

Article

An Effective Cuckoo Search Algorithm for Node Localization in Wireless Sensor Network

Jing Cheng and Linyuan Xia *

Guangdong Key Laboratory for Urbanization and Geo-simulation, School of Geography and Planning, Sun Yat-Sen University, Guangzhou 510275, China; chengjing921@126.com

* Correspondence: xialiny@mail.sysu.edu.cn; Tel./Fax: +86-20-8411-5833

Academic Editor: Leonhard M. Reindl

Received: 6 July 2016; Accepted: 10 August 2016; Published: 31 August 2016

Abstract: Localization is an essential requirement in the increasing prevalence of wireless sensor network (WSN) applications. Reducing the computational complexity, communication overhead in WSN localization is of paramount importance in order to prolong the lifetime of the energy-limited sensor nodes and improve localization performance. This paper proposes an effective Cuckoo Search (CS) algorithm for node localization. Based on the modification of step size, this approach enables the population to approach global optimal solution rapidly, and the fitness of each solution is employed to build mutation probability for avoiding local convergence. Further, the approach restricts the population in the certain range so that it can prevent the energy consumption caused by insignificant search. Extensive experiments were conducted to study the effects of parameters like anchor density, node density and communication range on the proposed algorithm with respect to average localization error and localization success ratio. In addition, a comparative study was conducted to realize the same localization task using the same network deployment. Experimental results prove that the proposed CS algorithm can not only increase convergence rate but also reduce average localization error compared with standard CS algorithm and Particle Swarm Optimization (PSO) algorithm.

Keywords: wireless sensor network; localization; cuckoo search algorithm; average localization error; convergence rate

1. Introduction

A wireless sensor network (WSN) is a self-organization network composed of a large number of small-size, low-cost sensor nodes which can monitor physical or environmental condition [1]. Recent advances of micro-electro-mechanical systems (MEMS) technology and wireless communication have propelled WSN applied to a variety of fields such as health monitoring [2], transportation management [3], business and home automation [4], global-scale wildlife [5], forest fire and environmental monitoring [6,7]. In almost all the above applications, on no account can we ignore the value of location information of sensor nodes, because it is extremely hard to distinguish or utilize the monitored information without location information. Besides that, geographic routing can save significant energy by eliminating the need for route discovery [8]. Therefore, it is highly significant to locate sensor nodes accurately.

Traditionally, each sensor node can be localized by using the BeiDou Navigation Satellite System (BDS) or Global Positioning System (GPS). However, practical considerations such as cost, power consumption and the volume of BDS/GPS receivers make it impossible for the use of BDS/GPS on each sensor node, especially for wide-range WSN. For example, sensor nodes are typically powered by small batteries, which generally cannot be easily changed or recharged. In addition, WSN is commonly

deployed in the urban canyon or indoors where the transmission of the satellite signals is adversely blocked [9].

In recent years, more and more researchers tried to overcome the drawbacks by proposing various localization methods. Generally WSN node localization process can be illustrated using Figure 1. In a two-dimensional (2-D) field, it is assumed that the unknown node's coordinate is (x, y) which needs to be positioned, anchor node's coordinates are (x_1, y_1) , (x_2, y_2) , (x_3, y_3) . The unknown node will be localizable if there are three (or more) anchors, which have prior knowledge of their coordinates within its communication range.

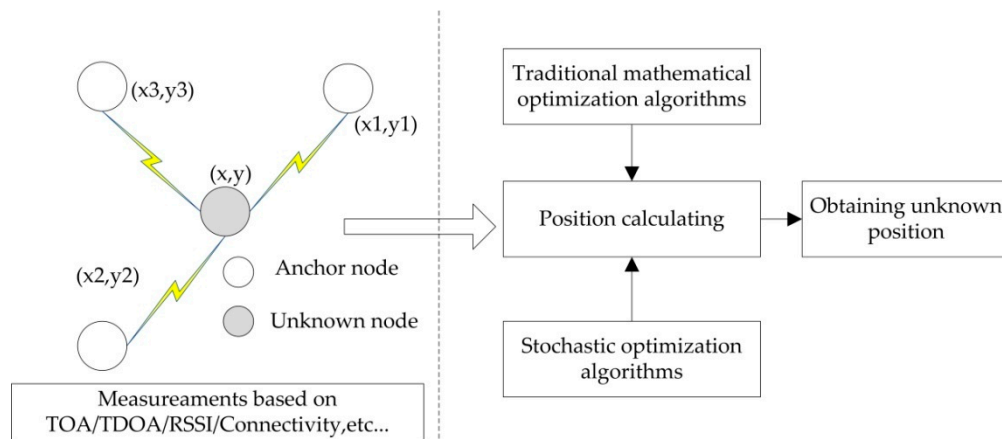


Figure 1. Node localization process in WSN.

As presented in Figure 1, the localization process consists of two stages: measurements based on inter-sensor distance, angle measurements or connectivity, such as Time of Arrival (TOA), Time Difference of Arrival (TDOA), Angle of Arrival (AOA), Received Signal Strength Indication (RSSI), etc. [10]; then calculating position of unknown nodes whose coordinates are unknown either by traditional mathematical optimization methods solving a set of simultaneous equations, or by using stochastic optimization algorithms that minimize localization error. Whatever measuring technology is adopted, in actual scenario measuring data is always disturbed by some noise, which will make localization results dissatisfactory [11]. Hence a lot of literatures focus on overcoming the problem.

A detailed survey of node localization in WSN has been reported in the literature [12,13]. Niculescu et al. [14] proposed the efficient Ad Hoc Positioning System (APS) that extends the capabilities of GPS to non-GPS enabled nodes in a hop by hop fashion in an ad hoc network where only a limited fraction of nodes have self-location capability. Anchors flood their location information to all the neighboring nodes and each unknown node estimates its own location by performing distance measurement from three or more anchors. To prevent error accumulation, Rabaey et al. [15] introduced a two-phase approach using connectivity between nodes for initial position estimates and only the distances to one-hop neighbors are considered for position refinement. This approach reduces insensitivity to range errors on some extent. Doherty et al. [16] reported an efficient second-order cone method to solve the localization problem by relaxing the proximity constraints between nodes. This method requires anchors deployed around the perimeter of the network, otherwise the position estimation of unknown nodes will tend to interior, yielding highly inaccurate results. Based on [16], other researchers [17] studied a semi-definite programming (SDP) relaxation method to solve sensor network localization problem using incomplete and inaccurate distance information. This method performs well even in highly noisy environments. In 2014, Simonetto et al. [18] presented a class of convex relaxations based on a maximum likelihood (ML) formulation, which derived a computational efficient edge-based version of this ML convex relaxation class and designed a distributed algorithm that enable sensor nodes to solve these edge-based convex programs locally by communicating only with their close neighbors. Employing this version of convex relaxations to message the original

non-convex formulation can offer a powerful handle on computing accurate solutions. Shang et al. [19] introduced a localization algorithm MDS-MAP (P), which worked in a distributed mode by using relative maps. This algorithm keeps better performance than basic MDS-MAP algorithm under the uniform layouts or irregularly-shaped networks, particularly when the number of anchors is small. However, it needs a high consumption of battery power for each sensor to construct relative maps. Although these traditional mathematical optimization techniques work well, they require enormous computational efforts and communication overhead, which grow exponentially as the problem size increases.

In recent years meta-heuristic algorithms gradually play an important role in engineering optimization [20,21]. This is attributed to the fact that they require moderate memory and computational efforts and more importantly can provide better results in comparison with traditional mathematical optimization methods. WSN node localization is often formulated as a multi-dimensional optimization problem. A survey of bio-inspired node localization can be found in [22,23]. To meet the localization requirement of large scale WSN, Yun et al. [24] reported two intelligent range-free WSN localization schemes, which utilize received signal strength (RSS) from the anchor nodes. First, the edge weight of each anchor node is considered to be separately and combined to compute the location of sensor nodes. The edge weights are modeled by the fuzzy logic system (FLS) and optimized by the genetic algorithm (GA). Second, localization is considered as a single problem and the entire sensor location mapping from the anchor node signals is approximated by a neural network (NN). To address the flip ambiguity problem, Kannan et al. [25] introduced a two phase simulated annealing (SA) based localization algorithm, which first obtains an accurate estimate of location, then if some nodes have flip ambiguity problem, optimization is performed based on neighborhood information of nodes and moved to the correct position. As a promising population-based optimization algorithm, Particle Swarm Optimization (PSO) algorithm was first developed by James and Russell [26] in 1995 and has been successfully applied in various fields. For example, Chih [27] presented extensive literature review on PSO and introduced a novel self-adaptive check and repair operator (SACRO) based on BPSO-TVAC and CBPSO-TVAC [28] to solve the multidimensional knapsack problem. Experimental results show that the proposed algorithm is more competitive and robust than the traditional CRO. To improve location accuracy, Gopakumar et al. [29] devised a WSN localization scheme using Particle Swarm Optimization (PSO) algorithm. However, it is likely to get trapped in local optimal that leads to pre-matured convergence. Biogeography based optimization (BBO) and its variants [30], hybrid algorithms such as bacterial foraging approach (BFA) and GA [31], PSO and BFA [32], PSO and DE [33], GA and SA [34], have been also developed in order to improve localization accuracy or convergence speed. However, the implementation of these techniques involves additional computational overheads.

In 2009, Cuckoo Search (CS) as a highly potential meta-heuristic algorithm was proposed by Xin-She Yang of Cambridge University and Suash Deb of C.V. Raman College of Engineering. Recent work [35,36] has found that the CS algorithm is dominant in minimizing localization error due to the characteristics of few parameters, being easy to realize, far more efficient global search ability than existing algorithms such as GA and PSO [37]. However, the CS algorithm exhibits slow convergence rate, which makes it require more resource to achieve the certain accuracy.

Motivated by the above observations, in this paper, we propose a new modified CS algorithm to obtain positioning results with high precision and rapid convergence. The proposed CS algorithm adopts the modified step size to enable the population to approach global optimal solution rapidly. In addition, to enhance population diversity and thereby avoid local convergence, the fitness of each solution is employed to build mutation probability that determines the chance of it being found and replaced by new random solution. Finally, the modified CS algorithm restricts the population in the certain range so that it can prevent the energy consumption caused by insignificant search. The modified CS algorithm is studied on the effects of parameters like anchor density, node density and communication range with respect to average localization error and localization success ratio.

Experimental results prove that the modified CS algorithm can not only increase convergence rate but also reduce average localization error compared with standard CS algorithm and PSO algorithm.

The rest of the paper is organized as follows: Section 2 provides a description of standard CS algorithm; Section 3 gives an introduction about the modified CS algorithm; WSN node localization process based on the modified CS approach is explained in Section 4; simulation results and localization performance analysis are given in Section 5; Section 6 gives the conclusion of the paper.

2. Standard Cuckoo Search Algorithm

Cuckoo Search (called CS), as one of the latest nature-inspired search algorithm, was proposed in 2009 by Xin-She Yang of Cambridge University and Suash Deb of C.V. Raman College of Engineering. CS is based on the brood parasitism of some cuckoo species. To apply CS as an optimization tool, Yang and Deb described the CS by three ideal rules [38]:

- (1) Each cuckoo lays one egg at a time, and dumps its egg in a randomly chosen nest;
- (2) The best nests with high-quality eggs will be carried over to the next generations;
- (3) The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability $Pa \in [0, 1]$. In this case, the host bird can either get rid of the egg, or simply abandon the nest and build a completely new nest.

As a further approximation, this last assumption can be approximated by a fraction Pa of the n host nests are replaced by new nests (new random solutions).

For the implementation point of view, we can generate new solutions (cuckoos) by two methods: *Le'vy* flight and a fraction Pa of the n host nests are replaced by new random solutions.

Firstly, the *Le'vy* flight process, which has previously been used in search algorithms, is a random walk that is characterized by a series of instantaneous jumps chosen from a probability density function which has a power law tail. When generating new solutions $x^{(t+1)}$ for, say, a cuckoo i , a *Le'vy* flight is performed by using Equation (1):

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus Le'vy(\lambda) \quad (1)$$

where $\alpha > 0$ is the step size which should be related to the scales of the problem of interests. In most cases, $\alpha = (L/10)$ where L is the characteristic scale of the problem of interest. The above Equation (1) is essentially the stochastic equation for a random walk. In general, a random walk is a Markov chain whose next status/location only depends on the current location (the first term in the above Equation (1)) and the transition probability (the second term in the above Equation (1)). The product \oplus means entry-wise multiplications. This entry-wise product is similar to those used in PSO, but here the random walk via *Le'vy* flight is more efficient in exploring the search space, as its step length is much longer in the long run.

The step length of random walk formed by *Le'vy* flight obeys the *Le'vy* distribution which has an infinite variance with an infinite mean as follows:

$$Le'vy \sim u = t^{-\lambda}, (1 < \lambda \leq 3) \quad (2)$$

where λ is a constant, and the step length can be calculated mathematically using the following equation:

$$Le'vy(\lambda) = \frac{u}{|v|^{1/\lambda}} \quad (3)$$

where u and v are drawn from normal distributions as followings:

$$u \sim N(0, \sigma_u^2), v \sim N(0, \sigma_v^2) \quad (4)$$

where:

$$\sigma_u = \left\{ \frac{\Gamma(1+\lambda)\sin(\pi\lambda/2)}{\Gamma[(1+\lambda)/2]\lambda 2^{(\lambda-1)/2}} \right\}^{1/\lambda}, \quad \sigma_v = 1 \quad (5)$$

Briefly speaking, the step length essentially forms a random walk process with a power-law step-length distribution with a heavy tail. Some of the new solutions should be generated by *Le'vy* walk around the best solution obtained so far, thus this will speed up the local search.

In addition, a fraction Pa of the solutions will be replaced by new random solutions whose locations should be far enough from the current best solution. This will make sure that the system will not be trapped in a local optimum.

3. Modified Cuckoo Search (CS) Algorithm

In standard CS algorithm, the efficiency of searching global best solution is mainly guided by step size control factor α and mutation probability Pa . Because the parameters are kept constant, i.e., $\alpha = 0.01$, $Pa = 0.25$, this can result in decreasing location precision which will affect the practical value of WSN applications and reducing convergence rate which will shorten the life time of sensor nodes with limited energy resource because of a large number of iterations and long computational time. Although some improvement schemes based on standard CS algorithm have been proposed to meet specific requirements, but they are not greatly significant for WSN localization [39,40]. Therefore, to solve the current situation, this paper tries to take strategies from the below aspects discussed to propose an effective CS algorithm for WSN node localization.

One aspect is the modification about step size α . As is shown in *Le'vy* flight Equation (1) in Section 2, if the value of α is set to be too small, the population in the search procedure will focus on a small field around the current local best solution found in the past search. In other words, it is insignificant for generating diverse solutions to explore the search space on the global scale. In the end, global best solution cannot be found unless there is a large number of iterations which may result in relatively high computational consumption for energy-limited WSN. If the value of α is set to be too large, the according step length will make the new solutions generated too far away from the current local best solution. Finally, it is not guaranteed to converge to global optimal solution. Consequently, selecting a proper step size is very important to find the global best solution as efficiently as possible.

Based on the above analysis, we propose that step size α is modified using the Equation (6).

$$\alpha = \alpha_{\max} - (N_{\text{iter}}/N_{\text{itertotal}}) \times (\alpha_{\max} - \alpha_{\min}) \quad (6)$$

where α_{\max} and α_{\min} denote the maximum and minimum of step size, respectively; N_{iter} and $N_{\text{itertotal}}$ denote the current iteration number and total number of iterations, respectively.

As expressed in the Equation (6), step size α decreases with the increasing of iteration numbers. At the beginning of iterations, step size takes the maximum which can promote global search, then local search is intensified gradually by decreasing step size. The reason for this modification is that the population tends to global optimal solution gradually with the increasing of iterations.

The other aspect, we employ the fitness of solutions to build the mutation probability Pa which can enhance population diversity and avoid local convergence. If mutation probability Pa is set to be too large, the convergence rate cannot be accelerated in time; if mutation probability Pa is set to be too small, then the accuracy decreases and a large number of iterations are caused in order to obtain good performance. Therefore, the setting of parameter Pa should be neither too large nor too small.

According to the standard CS algorithm, mutation probability Pa is associated with the fitness of the solution, which can simply be proportional to the value of an objective function. A nest having higher fitness value which is closer to current global optimum than the other nest that is far away, has a higher probability to be chosen as the member of populations of next generation, in other words, it can be more hard to be found and replaced by new random solution, that is, the value of mutation probability Pa can be relatively smaller, and vice versa.

The above analysis is taken as the basis for the following adjustment about mutation probability Pa :

$$Pa(j) = \begin{cases} Pa_{\min} + (Pa_{\max} - Pa_{\min}) \times K, K < 1 \\ \frac{Pa_{\max}}{N_iter}, otherwise \end{cases} \quad j = 1, 2, 3, \dots, n \quad (7)$$

where $K = fitness(j) - f_{\min}$, which depends on the current quality of j th solution; $fitness(j)$ and f_{\min} represent the current fitness of j th solution and current global optimal fitness of the population, respectively; Pa_{\max} and Pa_{\min} represent the maximum and minimum of mutation probability Pa , respectively.

From the Equation (7), it can be seen that the fitness of the solution is adopted to adjust Pa . If $K \leq 1$, it means that the current quality for j th solution is close to current global optimum. It is suggested that mutation probability Pa should be proportional to K which represents the current quality of j th solution. The higher the quality is, the smaller mutation probability Pa is. In contrast, if $K > 1$, it means that the current quality of j th solution is far away from current global optimum. We propose that mutation probability Pa vary inversely with iteration times because the population will approach the global optimum as the iteration number increases.

Besides that, the population is restricted in the certain range so that it can prevent the energy consumption caused by meaningless search. That is to say, when the solutions are outside the range, we use the range maximum or minimum to replace them. A modified CS algorithm is proposed herein relying on the above several aspects to find the global optimum as efficiently as possible. The detailed steps about realization of the modified CS algorithm are described in Algorithm 1.

Algorithm 1. Modified CS algorithm

1. Begin
 2. Generate initial population of n nests (solutions) $x_i, i = 1, 2, \dots, n$
 3. Define objective function $f(x); x = (x_1, x_2, \dots, x_d)$
 4. Set the range of α and $Pa: \alpha_{\min}, \alpha_{\max}, Pa_{\min}, Pa_{\max}$
 5. Set the range of the nest(solution): X_{\min}, X_{\max}
 6. Set the maximum number of iterations: $N_itertotal$
 7. For all x_i do
 8. Calculate the fitness $F_i = f(x_i)$
 9. End For
 10. $N_iter = 1$
 11. While ($N_iter < N_itertotal$) do
 12. For all x_i do
 13. Compute the step size for *Le'vy* flight using Equation (6)
 14. Generate a new cuckoo (x_j) from the nest x_i randomly by taking Lévy flight
 15. If ($x_j > X_{\max}$) then
 16. $x_j \leftarrow X_{\max}$
 17. End If
 18. If ($x_j < X_{\min}$) then
 19. $x_j \leftarrow X_{\min}$
 20. End If
 21. Calculate the fitness $F_j = f(x_j)$
 22. Choose a random nest (x_k) among n nest randomly
 23. If ($F_j > F_k$) then
 24. $x_j \leftarrow x_k$
 25. $F_j \leftarrow F_k$
-

Algorithm 1. *Cont.*

26. End If
 27. End For
 28. Keep the current global optimal fitness: f_{\min}
 29. Compute the probability Pa using Equation (7)
 30. A fraction (Pa) of worse nests abandoned and new ones/solutions are built/generated correspondingly
 31. For all the nests (say, x_i) to be built/generated do
 32. If ($x_i > X_{\max}$) then
 33. $x_i \leftarrow X_{\max}$
 34. End If
 35. If ($x_i < X_{\min}$) then
 36. $x_i \leftarrow X_{\min}$
 37. End If
 38. Calculate the fitness $F_i = f(x_i)$ and evaluate its quality/fitness F_i
 39. Keep best solutions (or nests with quality solutions)
 40. End For
 41. Rank all the solutions and find the current best
 42. End While
 43. End
-

4. WSN Node Localization Process Based on the Modified CS Algorithm

The objective of WSN localization is to estimate the coordinates of N unknown nodes based on M anchor nodes. Assuming that all sensor nodes are deployed in a two-dimensional sensor field. The WSN node localization process based on modified CS algorithm is shown in the Figure 2.

As presented in Figure 2, estimating the locations of unknown nodes mainly includes the following steps:

- Step 1: M anchor nodes and N unknown nodes are randomly deployed in a sensor field. The communication range for each sensor node is set to R.
- Step 2: Anchor nodes broadcast their locations frequently.
- Step 3: If i th unknown node has three or more than three anchors within communication range, it will be considered to be localizable. Owing to RSSI (received signal strength indicator) of simple implementation and low cost in hardware under actual deployment scenario, in this paper, we assume that distance measurement between neighboring nodes is realized based on RSSI that can then be transferred into equivalent distances for positioning intersection. However, RSSI-based ranging is usually affected by multi-path and obstacles blocking, which can be modeled as log-normal shadowing. The result of the log-normal model is that RSSI-based distance estimates have ranging error which follows a zero-mean Gaussian distribution with variance σ^2 . In addition, the standard deviation of ranging error is proportional to the actual distance d_{ij} between node (x_i, y_i) and (x_j, y_j) [8,10], as shown in the Equation (8).

$$\sigma^2 = \gamma^2 \times d_{ij}^2 \quad (8)$$

where noise factor γ is set to 0.1 in the experiment, d_{ij} is the real distance between the unknown node (x_i, y_i) and j th anchor (x_j, y_j) within communication range, such as the Equation (9).

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (9)$$

The measured distance d_{ij}' between unknown node (x_i, y_i) and j th anchor node (x_j, y_j) is modeled using Equation (10).

$$d_{ij}' = d_{ij} + N_{ij} \quad (10)$$

where N_{ij} represents ranging error between unknown node (x_i, y_i) and its neighboring anchor node (x_j, y_j) . Meanwhile in this paper, we assume that the random ranging error N_{ij} different from each other, that is to say, if $(i, j) \neq (k, p)$, $N_{ij} \neq N_{kp}$.

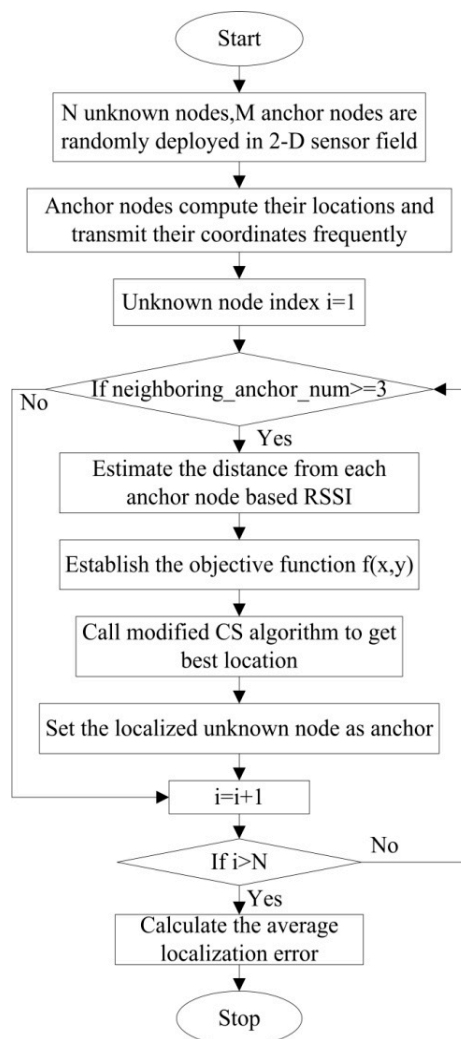


Figure 2. Flowchart of WSN node localization process based on modified CS algorithm.

Step 4: Establishing the objective function $f(x_i, y_i)$. The objective function representing mean of square of ranging error between the unknown node and anchors, is defined as Equation (11):

$$f(x_i, y_i) = \frac{1}{m} \sum_{j=1}^m (d_{ij} - d_{ij}')^2 \quad (11)$$

where $m(m \geq 3)$ is the number of anchor nodes within communication range, the unknown node can estimate its coordinate by running the modified CS algorithm. That is, when the objective function is minimized, corresponding (x_i, y_i) is the position of i th unknown node.

Step 5: The unknown nodes that get localized will act as anchors in the next iteration, thus the number of anchors increase along with the iteration progress.

- Step 6: Step 2~Step 5 are conducted repeatedly until no unknown nodes can be localized or termination conditions are reached.
- Step 7: Computing the average localization error. The average localization error is defined as the average Euclidean distance between the real and estimated locations of sensor nodes, thus average localization error can be calculated via the following Equation (12).

$$\text{Average localization error} = \frac{\sum_{i=1}^{NL} \sqrt{(X_i - x_i)^2 + (Y_i - y_i)^2}}{NL} \quad (12)$$

where NL is the number of localized node, (x_i, y_i) is the computed node location and (X_i, Y_i) is the actual node location. The smaller average localization error, the better the localization performance.

5. Simulation Experiments and Performance Evaluation

In this section, all the simulation experiments are carried out by using MATLAB. The performance of the modified CS algorithm is assessed from the effects of several factors, such as anchor density, communication range, node density in terms of average localization error, localization success ratio and a comparative study with standard CS algorithm and PSO algorithm is conducted to realize the same localization task using the same network deployment. Average localization error is computed using Equation (12) in Section 4.

Localization success ratio, which is defined as the number of unknown nodes which successfully acquire their locations over the total number of sensor nodes whose locations is unknown, is calculated using the following Equation (13). The higher localization success ratio is, the better the localization performance is.

$$\text{Localization success ratio} = \left(\frac{\text{Number of unknown nodes localized}}{\text{Total number of unknown nodes}} \right) \times 100\% \quad (13)$$

5.1. Simulation Setup

All the sensor nodes are randomly deployed in $100 \times 100 \text{ m}^2$ two-dimensional sensor area using a continuous uniform distribution pseudo-random generator. The total number of sensor nodes, namely node density, is set to be 100, 200, 300 and 400. Anchor nodes account for 10%~60% of all sensor nodes. Anchor nodes are deployed randomly. It is assumed that there is no localization error for anchor nodes. Communication range for all sensor nodes is set to be 10 m~50 m. To eliminate the effects of randomness of topology generation and bio-inspired algorithm, each data point is averaged over 10 different test network and each result for a kind of network topology is averaged by running 30 times repeatedly. Extensive experiments have been conducted to design the parameters in the proposed CS algorithm and Experimental results demonstrate that higher localization precision with less iterations can be achieved when the parameters setting of the proposed CS algorithm are as follows, α_{\min} and α_{\max} are set to 0.9 and 1.0 respectively, Pa_{\min} and Pa_{\max} are set to 0.05 and 0.25 respectively. To prevent the search length over the sensor field, the boundary of sensor field is set to be the range of each nest. Thus, in $100 \times 100 \text{ m}^2$ two-dimensional sensor area, X_{\min} and X_{\max} are set to 0 and 100 respectively. As standard CS algorithm, the value of nest number is 25. The iteration threshold for each algorithm is 100 times.

5.2. Experimental Results and Performance Analysis

5.2.1. The Effect of Anchor Density

Anchor density is an important parameter affecting the localization performance and cost for WSN. In this subsection, the effects of anchor density on localization performance are evaluated.

Anchor ratio is set to be 10%, 20%, 30%, 40%, 50% and 60% of all sensor nodes. The communication range of each sensor node is 25 m.

In Figure 3, using the proposed CS algorithm, the average localization error and confidence interval of location error are evaluated by varying anchor ratio in the network for different node density. The confidence interval represents a range estimate of average localization error. It can be clearly observed that when the anchor ratio in the network increases from 10% to 40%, the localization accuracy improves significantly. Because with the increasing number of anchor nodes, the number of unknown nodes that can realize localization based on original anchor nodes (the nodes are localized by BDS/GPS receivers or manual deployment) within their communication range increases. However, when anchor ratio continues to expand, the effects on average localization error become insignificant. It means that sometimes there is no need to increase the number of anchor nodes requiring the extra specific hardware which may be expensive. In addition, as node density increases, average localization error decreases correspondingly. This is because when node density increases, the number of anchor nodes within communication range increases accordingly.

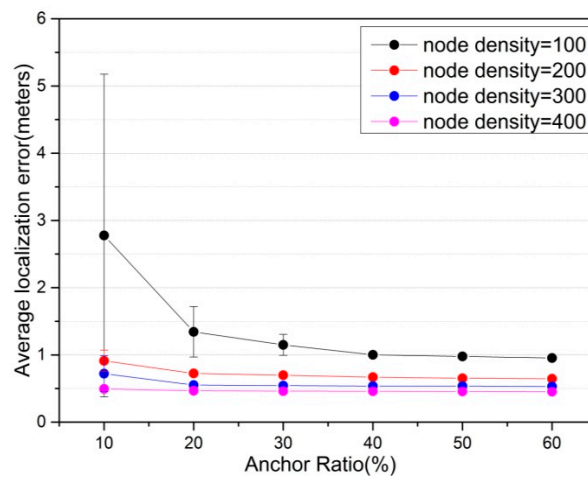


Figure 3. The effect of anchor ratio on average localization error. In addition, error bar represents 95% confidence interval of average localization error.

The simulation results in Figure 3 show that the confidence interval is widened obviously when anchor ratio is 10% and node density is 100. This may be attributed to the fact that there are very few original anchors within communication range for many unknown nodes and error propagation occurs between sensor nodes, which has a negative impact on the average localization error.

Figure 4 illustrates the variation of localization success ratio under different anchor ratio and node density in the network. It demonstrates the fact that when anchor ratio increases, localization success ratio also increases. In addition, it is noted that 100% localization success ratio is achieved under the certain anchor ratio. In addition, expanding the node density improves localization success ratio correspondingly. The simulation results above show that it is necessary to deploy proper number of anchor nodes and node density for the efficient localization accuracy and localization success ratio.

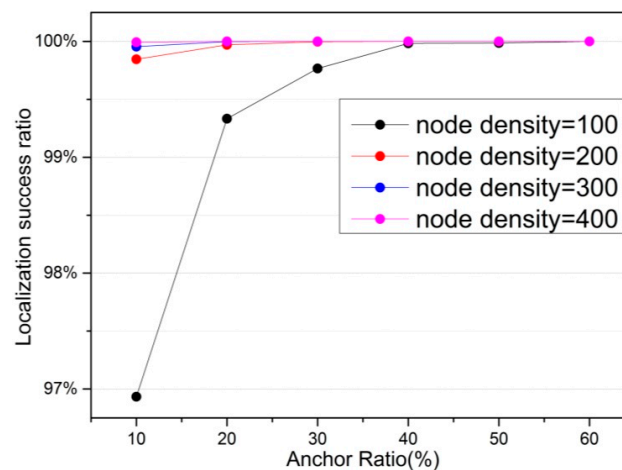


Figure 4. The effect of anchor ratio on localization success ratio.

5.2.2. The Effect of Communication Range

Communication range is another important parameter determining localization performance and energy consumption of sensor nodes. The impact of communication range on the proposed CS algorithm in terms of average localization error under different node density is shown in Figure 5. The number of anchor nodes is set as 20% of all sensor nodes. The variation of communication range is from 10 m to 50 m. From Figure 5, we can observe that if the communication range is smaller than 20 m, the average localization error is slightly larger. This is because the network is not connected for many nodes, such as the example of network topology in Figure 6. As communication range increases, average localization error decreases greatly. This is due to the fact that there is more anchor information available for computing the location of unknown nodes. Even so, when communication range increases to a certain value, average localization error drops slightly. The localization accuracy also depends on the node density that is introduced in the above. It can be observed that the average localization error is decreased gradually with respect to the increasing of node density. This is due to the fact that when node density increases, the network connectivity between sensor nodes becomes high and the number of anchor nodes available within communication range increases, as the example presented in Figure 6.

Figure 7 presents the variation of localization success ratio under the impact of communication range for different node density. In Figure 7, it is remarkable that when the communication range is smallest (that is 10 m) and node density is 100, localization success ratio is around 20.8%. This is because there are few anchors for many unknown nodes within the communication range, and the network connectivity between sensor nodes is poor as the example illustrated in Figure 6a. When the communication range increases, localization success ratio is improved evidently, because as mentioned above, there are more anchor information available in communication range. Therefore it is more easily to locate the unknown nodes. When the communication range increases to a certain degree, the localization success rate will reach the highest, 100%. In addition, when node density increases, localization success ratio is also increased correspondingly. Figures 5 and 7 indicate that in practical scenario the high localization success ratio and small average localization error can be achieved by using the proposed CS algorithm. The choice of communication range depends on the localization requirements, such as localization accuracy, localization success ratio and the energy constraint of sensor nodes.

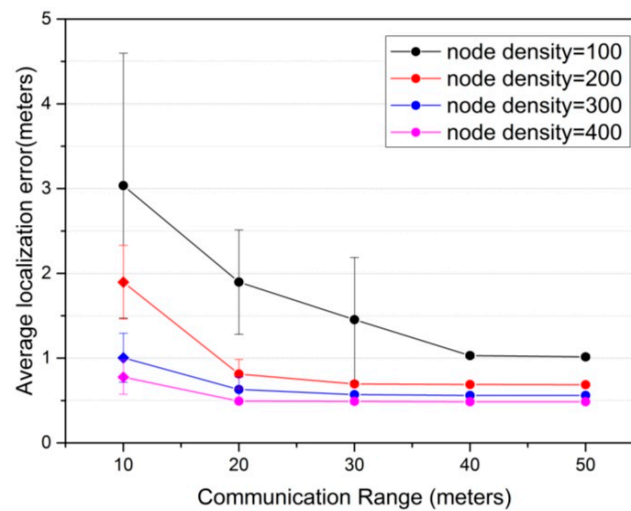


Figure 5. The effect of communication range on average localization error. In addition, the error bar denotes 95% confidence interval of average localization error.

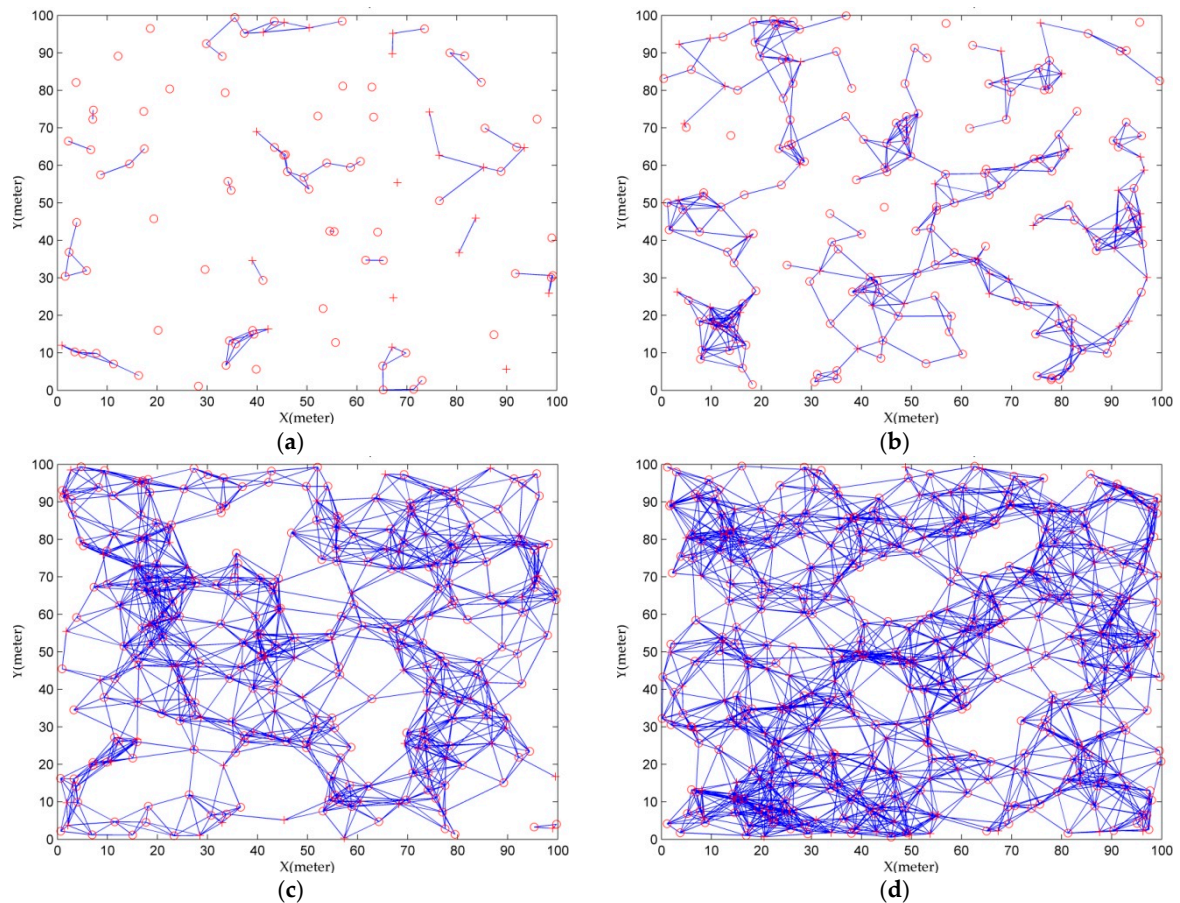


Figure 6. Example of network topology under different node density, (a) 100; (b) 200; (c) 300; (d) 400 when communication range is 10 m and the number of anchor nodes (+) is 20% of node density. In addition, the (o) represents unknown nodes, the (–) represents the communication connectivity between sensor nodes.

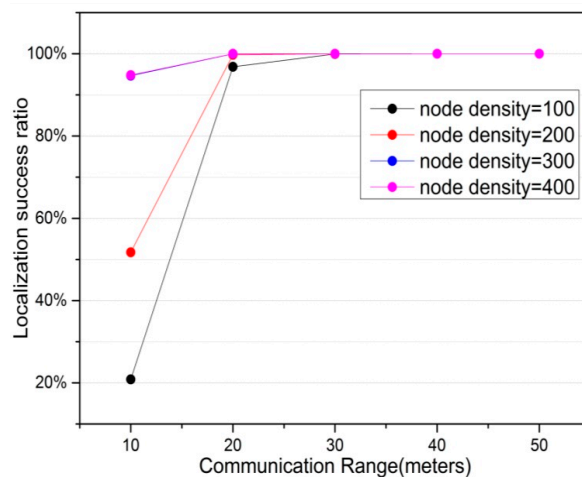


Figure 7. The effect of communication range on localization success ratio.

5.2.3. Comparison with Standard CS and PSO Algorithm

To evaluate the effectiveness of modified CS algorithm, a comparative study was carried out with standard CS algorithm in this subsection. The parameters for standard CS algorithm are set (according to the literature [38]) as follows, the number of nests is 25, the mutation probability Pa is 0.25, the step size α is 0.01. Besides, to prevent the insignificant search over the sensor field, X_{\min} and X_{\max} for the modified CS algorithm are set to 0 and 15 respectively. The network deployment is shown in Figure 8. 40 unknown nodes and 4 anchors are deployed in a $15 \times 15 \text{ m}^2$ sensor field. Anchors are specially deployed in the four corner of sensor area to avoid the collinear problem like the literature [8]. Their coordinates are (0, 0), (0, 15), (15, 0) and (15, 15). In addition, it is assumed that each sensor node can communicate with every other sensor node in the sensor field, as shown the network connectivity in Figure 9.

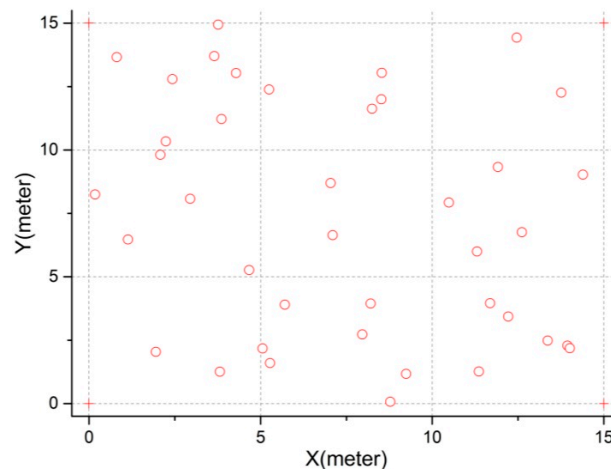


Figure 8. Deployment diagram of 44 sensor nodes with 4 anchor nodes (+) and 40 unknown nodes (o) in $15 \times 15 \text{ m}^2$ sensor field.

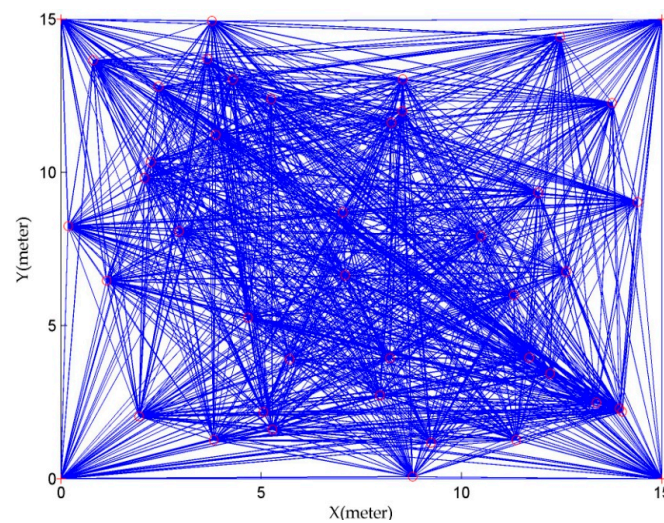


Figure 9. Network topology of sensor nodes, the (o) represents unknown nodes, the (+) represents anchor nodes and the (–) represents the communication connectivity between sensor nodes.

It has been reported in previous work [30,41,42] that the PSO algorithm provides much better performance than Simulated Annealing (SA), Genetic Algorithm (GA), BBO variants and traditional mathematical optimization methods in terms of computational complexity and location accuracy. Thus, localization performance of the modified CS algorithm is also compared with PSO algorithm. The parameters of PSO algorithm are set (according to the literature [29,30]) as follows:

- (1) Population size = 20;
- (2) Acceleration constants $c1 = c2 = 2$;
- (3) Inertia weight $w = 0.7$;
- (4) Limits on particle velocity: $V_{\max} = 15 \text{ m/s}$, $V_{\min} = -15 \text{ m/s}$.

To assess the convergence speed to a minimum average localization error, each algorithm is conducted for thirty iterations. The average localization error with respect to iterations is shown in Figure 10, in which each data point is the average of the results of thirty independent experiments. This is due to the fact that standard CS, PSO and modified CS algorithm are stochastic algorithms; the same result cannot be obtained in all the runs even with the same network deployment. Taking into account the characteristics of stochastic algorithm, it is assumed that convergence is reached if the fluctuation of convergence curve with respect to average localization error is less than 0.01 m.

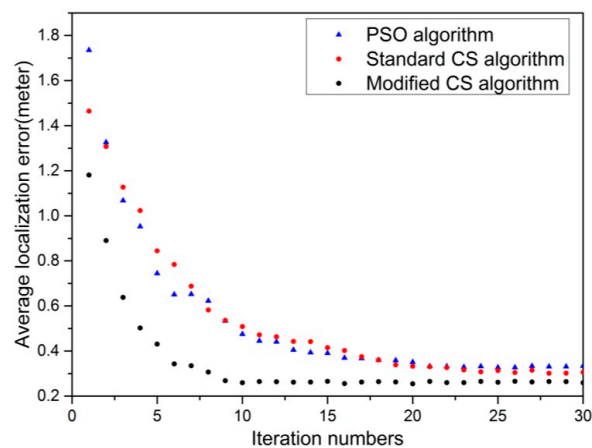


Figure 10. Comparison with standard CS and PSO algorithm.

Figure 10 reflects that as iterations proceed, all algorithms gradually converge to a minimum average localization error at different speeds. The convergence speed reflects the efficiency of the algorithm to find optimal locations. In addition, the algorithm with less iteration requires less resource, which makes it more suitable for WSN. The average localization error of the modified CS algorithm decreases more rapidly than that of PSO algorithm and standard CS algorithm during the first six iterations and the modified CS algorithm converges approximately in the tenth iteration with the average localization error of 0.259 m while the standard CS algorithm takes about twenty-four iterations to reach the convergence with average localization error of 0.307 m. In addition, the PSO algorithm seems to converge at approximately twenty-second iteration with the average localization error of 0.332 m. From the results, it can be seen that the modified CS algorithm can reach the same localization accuracy as standard CS algorithm and PSO algorithm with much fewer iterations. It proves that the modifications of the proposed CS algorithm in increasing convergence speed and location accuracy are observable and effective. This means that it is possible to adopt the proposed CS algorithm in optimizing WSN node localization instead of standard CS algorithm and PSO algorithm.

6. Conclusions

In this paper, we propose a modified CS algorithm for optimizing node localization in WSN. The algorithm adopts the modified step size to enable the population to approach global optimal solution rapidly, and the fitness of each solution is employed to build mutation probability to avoid local convergence. In addition, to prevent the energy consumption caused by insignificant search, the approach restricts the population in the certain range. Extensive experiments have been performed to study the impacts of several factors like anchor density, node density and communication range on the proposed algorithm with respect to average localization error and localization success ratio. This provides the basis for optimizing node localization using the proposed algorithm in practical WSN applications. Additionally, a comparative study has been conducted and experimental results prove that when compared with standard CS and PSO algorithm, the modified CS algorithm performs better in terms of reducing average localization error and increasing the convergence rate, which is favorable to reduce computational consumption and thus prolong the lifetime of sensor nodes. The effectiveness of the modified CS algorithm was verified through simulation results. In the future, we will focus on designing an experiment system to test the modified CS algorithm in practical applications.

Acknowledgments: This work is supported by the fund from Science and Technology Planning Projects of Guangdong Province (No. 2015B010104003).

Author Contributions: Jing Cheng and Linyuan Xia conceived and designed the experiments; Jing Cheng performed the experiments and wrote the paper.

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

References

1. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. Wireless sensor networks: A survey. *Comput. Netw.* **2002**, *38*, 393–422. [[CrossRef](#)]
2. Kim, S.; Pakzad, S.; Culler, D.; Demmel, J.; Fennes, G.; Glaser, S.; Turon, M. Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks. In Proceedings of the 6th International Symposium on Information Processing in Sensor Networks, Cambridge, MA, USA, 25–27 April 2007; pp. 254–263.
3. Da Silva, V.B.C.; Sciamarella, T.; Campista, M.E.M.; Costa, L.H.M.K. A Public Transportation Monitoring System Using IEEE 802.11 Networks. In Proceedings of the IEEE Computer Networks and Distributed Systems, Florianopolis, Brazil, 5–9 May 2014; pp. 451–459.
4. Suryadevara, N.K.; Mukhopadhyay, S.C.; Kelly, S.D.T.; Gill, S.P.S. WSN-Based Smart Sensors and Actuator for Power Management in Intelligent Buildings. *IEEE/ASME Trans. Mechatron.* **2015**, *20*, 564–571. [[CrossRef](#)]
5. Garcia-Sanchez, A.J.; Garcia-Sanchez, F.; Losilla, F.; Kulakowski, P.; Garcia-Haro, J.; Rodríguez, A.; López-Bao, J.; Palomares, F. Wireless Sensor Network Deployment for Monitoring Wildlife Passages. *Sensors* **2010**, *10*, 7236–7262. [[CrossRef](#)] [[PubMed](#)]

6. Aslan, Y.E.; Korpeoglu, I.; Ulusoy, Ö. A framework for use of wireless sensor networks in forest fire detection and monitoring. *Comput. Environ. Urban Syst.* **2012**, *36*, 614–625. [[CrossRef](#)]
7. Huang, X.; Yi, J.; Chen, S.; Zhu, X. A Wireless Sensor Network-Based Approach with Decision Support for Monitoring Lake Water Quality. *Sensors* **2015**, *15*, 29273–29296. [[CrossRef](#)] [[PubMed](#)]
8. Patwari, N.; Ash, J.N.; Kyperountas, S.; Hero, A.O.; Moses, R.L.; Correal, N.S. Locating the nodes: Cooperative localization in wireless sensor networks. *IEEE Signal Process. Mag.* **2005**, *22*, 54–69. [[CrossRef](#)]
9. Bulusu, N.; Heidemann, J.; Estrin, D. GPS-less low-cost outdoor localization for very small devices. *IEEE Pers. Commun.* **2000**, *7*, 28–34. [[CrossRef](#)]
10. Vecchio, M.; López-Valcarce, R.; Marcelloni, F. A two-objective evolutionary approach based on topological constraints for node localization in wireless sensor networks. *Appl. Soft Comput.* **2012**, *12*, 1891–1901. [[CrossRef](#)]
11. Boukerche, A.; Oliveira, H.A.; Nakamura, E.F.; Loureiro, A.A.F. Localization systems for wireless sensor networks. *IEEE Wirel. Commun.* **2007**, *14*, 6–12. [[CrossRef](#)]
12. Mao, G.; Fidan, B.; Anderson, B.D. Wireless sensor network localization techniques. *Comput. Netw.* **2007**, *51*, 2529–2553. [[CrossRef](#)]
13. Pal, A. Localization algorithms in wireless sensor networks: Current approaches and future challenges. *Netw. Protoc. Algorithms* **2010**, *2*, 45–73. [[CrossRef](#)]
14. Niculescu, D.; Nath, B. Ad hoc positioning system (APS). In Proceedings of the IEEE Global Telecommunications Conference, San Antonio, TX, USA, 25–29 November 2001; pp. 2926–2931.
15. Rabaey, C.S.J.; Langendoen, K. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In Proceedings of the USENIX technical annual conference, Monterey, CA, USA, 10–15 June 2002; pp. 317–327.
16. Doherty, L.; Pister, K.S.J.; El Ghaoui, L. Convex position estimation in wireless sensor networks. In Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, Anchorage, AK, USA, 22–26 April 2001; pp. 1655–1663.
17. Biswas, P.; Lian, T.C.; Wang, T.C.; Ye, Y. Semidefinite programming based algorithms for sensor network localization. *ACM Trans. Sensors Netw.* **2006**, *2*, 188–220. [[CrossRef](#)]
18. Simonetto, A.; Leus, G. Distributed maximum likelihood sensor network localization. *IEEE Trans. Signal Process.* **2014**, *62*, 1424–1437. [[CrossRef](#)]
19. Shang, Y.; Ruml, W. Improved MDS-based localization. In Proceedings of the Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, Hong Kong, China, 7–11 March 2004; pp. 2640–2651.
20. Simon, D. Biogeography-based optimization. *IEEE Trans. Evolut. Comput.* **2008**, *12*, 702–713. [[CrossRef](#)]
21. Blum, C.; Roli, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv.* **2003**, *35*, 268–308. [[CrossRef](#)]
22. 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, San Antonio, TX, USA, 11–14 October 2009; pp. 205–210.
23. Kumar, A.; Khosla, A.; Saini, J.S.; Singh, S. Meta-heuristic range based node localization algorithm for Wireless Sensor Networks. In Proceedings of the IEEE International Conference on Localization and GNSS, Starnberg, Munich, Germany, 25–27 June 2012; pp. 1–7.
24. Yun, S.; Lee, J.; Chung, W.; Kim, E.; Kim, S. A soft computing approach to localization in wireless sensor networks. *Expert Syst. Appl.* **2009**, *36*, 7552–7561. [[CrossRef](#)]
25. Kannan, A.A.; Mao, G.; Vucetic, B. Simulated annealing based wireless sensor network localization with flip ambiguity mitigation. In Proceedings of the 63rd IEEE Vehicular Technology Conference, Melbourne, Australia, 7–10 May 2006; pp. 1022–1026.
26. James, K.; Russell, E. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948.
27. Chih, M. Self-adaptive check and repair operator-based particle swarm optimization for the multidimensional knapsack problem. *Appl. Soft Comput.* **2015**, *26*, 378–389. [[CrossRef](#)]
28. Chih, M.; Lin, C.J.; Chern, M.S.; Ou, T.Y. Particle swarm optimization with time-varying acceleration coefficients for the multidimensional knapsack problem. *Appl. Math. Model.* **2014**, *38*, 1338–1350.

29. Gopakumar, A.; Jacob, L. Localization in wireless sensor networks using particle swarm optimization. In Proceedings of the IET International Conference on Wireless, Mobile and Multimedia Networks, Mumbai, India, 11–12 January 2008; pp. 227–230.
30. Singh, S.; Mittal, E. Range based wireless sensor node localization using PSO and BBO and its variants. In Proceedings of the IEEE International Conference on Communication Systems and Network Technologies (CSNT), Gwalior, India, 6–8 April 2013; pp. 309–315.
31. Kim, D.H.; Abraham, A.; Cho, J.H. A hybrid genetic algorithm and bacterial foraging approach for global optimization. *Inf. Sci.* **2007**, *177*, 3918–3937. [[CrossRef](#)]
32. Abd-Elazim, S.M.; Ali, E.S. A hybrid particle swarm optimization and bacterial foraging for optimal power system stabilizers design. *Int. J. Electr. Power Energy Syst.* **2013**, *46*, 334–341. [[CrossRef](#)]
33. Das, S.; Suganthan, P.N. Differential evolution: A survey of the state-of-the-art. *IEEE Trans. Evolut. Comput.* **2011**, *15*, 4–31. [[CrossRef](#)]
34. Chagas, S.H.; Martins, J.B.; de Oliveira, L.L. Genetic algorithms and simulated annealing optimization methods in wireless sensor networks localization using artificial neural networks. In Proceedings of the 55th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), Boise, ID, USA, 5–8 August 2012; pp. 928–931.
35. Li, S.P.; Wang, X.H. The research on Wireless Sensor Network node positioning based on DV-hop algorithm and cuckoo searching algorithm. In Proceedings of the IEEE International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC), Shenyang, China, 20–22 December 2013; pp. 620–623.
36. Goyal, S.; Patterh, M.S. Wireless sensor network localization based on cuckoo search algorithm. *Wirel. Pers. Commun.* **2014**, *79*, 223–234. [[CrossRef](#)]
37. Yang, X.S.; Deb, S. Cuckoo search via Lévy flights. In Proceedings of the IEEE World Congress on Nature & Biologically Inspired Computing, Coimbatore, India, 9–11 December 2009; pp. 210–214.
38. Yang, X.S. *Nature-Inspired Metaheuristic Algorithms*, 2nd ed.; Luniver Press: Somerset, UK, 2010.
39. Walton, S.; Hassan, O.; Morgan, K.; Brown, M.R. Modified cuckoo search: a new gradient free optimisation algorithm. *Chaos Solitons Fractals* **2011**, *44*, 710–718. [[CrossRef](#)]
40. Valian, E.; Tavakoli, S.; Mohanna, S.; Haghi, A. Improved cuckoo search for reliability optimization problems. *Comput. Ind. Eng.* **2013**, *64*, 459–468. [[CrossRef](#)]
41. Namin, P.H.; Tinati, M.A. Node localization using particle swarm optimization. In Proceedings of the Seventh IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Melbourne, Australia, 6–9 December 2011; pp. 288–293.
42. Li, Z.; Lei, L. Sensor node deployment in wireless sensor networks based on improved particle swarm optimization. In Proceedings of the IEEE International Conference on Applied Superconductivity and Electromagnetic Devices, Chengdu, China, 25–27 September 2009; pp. 215–217.



© 2016 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/>).