1/2.2,1.7,—] | $\omega = 1.4/1.5$ | 35 | 0.1R |
| | star | [2.5,1.9,—] | $\omega = 1.3$ | 80 | 0.1R |
| MPSO | global-best | [1/1.25,0.25/0.5,—],[1.5,0.25,—] | [0.9,0] | 45 | 0.1R |
| HPSO | global-best | [1.494,1.494,1.494] | $\omega = 0.7$ | 45 | 0.1R |

4.3. Performance Comparison of Fitness Function

Figure 2a,b shows that \overline{Itr} of f_4 is [306.86, 326.41], which is about 89.65%–91.57% of those of f_1 , f_2 , and f_3 . Figure 2c,d shows that f_4 is still the best. In detail, f_3 and f_4 has almost the same \bar{E} , which is [0.035, 0.048]. However, f_4 has the minimal σ_E .

The performance of each fitness function using the other PSO algorithms with different swarm topologies also shows that f_4 outperforms the other functions in all cases.

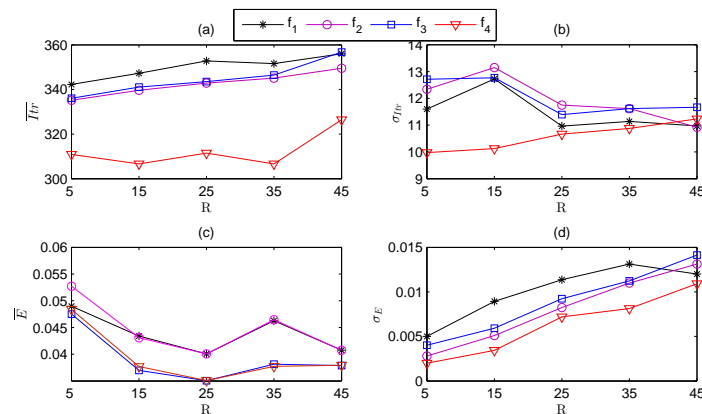


Figure 2. Localization performance of different f using WPSO with global-best model. (a) \overline{Itr} ; (b) σ_{Itr} ; (c) \overline{E} ; (d) σ_E .

5. Performance Comparisons of PSO-Based Localization Algorithms

5.1. Simulations Setup

The performance of all PSO algorithms is compared by simulations of a whole WSN. The parameters of each PSO variant are set based on Section 4. During localization, we utilize iterative localization to localize as many unknown nodes as possible. We also demonstrate the impact of radio irregularity on the localization performance of different algorithms, because the actual transmission range of sensor nodes is not a perfect circle due to multi-path fading, shadowing, and noise. The radio model in [44] is used to represent the degree of irregularity D . Based on this model, there is an upper bound R and a lower bound $(1 - D)R$ of the communication range. The simulation setups are: $L = 100$ units, $N_U = 100:100:500$, $N_A = 10:10:50$, $R = 25:10:45$ units, $e = 0:0.05:0.2$, $D = 0:0.1:0.5$.

5.2. Comparisons of Different Swarm Topologies in Same PSO

There are 2250 test cases to simulate, and the best swarm topology should outperform the other ones in most test cases, instead of in several test cases. Therefore, we count the number of optimal values of each swarm topology of all test cases, and try to find the topology which achieves the optimal values in most test cases. The “optimal values” are the evaluation criteria mentioned in Section 2.3. The results are shown in Figure 3.

For WPSO, Von Neumann and ring topologies perform almost the same, and they outperform the other topologies. Ring topology can gain the optimal \overline{E} , σ_E , \overline{Itr} , and σ_{Itr} in more than 93.2%, 99.5%, 98.1%, and 99.5% of test cases, respectively, and Von Neumann topology can gain the optimal \overline{E} , σ_E , \overline{Itr} , and σ_{Itr} in more than 93.4%, 99.5%, 97.5%, and 99.5% of test cases, respectively.

It is hard to say which topology is the best for some algorithms, because none of these topologies performs the best in all four percentages. In this case, we give more importance to \overline{E} than the other criteria. Taking PPSO as an example (Figure 3d), its global-best and Von Neumann topologies are obviously better than the others, but Von Neumann outperforms global-best model in \overline{E} , while global-best model is better than Von Neumann in the other three criteria. Since we put \overline{E} as the first criterion to choose the best swarm topology, Von Neumann is the best one which achieves optimal \overline{E} in more than 92.3% cases, while the percentage of global best model is 70.7%.

Using a similar idea, the best swarm topologies of all algorithms are: ring topology for WPSO, CPSO, and TPSO; Von Neumann topology for PPSO and DPSO; random topology for EPSO and BPSO.

Furthermore, we compare the performance of different swarm topologies used in the same PSO algorithm, with different N_U , N_A , R , D , e one by one, and we find these results are consistent with the aforementioned results.

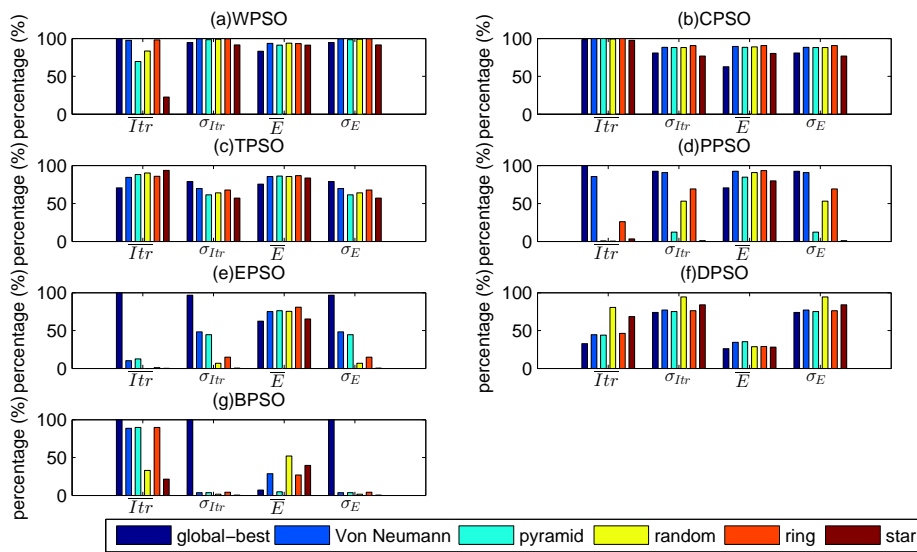


Figure 3. Percentage of each swarm topology achieving the optimal values. (a) WPSO; (b) CPSO; (c) TPSO; (d) PPSO; (e) EPSO; (f) DPSO; (g) BPSO.

5.3. Comparison of PSO-Based Localization Algorithms

Using the results introduced in Section 4, the PSO-based localization algorithms are compared.

5.3.1. General Analysis

In general, as shown in Figure 4, CPSO and PPSO needs fewer iterations than the other PSO-based localization algorithms. In fact, the average $\overline{I_{tr}}$ of all test cases required by CPSO and PPSO are 220.16 and 222.29, respectively, while those of WPSO, TPSO, MPSO, HPSO, EPSO, DPSO, and BPSO are 312.88, 484.99, 268.84, 281.68, 310.26, 253.52, and 252.47, respectively. $\sigma_{I_{tr}}$ of CPSO is also the smallest (6.67), while $\sigma_{I_{tr}}$ of the other algorithms are larger than 10. On the other hand, CPSO and PPSO achieve the optimal \overline{E} for more than 89.69% of test cases, and the optimal σ_E for more than 60.84% of test cases. Furthermore, the average \overline{E} and σ_E of all test cases of CPSO and PPSO are the first two best ones among all algorithms. CPSO has the least operations during one iteration among all algorithms. Compared with CPSO, the other algorithms need more parameters and operations such, as V_{max} , particle sort, or migration.

In one word, CPSO is the best PSO-based localization algorithm.

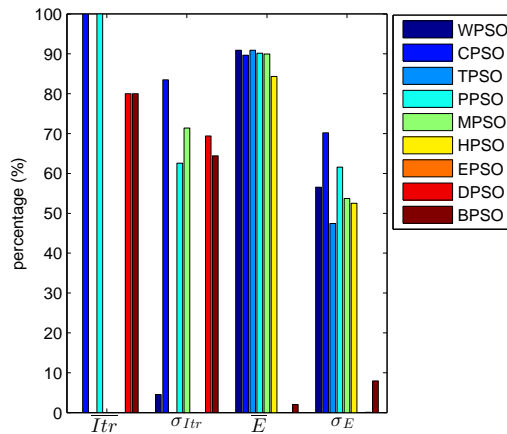


Figure 4. Percentage of PSO-based localization achieving the optimal values.

5.3.2. Impacts of Network Parameters on Performance

In order to analyze the impacts of network setups on localization performance, CPSO is taken as an example, because we find that the other algorithms have the same trends as CPSO, except the variation range. As illustrated in Figure 5, we cannot draw any rule of the impacts of N_U and N_A on $\bar{I}tr$, σ_{Itr} . However, \bar{E} and σ_E decrease as N_A increases, because the more anchors exist in a WSN, the less unknown nodes need iterative localization, as shown in Figure 6. Figure 6 shows that the number of unknown nodes localized by neighboring anchors instead of localized unknown nodes under $N_A = 500$ is about two times the number under $N_A = 100$.

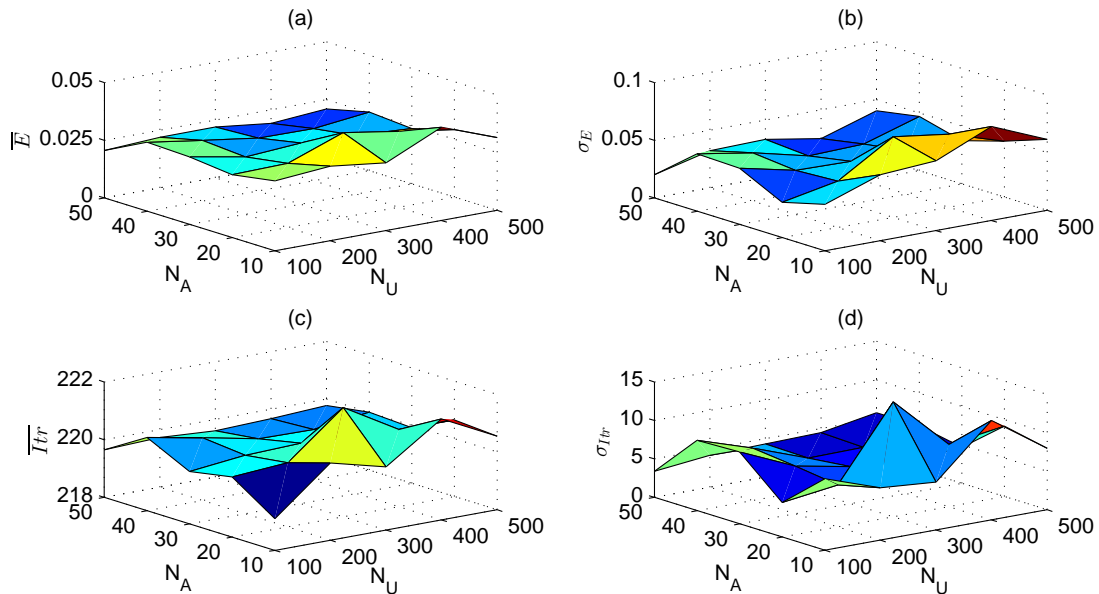


Figure 5. Impacts of N_U and N_A on the localization performance of CPSO. (a) \bar{E} ; (b) σ_E ; (c) $\bar{I}tr$; (d) σ_{Itr} .

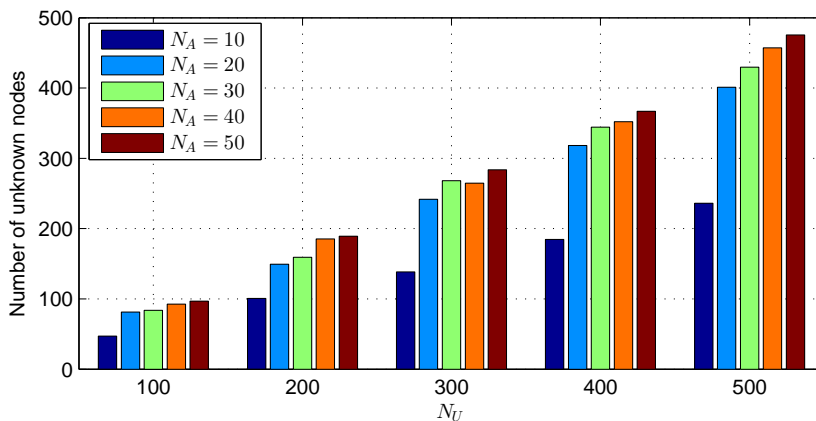


Figure 6. Number of unknown nodes localized by neighboring anchors under different N_U and N_A with CPSO.

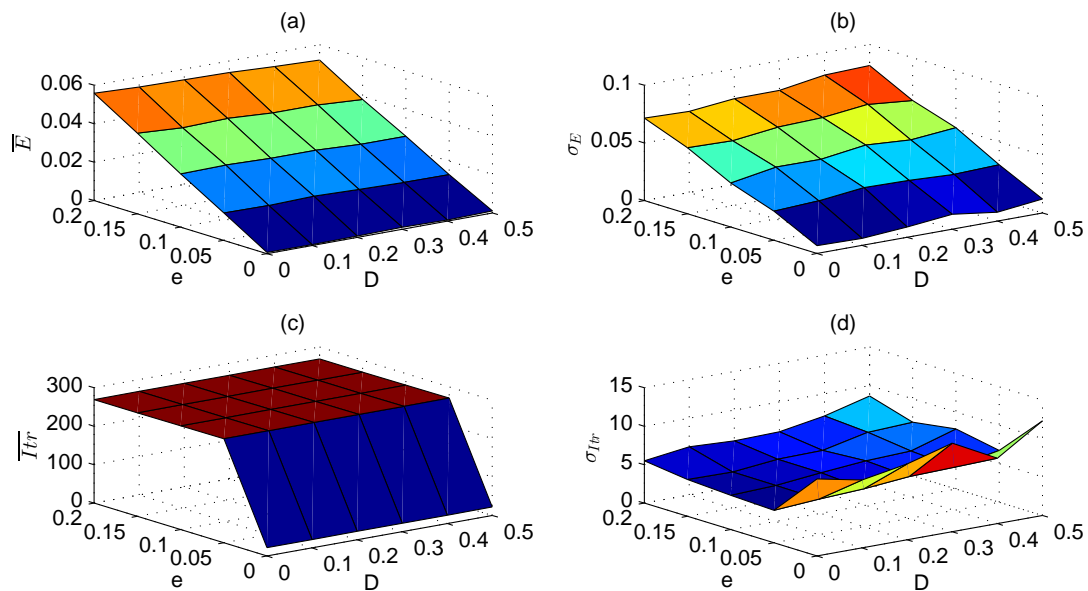
Table 4 denotes that the variations are very small except σ_{Itr} . \bar{E} and σ_E are almost the same, but CPSO and PPSO have the smallest $\bar{I}tr$.

Table 4. Variation range of different N_U and N_A .

| Criteria | WPSO | CPSO | TPSO | PPSO | MPSO | HPSO | EPSO | DPSO | BPSO |
|------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| \overline{Itr} | 311.6–314.9 | 219.1–221.8 | 482.4–485.8 | 220.8–225.6 | 267.8–270.6 | 279.6–285.5 | 304.6–316.6 | 252.2–254.7 | 251.1–252.7 |
| σ_{Itr} | 13.91–19.96 | 3.39–14.92 | 40.64–50.77 | 7.86–20.34 | 9.06–14.57 | 17.94–27.31 | 83.20–90.10 | 0.78–16.83 | 3.38–19.27 |
| \overline{E} | 0.02–0.04 | 0.02–0.04 | 0.02–0.044 | 0.03–0.05 | 0.02–0.05 | 0.02–0.04 | 0.14–0.27 | 0.17–0.23 | 0.06–0.08 |
| σ_E | 0.02–0.09 | 0.02–0.08 | 0.02–0.09 | 0.02–0.08 | 0.03–0.10 | 0.02–0.10 | 0.14–0.41 | 0.10–0.29 | 0.06–0.13 |

From Figure 7a,b, we can see that D impacts the localization error very little, where \overline{E} (σ_E) only differs less than 0.003 (0.006) with the same e and different D . However, e affects the localization error greatly: a larger e leads to a larger localization error. As shown in Figure 7c, \overline{Itr} is minimal when $D = e = 0$, and it increases significantly when both D and e are greater than 0. However, it keeps almost the same when $D > 0$ and $e > 0$, which can also be proved from Table 5. On the contrary, σ_{Itr} reaches maximum when $D = e = 0$, and it decreases when both D and e are greater than 0, as shown in Figure 7d.

Table 5 shows that \overline{E} and σ_E of EPSO, DPSO, and BPSO are larger than the other algorithms, and D and e have little impacts on \overline{Itr} , but σ_{Itr} varies very much under different D and e .

**Figure 7.** Impacts of D and e on localization performance of CPSO. (a) \overline{E} ; (b) σ_E ; (c) \overline{Itr} ; (d) σ_{Itr} .**Table 5.** Variation range of different $D > 0$ and $e > 0$.

| Criteria | WPSO | CPSO | TPSO | PPSO | MPSO | HPSO | EPSO | DPSO | BPSO |
|------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| \overline{Itr} | 359.7–366.1 | 269.2–270.6 | 485.0–486.7 | 271.0–273.2 | 316.6–320.9 | 327.3–335.5 | 308.6–311.4 | 269.3–269.7 | 262.2–262.6 |
| σ_{Itr} | 13.3–17.65 | 4.62–7.82 | 39.99–43.6 | 9.36–13.62 | 8.68–12.79 | 18.65–22.84 | 85.49–86.80 | 0.89–2.79 | 3.47–5.34 |
| \overline{E} | 0.01–0.04 | 0.01–0.04 | 0.01–0.04 | 0.01–0.04 | 0.01–0.04 | 0.01–0.04 | 0.16–0.19 | 0.17–0.19 | 0.16–0.17 |
| σ_E | 0.03–0.07 | 0.03–0.07 | 0.03–0.07 | 0.03–0.07 | 0.03–0.08 | 0.04–0.08 | 0.22–0.26 | 0.17–0.21 | 0.17–0.19 |

Figure 8a,b denote that the larger R has higher localization precision. \overline{E} of EPSO, DPSO, and BPSO are larger than the other algorithms, while the other algorithms have almost the same \overline{E} . Moreover, \overline{E} of EPSO and DPSO decrease with increasing R , while those of the other algorithms show little change. σ_E has the same rule as \overline{E} , as shown in Figure 8b. Figure 8c,d show that the impacts of R on the number of iterations are very little, and CPSO and PPSO requires the fewest iterations, while TPSO needs the most iterations.

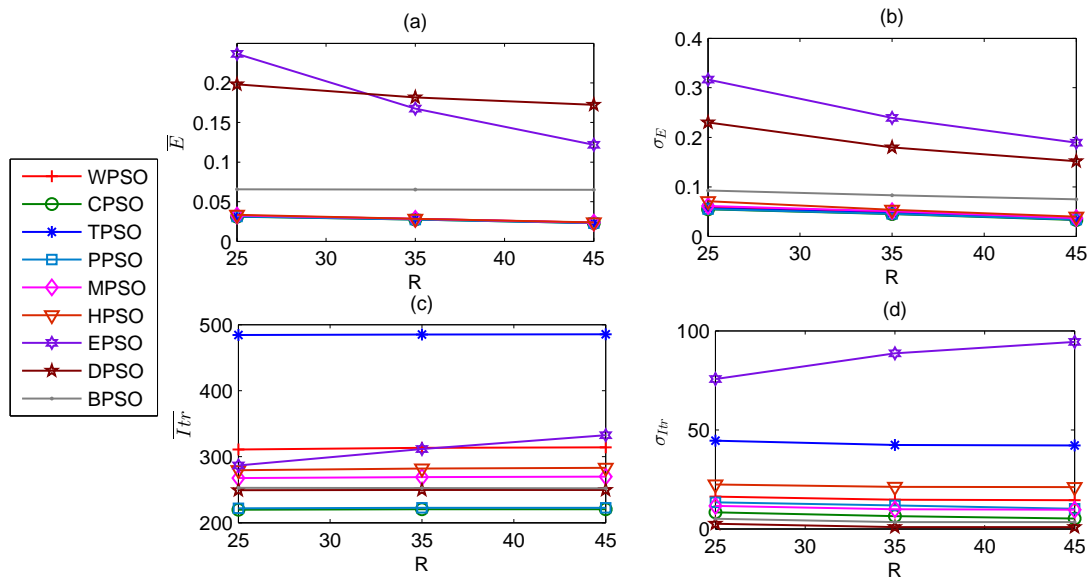


Figure 8. Impacts of R on localization performance. (a) \bar{E} ; (b) σ_E ; (c) \bar{I}_{tr} ; (d) $\sigma_{I_{tr}}$.

5.4. Comparison between CPSO and SOCP

Table 1 shows the comparison between PSO and the other optimization algorithms. We compare CPSO and second-order cone programming (SOCP) in this section, because SOCP is also a popular optimization algorithm used in localization problems [4,5], and CPSO has not been compared with SOCP. The SOCP algorithm is implemented in Matlab by CVX [45].

As shown in Figure 9, CPSO with global-best model outperforms SOCP under different N_U , N_A , D , and e . \bar{E} and σ_E of CPSO are 0.0085–0.0684 and 0.0526–0.1311, respectively, and those of SOCP are 0.2909–0.3549 and 0.2707–0.3, respectively. Further, SOCP takes much longer than CPSO. For example, there are 450 test cases when $N_U = 100$, and SOCP uses 2 hours and 58 minutes to obtain the results, while CPSO only takes 12 minutes.

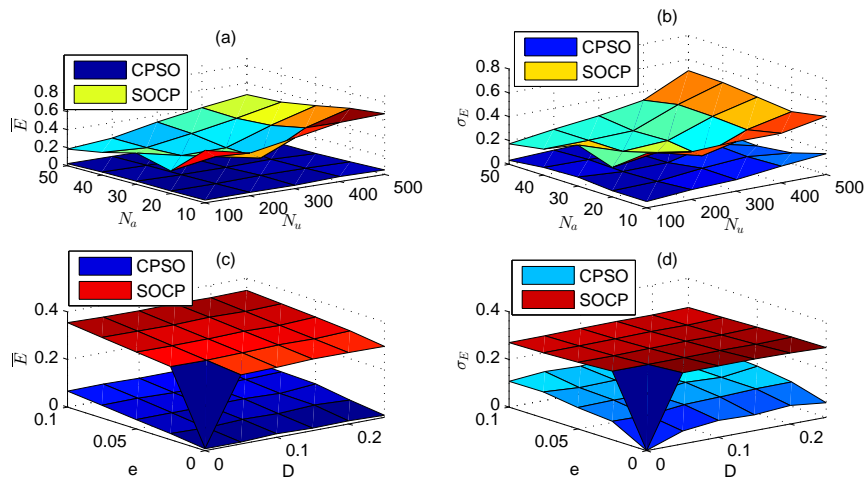


Figure 9. Comparison of CPSO and SOCP. (a) \bar{E} of different N_U and N_A ; (b) σ_E of different N_U and N_A ; (c) \bar{E} of different D and e ; (d) σ_E of different D and e .

6. Conclusions

As a classical swarm intelligence algorithm, particle swarm optimization has many advantages over other optimization algorithms to solve the localization problem of wireless sensor networks, and

many particle swarm optimization-based localization algorithms have been proposed in recent years, but it lacks of parameter selection and comprehensive comparison of these algorithms. This paper surveys the existing particle swarm optimization-based localization algorithms, and chooses the best parameters based on simulations. Further, we compare currently widely-used particle swarm optimization-based localization algorithms with six types of swarm topologies, and the results show that particle swarm optimization with constriction coefficient and ring topology is the best choice to solve the localization algorithm of wireless sensor networks.

Acknowledgments: This research is supported by the Open Research Fund from Shandong provincial Key Laboratory of Computer Networks, Grant No. SDKLCN-2015-03.

Author Contributions: Huanqing Cui and Minglei Shu conceived and designed the experiments; Huanqing Cui and Yinglong Wang performed the experiments; Huanqing Cui analyzed the data; Huanqing Cui and Min Song wrote the paper.

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

References

1. Yick, J.; Mukherjee, B.; Ghosal, D. Wireless sensor network survey. *Comput. Netw.* **2008**, *52*, 2292–2330.
2. Cheng, L.; Wu, C.; Zhang, Y.; Wu, H.; Li, M.; Maple, C. A survey of localization in wireless sensor network. *Int. J. Distrib. Sens. Netw.* **2012**, *2012*, 962523.
3. Han, G.; Xu, H.; Duong, T.Q.; Jiang, J.; Hara, T. Localization algorithms of wireless sensor networks: A survey. *Telecommun. Syst.* **2011**, *48*, 1–18.
4. Naddafzadeh-Shirazi, G.; Shenouda, M.B.; Lampe, L. Second order cone programming for sensor network localization with anchor position uncertainty. *IEEE Trans. Wirel. Commun.* **2014**, *13*, 749–763.
5. Beko, M. Energy-based localization in wireless sensor networks using second-order cone programming relaxation. *Wirel. Pers. Commun.* **2014**, *77*, 1847–1857.
6. Erseghe, T. A distributed and maximum-likelihood sensor network localization algorithm based upon a nonconvex problem formulation. *IEEE Trans. Signal Inf. Process. Netw.* **2015**, *1*, 247–258.
7. Ghari, P.M.; Shahbazian, R.; Ghorashi, S.A. Wireless sensor network localization in harsh environments using SDP relaxation. *IEEE Commun. Lett.* **2016**, *20*, 137–140.
8. Kulkarni, R.V.; Förster, A.; Venayagamoorthy, G.K. Computational intelligence in wireless sensor networks: A survey. *IEEE Commun. Surv. Tutor.* **2011**, *13*, 68–96.
9. Kumar, A.; Khoslay, A.; Sainiz, J.S.; Singh, S. Computational intelligence based algorithm for node localization in wireless sensor networks. In Proceedings of the 2012 6th IEEE International Conference Intelligent Systems (IS), Sofia, Bulgaria, 6–8 September 2012; pp. 431–438.
10. Cheng, J.; Xia, L. An effective Cuckoo search algorithm for node localization in wireless sensor network. *Sensors* **2016**, *16*, 1390.
11. Chuang, P.J.; Jiang, Y.J. Effective neural network-based node localisation scheme for wireless sensor networks. *IET Wirel. Sens. Syst.* **2014**, *4*, 97–103.
12. Kulkarni, R.V.; Venayagamoorthy, G.K.; Cheng, M.X. Bio-inspired node localization in wireless sensor networks. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC 2009), San Antonio, TX, USA, 11–14 October 2009; pp. 205–210.
13. Goyal, S.; Patterh, M.S. Modified bat algorithm for localization of wireless sensor network. *Wirel. Pers. Commun.* **2016**, *86*, 657–670.
14. Kumar, A.; Khoslay, A.; Sainiz, J.S.; Singh, S. Meta-heuristic range based node localization algorithm for wireless sensor networks. In Proceedings of the 2012 International Conference on Localization and GNSS (ICL-GNSS), Starnberg, Germany, 25–27 June 2012; pp. 1–7.
15. Li, Z.; Liu, X.; Duan, X. Comparative research on particle swarm optimization and genetic algorithm. *Comput. Inf. Sci.* **2010**, *3*, 120–127.
16. Jordehi, A.R.; Jasni, J. Parameter selection in particle swarm optimization: A survey. *J. Exp. Theor. Artif. Intell.* **2013**, *25*, 527–542.
17. Kulkarni, R.V.; Venayagamoorthy, G.K. Particle swarm optimization in wireless sensor networks: A brief survey. *IEEE Trans. Syst. Man Cybern. C* **2011**, *41*, 262–267.

18. Gharghan, S.K.; Nordin, R.; Ismail, M.; Ali, J.A. Accurate wireless sensor localization technique based on hybrid PSO-ANN algorithm for indoor and outdoor track cycling. *IEEE Sens. J.* **2016**, *16*, 529–541.
19. Cao, C.; Ni, Q.; Yin, X. Comparison of particle swarm optimization algorithms in wireless sensor network node localization. In Proceedings of the 2014 IEEE International Conference on Systems, Man and Cybernetics (SMC), San Diego, CA, USA, 5–8 October 2014; pp. 252–257.
20. Xu, L.; Zhang, H.; Shi, W. Mobile anchor assisted node localization in sensor networks based on particle swarm optimization. In Proceedings of the 2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), Chengdu, China, 23–25 September 2010; pp. 1–5.
21. Han, W.; Yang, P.; Ren, H.; Sun, J. Comparison study of several kinds of inertia weights for PSO. In Proceedings of the 2010 IEEE International Conference on Progress in Informatics and Computing (PIC), Shanghai, China, 10–12 December 2010; pp. 280–284.
22. Valle, Y.D.; Venayagamoorthy, G.K.; Mohagheghi, S.; Hernandez, J.C.; Harley, R.G. Particle swarm optimization: Basic concepts, variants and applications in power systems. *IEEE Trans. Evol. Comput.* **2008**, *12*, 171–195.
23. Medina, A.J.R.; Pulido, G.T.; Ramírez-Torres, J.G. A comparative study of neighborhood topologies for particle swarm optimizers. In Proceeding of the International Joint Conference Computational Intelligence, Funchal, Madeira, Portugal, 5–7 October, 2009; pp. 152–159.
24. Clerc, M. Back to Random Topology. Available online: http://clerc.maurice.free.fr/psa/random_topology.pdf (accessed on 28 February 2017).
25. Gopakumar, A.; Jacob, L. Localization in wireless sensor networks using particle swarm optimization. In Proceeding of the IET International Conference on Wireless, Mobile and Multimedia Networks, Beijing, China, 11–12 January 2008 ; pp. 227–230.
26. Gao, W.; Kamath, G.; Veeramachaneni, K.; Osadciw, L. A particle swarm optimization based multilateration algorithm for UWB sensor network. In Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECE '09), Budapest, Hungary, 3–6 May 2009; pp. 950–953.
27. Okamoto, E.; Horiba, M.; Nakashima, K.; Shinohara, T.; Matsumura, K. Particle swarm optimization-based low-complexity three-dimensional UWB localization scheme. In Proceedings of the 2014 Sixth International Conf on Ubiquitous and Future Networks (ICUFN), Shanghai, China, 8–11 July 2014; pp. 120–124.
28. Chuang, P.J.; Wu, C.P. Employing PSO to enhance RSS range-based node localization for wireless sensor networks. *J. Inf. Sci. Eng.* **2011**, *27*, 1597–1611.
29. Liu, Z.; Liu, Z. Node self-localization algorithm for wireless sensor networks based on modified particle swarm optimization. In Proceedings of the 2015 27th Chinese Control and Decision Conference (CCDC), Qingdao, China, 23–25 May 2015; pp. 5968–5971.
30. Mansoor-ul-haque, F.A.K.; Iftikhar, M. Optimized energy-efficient iterative distributed localization for wireless sensor networks. In Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Manchester, UK, 13–16 October 2013; pp. 1407–1412.
31. Wei, N.; Guo, Q.; Shu, M.; Lv, J.; Yang, M. Three-dimensional localization algorithm of wireless sensor networks base on particle swarm optimization. *J. China Univ. Posts Telecommun.* **2012**, *19*, 7–12.
32. Dong, E.; Chai, Y.; Liu, X. A novel three-dimensional localization algorithm for wireless sensor networks based on particle swarm optimization. In Proceedings of the 2011 18th International Conference on Telecommunications (ICT), Ayia Napa, Cyprus, 8–11 May 2011; pp. 55–60.
33. Zhang, Y.; Liang, J.; Jiang, S.; Chen, W. A localization method for underwater wireless sensor networks based on mobility prediction and particle swarm optimization algorithms. *Sensors* **2016**, *16*, 212.
34. Zhang, X.; Wang, T.; Fang, J. A node localization approach using particle swarm optimization in wireless sensor networks. In Proceedings of the 2014 International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI), Beijing, China, 17–18 October 2014; pp. 84–87.
35. Kulkarni, R.V.; Venayagamoorthy, G.K. Bio-inspired algorithms for autonomous deployment and localization of sensor nodes. *IEEE Trans. Syst., Man, Cybern. C* **2010**, *40*, 663–675.
36. Clerc, M.; Kennedy, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **2002**, *6*, 58–73.
37. Monica, S.; Ferrari, G. Swarm intelligent approaches to auto-localization of nodes in static UWB networks. *Appl. Soft Comput.* **2014**, *25*, 426–434.

38. Zhang, Q.; Cheng, M. A node localization algorithm for wireless sensor network based on improved particle swarm optimization. *Lect. Note Elect. Eng.* **2014**, *237*, 135–144.
39. Zain, I.F.M.; Shin, S.Y. Distributed localization for wireless sensor networks using binary particle swarm optimization (BPSO). In Proceedings of the 2014 IEEE 79th Vehicular Technology Conference (VTC Spring), Seoul, Korea, 18–21 May 2014; pp. 1–5.
40. Ma, G.; Zhou, W.; Chang, X. A novel particle swarm optimization algorithm based on particle migration. *Appl. Math. Comput.* **2012**, *218*, 6620–6626.
41. Guo, H.; Low, K.S; Nguyen, H.A. Optimizing the localization of a wireless sensor network in real time based on a low-cost microcontroller. *IEEE Trans. Ind. Electron.* **2011**, *58*, 741–749.
42. Namin, P.H.; Tinati, M.A. Node localization using particle swarm optimization. In Proceedings of the 2011 Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Adelaide, Australia, 6–9 December 2011; pp. 288–293.
43. Liu, Q.; Wei, W.; Yuan, H.; Zha, Z.; Li, Y. Topology selection for particle swarm optimization. *Inf. Sci.* **2016**, *363*, 154–173.
44. Bao, H.; Zhang, B.; Li, C.; Yao, Z. Mobile anchor assisted particle swarm optimization (PSO) based localization algorithms for wireless sensor networks. *Wirel. Commun. Mob. Comput.* **2012**, *12*, 1313–1325.
45. CVX Research. CVX: Matlab Software for Disciplined Convex Programming. Available online: <http://cvxr.com/cvx/> (accessed on 28 February 2017).



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).