

Research Article

A Faster Convergence Artificial Bee Colony Algorithm in Sensor Deployment for Wireless Sensor Networks

Xiangyu Yu,¹ Jiaxin Zhang,¹ Jiaru Fan,¹ and Tao Zhang²

¹ School of Electronic and Information Engineering, South China University of Technology, Guangzhou, Guangdong 510640, China

² School of Electronic and Communication Engineering, Guiyang University, Guiyang, Guizhou 550005, China

Correspondence should be addressed to Xiangyu Yu; yuxy@scut.edu.cn

Received 27 June 2013; Revised 26 August 2013; Accepted 28 August 2013

Academic Editor: Chang Wu Yu

Copyright © 2013 Xiangyu Yu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In wireless sensor networks (WSN), sensor deployment is one of the main topics for enhancing the sensor's coverage rate. In this paper, by modifying updating equation of onlooker bee and scout bee of original artificial bee colony (ABC) algorithm, a sensor deployment algorithm based on the modified ABC algorithm is proposed. Some new parameters such as forgetting and neighbor factor for accelerating the convergence speed and probability of mutant for maximizing the coverage rate are introduced. Simulation results showed that comparing with the deployment method based on original ABC and particle swarm optimization (PSO) algorithm, the proposed approach can achieve a better performance in coverage rate and convergence speed while needing a less total moving distance of sensors.

1. Introduction

Integrating with sensor technology, distributed information process, embedded technology, wireless communication and microelectronic technique, and so forth, wireless sensor networks (WSN) have become active fields of research. Because of their advantages in low energy consumption, distributed self-organization ability, WSN are extensively used in fields such as target tracking, environment monitoring and national defense, and underwater detecting [1, 2].

Due to its relation with energy saving, connectivity, and network reconfiguration, coverage is an important issue in WSN. It mainly addresses how to deploy the sensors to achieve sufficient coverage of the service area so that each position in the service area is monitored at least by one sensor. A good coverage is indispensable for the effectiveness of WSN. An efficient deployment of sensors will reduce the configuration and communication consumption of the network and improve the resource management; thus sensor deployment becomes a substantial work [3, 4].

Sensor deployment algorithms can be divided into static or dynamic ones. In some cases such as underwater, battlefield, and abominable environment where it is hard to arrange

sensors manually, only dynamic deployment approaches can be applied. In dynamic deployment, sensors are initially located with random coordinates and can change their positions by using their knowledge of other's positions and other information collected. In order to increase the coverage rate of the whole network, many approaches have been proposed for dynamic node deployment, such as potential [5], virtual force [6–8], Voronoi diagram [9], and bionic intelligence algorithm [10–16].

In fact, the object of sensor deployment is to seek an optimal sensor placement, which has a resemblance on the bionic intelligence algorithm. Bionic intelligence algorithms are inspired by the social behavior of different animals or species and have become a research of interest in different domains for solving complex optimization problems. In order to ensure the effectiveness and the quality of the WSN, bionic intelligence and swarm intelligence algorithms are introduced in sensor deployment, for instance, the genetic algorithm (GA) [10], the particle swarm optimization (PSO) [11–13], the ant colony optimization (ACO), the glowworm (GSO) [14], and so forth. Among which, ABC-based deployment algorithm has been proved to be competitive with some conventional optimization algorithms [15, 16].

Based on the foraging behavior of honey bee swarms, ABC algorithm was proposed to optimize multivariable and multimodal continuous functions. It is found several applications in electrical engineering, mechanical and civil engineering areas, electronics, software and control engineering areas, image processing area, and so forth [17]. The original ABC algorithm, however, has low convergence speed and is easily trapped into the local optimum when solving complex multimodal problems. Therefore, a few modified versions were proposed by many researchers to improve it [18–23].

In this paper, a sensor deployment algorithm based on modified ABC algorithm called FNF, (forgetting and neighbor factor)-BL (backward learning) ABC algorithm is proposed. In order to have a better coverage and a faster convergence speed, the onlooker bee phase and the scout bee phase of the ABC algorithm have been modified. In onlooker bee phase, a modification called FNF, which introduced forgetting factor and neighbor factor into the original equation, is adopted. The scout bee phase is all new designed by changing the original method into the BL method.

The rest of this paper is organized as follows: original ABC algorithm and its application in sensor deployment algorithm are described in Section 2; our modifications of ABC algorithm are presented in Section 3; experiment results and analysis are shown in Section 4; some conclusions are drawn and our future works are discussed in Section 5.

2. ABC Algorithm and Its Application in Deployment

2.1. ABC Algorithm. In ABC algorithm, the colony of artificial bees consists of three groups of bees: employed bees, onlooker bees, and scout bees. The food searching of bees is collectively performed by these three kinds of bees. The employed bees search the food around the food sources in their memory and deliver the food information, such as the fitness value, to the onlooker bees. The onlooker bees select good food sources from this information and search the food around the selections for a better one. The scout bees are translated from a few employed bees, which abandon their food sources and search new ones. Each cycle of the searching consists of three phases, as describe below.

At initiation, food sources are randomly generated by:

$$x_{il} = x_l^{\min} + (x_l^{\max} - x_l^{\min}) \times \text{rand}(0, 1), \quad (1)$$

where x_l^{\max} and x_l^{\min} are the lower and upper bounds of the position and $\text{rand}(0, 1)$ is the random number in the interval $(0, 1)$.

Then at the employed bee phase (first phase), a neighbor of the current food source k is randomly selected for the employed bee and then a new solution is produced, and its fitness value is calculated. For each employed bee, a candidate food source solution from the previous one is generated by:

$$V_{il} = x_{il} + \text{rand}(-1, 1) \times (x_{il} - x_{kl}), \quad (2)$$

where $l \in \{1, 2, \dots, M\}$, M is the number of variables (problem dimension) to be optimized, $i, k \in \{1, 2, \dots, J\}$, and $i \neq k$, J is

the number of employed bees (the number of food sources), and $\text{rand}(-1, 1)$ is a random number between $(-1, 1)$.

After comparing the fitness value of the new food source (solution) with the previous one, the employed bee memorizes the one which has higher fitness value. After all employed bees complete the searching procedure, they translate the fitness values of the food sources (solutions) and their positions information to the onlooker bees in the hive.

At the onlooker bee phase (second phase), for each onlooker bee, a food source is inherited depending on the probability value associated with that food source, p_i , calculated by:

$$p_i = \frac{\text{fitness}(\mathbf{x}_i)}{\sum_{i=1}^S \text{fitness}(\mathbf{x}_i)}. \quad (3)$$

After an onlooker bee reaches a food source, it looks for a new source within the neighborhood of the previous one and memorizes the food sources according to their fitness value. The update process used in the onlooker bee phase is the same as that in the employed bee phase, referring to (2). The main distinction between the employed bee phase and the onlooker phase is that every solution in the employed bee phase involves the update process, but only the selected solutions depending on the probability value have the opportunity to update in the onlooker bee phase.

The selection of the scout bee is controlled by a control parameter called “limit,” which is a predetermined number of trials. If a solution representing a food source cannot be improved by the number of “limit,” the source is considered to be exhausted and then the employed bee of this food source becomes a scout. The position of the abandoned food source is replaced with a random search by (1). So “limit” controls the selection of the scout bee and the qualities of solutions. This is the final phase of the ABC algorithm.

These three phases are repeated until the value of cycle is up to maximum cycle number (MaxCycle).

2.2. ABC Algorithm in Deployment Problem. The object of sensor deployment algorithm is to determine an optimum sensor distribution in an area of interest. So applying ABC algorithm into sensor deployment problem is possible. For each sensor in WSN there is a sensing range, which indicates the sensing ability of the sensor, and a communication range, which refers to the longest distance through which two sensors can exchange information with each other. These two ranges restraint the performance of a deployment algorithm. In this paper, there are three assumptions: first, sensing radius of the sensors indicating the sensing ability is same; second, all of the sensors are interconnected; and finally, all of the sensors can move. Table 1 shows the relationship between the parameters in ABC algorithm and the problem of sensor deployment. Notice that in this paper the coverage rate is set as the reciprocal of the fitness of food source fitness. So the optimum sensor distribution corresponds to the lowest fitness. Simulation results in [16] have verified that the ABC algorithm is more successful than the PSO algorithm for the dynamic deployment problem of WSN using a probabilistic detection model. Deployments found by ABC are better than

TABLE 1: Corresponding parameters in ABC algorithm and sensor deployment.

ABC algorithm	Sensor deployment problem
Solution of a food source	Sensor distribution
M dimensions in each solution	M sensors' coordinate
Fitness of the solution	Coverage rate of interest area
Lowest fitness	Optimum sensor distribution

those by PSO for all of the 30 independent runs which are started with the same initial deployment in that literature.

3. Modified ABC Algorithm and Its Application in Deployment

In sensor deployment, coverage and convergence are very important issues. In order to achieve the higher coverage rate and faster convergence speed, two modifications are introduced in our FNF-BL-ABC-based approach.

3.1. Modification in Onlooker Bee Phase. The original food source updating mechanism is calculated by (2). Because of the neighbor food source is randomly chosen, the fitness of the neighbor food source may be higher, which means the sensors distribution is worse than the current one. In addition, the updating equation remains unchanged as the iteration processes. Thus, this original updating mechanism cannot help bee to find the optimum food sources effectively, which results in low speed of achieving the optimum solution. In order to accelerate the whole procedure, two parameters named forgetting factor and neighbor factor [22] are introduced into (2):

$$v_{il} = \eta \times x_{il} + \tau \times \text{rand}(-1, 1) \times (x_{il} - x_{kl}), \quad (4)$$

where τ is the forgetting factor and η is the neighbor factor.

Among these two parameters, the value of η is related to the fitness of neighbor food source. The variation tendency of the value of neighbor factor is increased to prevent the global searching ability from being lost. The neighbor factor η can strengthen or weaken the relation between present and neighbor food source. The forgetting factor τ stands for the intension of present food source memory when searching for the next food source. In order to fully use the neighbor information and find the global best, the forgetting factor is dynamically decreased as

$$\tau = \lambda \times w_\tau. \quad (5)$$

The value of neighbor factor η is related to the fitness of neighbor food source as in

$$\eta = \lambda \times w_\eta. \quad (6)$$

In (5) and (6), when the fitness value of neighbor food source is lower than the present one, $\lambda > 1$; when the fitness

value of neighbor food source is bigger than the present one, $\lambda < 1$. Consider

$$w_\eta = w_4 - \left(\frac{\text{maxcycle} - \text{iter}}{\text{maxcycle}} \right)^\alpha * (w_4 - w_3), \quad (7)$$

$$w_\tau = w_1 + \left(\frac{\text{maxcycle} - \text{iter}}{\text{maxcycle}} \right)^\beta * (w_2 - w_1). \quad (8)$$

In (7) and (8), w_1 , w_2 , w_3 , and w_4 are constants. The interval of α and β is $[0.8, 1]$ and $[1, 1.2]$. The parameter w_η becomes larger from w_3 to w_4 , and w_τ is reduced from w_2 as w_1 with the iteration goes.

3.2. Modification in Scout Bee Phase. In original ABC algorithm, the work of scout bee is to abandon the food source remaining unchanged by the “limit” time and randomly generating a food source for avoiding trapping into local optimum. The scout bee transforms to employed bee. The performance of deployment algorithm is affected by the parameter “limit.” In order to reach a high coverage rate, the value of “limit” is important. However, it needs numerous experiments to find an appropriate value. Thus, a modification called backward learning [20] is introduced into the ABC algorithm. This approach can be described as shown in Pseudocode 1.

The following equation can calculate the value of probability of mutant (PM):

$$PM = 0.01 + 0.1 \times (2 - e^{\text{iter} * \ln 2 / \text{maxcycle}}). \quad (9)$$

The backward learning strategy can be expressed as

$$v_{il} = x_{wl} + x_{bl} + \text{rand}(0, 1) \times x_{il}, \quad (10)$$

where x_w , and x_b are the worst and best quality food source. Record v_i if it is better than x_i .

4. Simulation Results

PSO and ABC are two most popular swarm intelligence algorithms; they both simulate the collective behavior of decentralized, self-organized systems and have many ideas in common, so in many research work the simulation results of ABC algorithm are compared with those of the PSO algorithm, which is also taken in this paper.

In this section, PSO algorithm and its parameters are first presented in detail, followed by explanations how the parameters in our proposed algorithm are chosen. Two sets of simulations are made to verify the effectiveness of the proposed approach. The first is an ideal case in a square room; the second is in a general room with obstacles. Three kinds of approaches, which are original ABC-based, FNF-BL-ABC-based, and PSO-based deployment, are compared. To guarantee the comparability between them, the experimental circumstances are identical consisting of the initial deployment and iteration time.

```

For i = 1:n (n stands for the number of food source)
    If rand < probability of mutant (PM)
        backward learning.
    end
end

```

PSEUDOCODE 1: Pseudocode of backward learning method.

4.1. Parameters Selection

4.1.1. Parameters Selection for PSO Algorithm. A standard version of PSO is applied in the sensor deployment in [24]. In PSO algorithm, several individuals are “evolved” by learning the experience from their own and companions through generation. Each individual is called “particle.” Each particle also has a l -dimension space, which is similar to the food source in the ABC algorithm. Each particle is also randomly generated with (1).

The updating processing of the particle in PSO algorithm is divided into two phases.

In the first phase, the velocity is calculated as

$$v_{il} = v_{il} + c_1 * \text{rand1}() * (p_{il} - x_{il}) + c_2 * \text{rand2}() * (p_{gl} - x_{il}). \quad (11)$$

In the second phase, the position of the particle is updated as (12)

$$x_{il} = x_{il} + v_{il}. \quad (12)$$

There are some parameters that should be explained in these equations: $\text{rand}()$, $\text{rand1}()$, and $\text{rand2}()$ are three numbers independently generated by the function “ $\text{rand}()$ ” in MATLAB. p_{il} is the best position of the i th particle, and p_{gl} is the best position of the particles in the neighbor of i th particle. x_{il} is the i th particle’s position and v_{il} is its velocity. Based on [13, 24, 25], c_1 and c_2 are two positive numbers which can influence the performance of the algorithm and are calculated by

$$c_1 = c_{1_max} - (c_{1_max} - c_{1_min}) * \frac{t}{gen}, \quad (13)$$

$$c_2 = c_{2_max} - (c_{2_max} - c_{2_min}) * \frac{t}{gen},$$

where c_{1_max} , c_{1_min} , c_{2_max} , c_{2_min} are parameters set to change the performance of the algorithm. t is the iteration times, and gen is the maximum iteration times.

In this paper, 11 combinations of c_{1_max} , c_{1_min} , c_{2_max} , and c_{2_min} have been tried to improve the performance in the optimizer processing. In this set of experiments, both c_{1_min} and c_{2_min} are invariable, equal to 1. The threshold of velocity (max_velocity) equals 6. The experiment results are shown in Table 2. The eighth experiment, in which c_{1_max} and c_{2_max} equal 3.4, has the best performance among the 11 combinations. Thus, in the comparison with FNF-BL-ABC

TABLE 2: Experiment results with different values of parameters in PSO algorithm.

Combination		Coverage rate
$c_{1_max} = 2$	$c_{2_max} = 2$	0.9035
$c_{1_max} = 2.2$	$c_{2_max} = 2.2$	0.8972
$c_{1_max} = 2.4$	$c_{2_max} = 2.4$	0.9059
$c_{1_max} = 2.6$	$c_{2_max} = 2.6$	0.9136
$c_{1_max} = 2.8$	$c_{2_max} = 2.8$	0.9136
$c_{1_max} = 3$	$c_{2_max} = 3$	0.9253
$c_{1_max} = 3.2$	$c_{2_max} = 3.2$	0.9369
$c_{1_max} = 3.4$	$c_{2_max} = 3.4$	0.9466
$c_{1_max} = 3.6$	$c_{2_max} = 3.6$	0.9460
$c_{1_max} = 3.8$	$c_{2_max} = 3.8$	0.9253
$c_{1_max} = 4$	$c_{2_max} = 4$	0.8910

and original ABC algorithm, the parameters in PSO are chosen as $c_{1_max} = 3.4$, $c_{1_min} = 1$, $c_{2_max} = 3.4$, $c_{2_min} = 1$, and the threshold of velocity (max_velocity) = 6.

4.1.2. Parameter Selection for the Proposed Algorithm. As for the proposed algorithm, there are three different sets of parameters:

- (i) w_i ($i = 1, \dots, 4$),
- (ii) λ ,
- (iii) α , β .

In this paper, we set $\alpha = 0.8$ and $\beta = 1.2$ for the reason that has been described in [21]. So a large number of tests have been made for the first two parameter sets. Among all the tests for w_i , the best 5 combinations are (1) $w_1 = w_3 = 0.2$ and $w_2 = w_4 = 1.2$; (2) $w_1 = w_3 = 0.2$ & $w_2 = w_4 = 1.6$; (3) $w_1 = w_3 = 0.2$ & $w_2 = w_4 = 2.0$; (4) $w_1 = w_3 = 0.2$ & $w_2 = w_4 = 2.4$; (5) $w_1 = w_3 = 0.2$ & $w_2 = w_4 = 2.8$. Among which, the simulation results of combinations 2 and 3 are better in convergence speed than the one $w_1 = w_3 = 0.2$ & $w_2 = w_4 = 1.2$ given in [21], while combination 4 and 5 have almost the same performance as that in [21]. So in our simulations, we set $w_1 = w_3 = 0.2$ & $w_2 = w_4 = 1.6$, and this set of parameters is used for choosing λ . As for parameter λ , after numerous experiments, the best six sets are, from the best to the worst, (1) 1.4 or 0.6; (2) 1.3 or 0.7; (3) 1.2 or 0.8 (which is used in [21]); (4) 1.5 or 0.5; (5) 1.6 or 0.4; (6) 1.7 or 0.3 in (6) and (7), and we choose (1) in our simulation for a better performance in convergence speed.

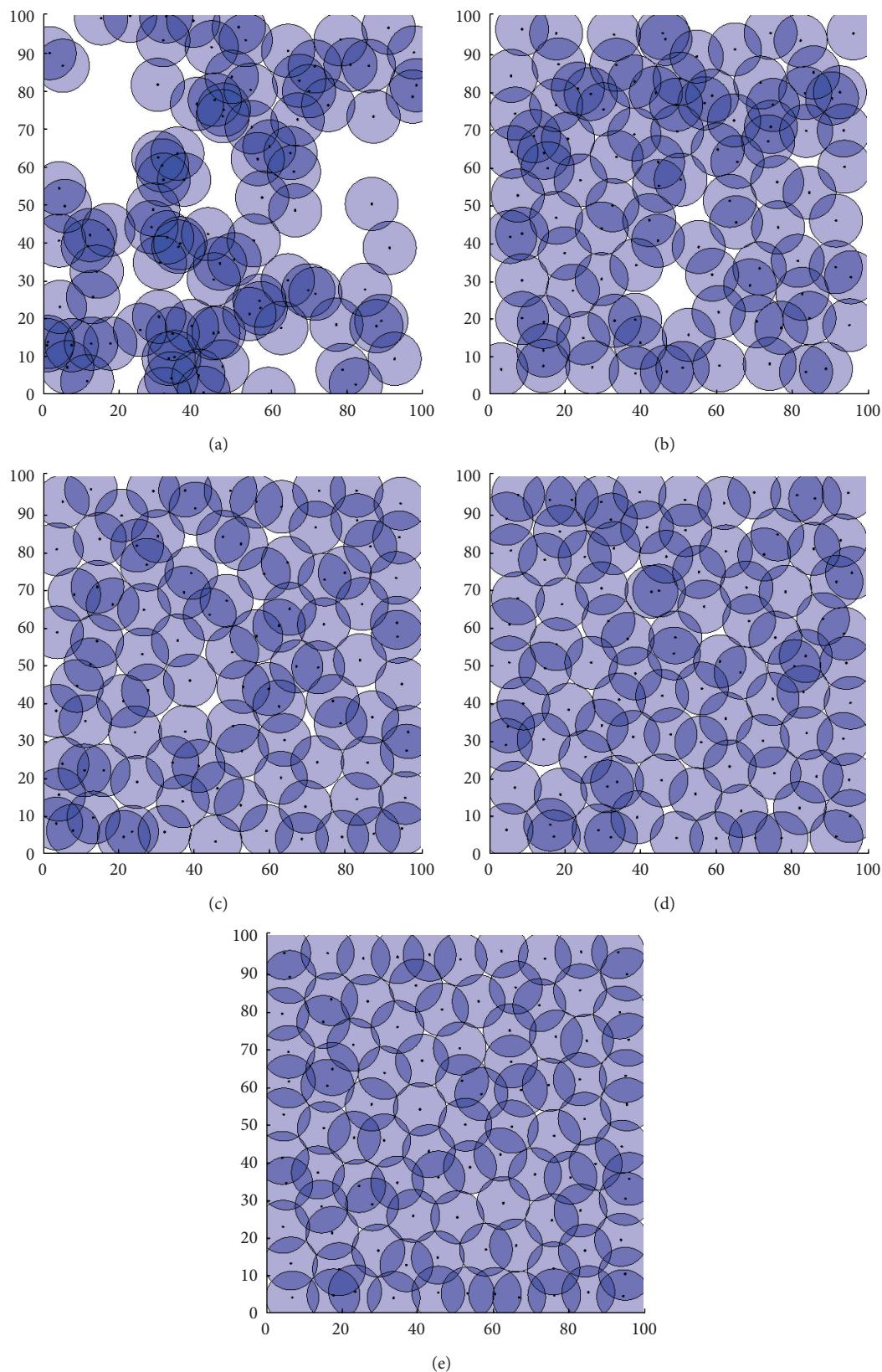


FIGURE 1: Sensor distribution: (a) initial, (b) 100 iterations, (c) 500 iterations, (d) 1000 iterations, and (e) 10000 iterations.

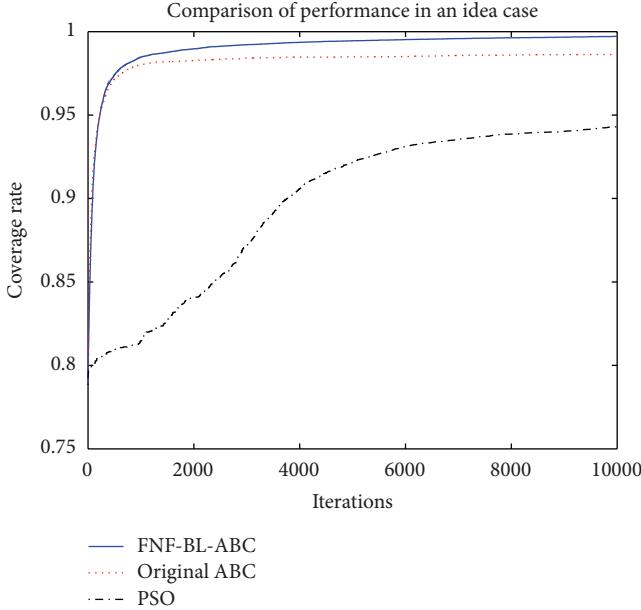


FIGURE 2: Coverage rate comparison between FNF-BL-ABC, original ABC, and PSO for an ideal room.

4.2. Simulations in an Ideal Room. In order to make the simulation more reliable, the parameters such as the number of sensors and their sensing radius are the same as those in [16]. There are 100 mobile sensors in our simulation. The sensing radius of each sensor r is 7. The proportion of target area is 100×100 . The max cycle is 10000. The colony size is 20. The run times are 20, and we calculate the average to make the result more reliable and a higher robustness. In the original ABC algorithm, there is a parameter called “limit” which needs to be set, however, since the scout bee phase is modified, we discard this parameter and the “limit” is dispensable.

Figure 1 shows the distribution of initial, 100 iterations, 500 iterations, 1000 iterations, and 10000 iterations. From these figures we can find that the proposed approach can cover the entire area effectively.

The performance of original ABC-based, FNF-BL-ABC-based and PSO-based deployment algorithm is compared in Figure 2. Each algorithm is based on the same initial deployment as in Figure 1.

As can be seen in Figure 2, at iteration 10000, the coverage rate is 99.71% for the proposed FNF-BL-ABC approach, which is larger than the 98.63% of the original ABC-based one and 94.31% of the PSO-based one, which showed that the proposed approach has better coverage rate than the other two. Meanwhile, we can also find that proposed one has faster convergence speed than the other two in Figure 2 and Table 3 and the effectiveness of the modifications is verified.

In (2), the first part is the information of the current food source. The second part is the different information between the current and neighbor food source, because the neighbor is selected randomly, which means the fitness of them is not considered. Moreover, the iteration number of times is also ignored by the food source updating process. These aspects

TABLE 3: Deployment results comparison between 3 algorithms at some iterations.

Number of iterations	100	500	2000	5000	10000
FNF-BL-ABC	0.9095	0.9740	0.9897	0.9945	0.9971
Original ABC	0.9184	0.9713	0.9826	0.9848	0.9863
PSO	0.7993	0.8097	0.8403	0.9215	0.9431

will reduce the convergence speed. Equation (2) is replaced by (4). In (5) and (6), the values of τ and η are associated with the comparison between the fitness of the neighbor and current food source. They are also affected by the iteration number of times in (8) and (9).

In the later stage of the whole algorithm, each food source has a better quality than before. However, some or all of them may be lost in the local optimum. In (9), the probability of mutant is increased with the process of the algorithm. The original method in (1) is altered by the backward learning method in (10). This backward learning method, which replaces the original solution by its relative solution, can acquire a better evaluation than the random solution. This conclusion has been testified in [26]. For one, the convergence speed can be increased immensely. For other, the local optimum can be somewhat avoided. Thus, the modified version of ABC algorithm can achieve a better performance in convergence.

In the practical use of WSN, power consumption must be considered, which is mainly in two ways: transmission/receive signal and movement. The energy consumption in moving is related to the total moving distance of all nodes. In our implement, we introduce the concept of “virtual move,” which means that sensors do not move at each step, but move to the destination position after the final iteration only once. We record the total moving distance for each sensor from initial to final positions; for the original ABC-based algorithm, the total moving distance is 5216.1 while, for the proposed approach, the value is 3541.6. As for PSO-based algorithm, since it has worst coverage rate, we do not compare its moving distance here. Thus, we can make a conclusion that the energy consumption of the proposed approach is better than that of original ABC-based algorithm.

4.3. Simulations in a Room with Obstacles. In order to verify the effectiveness of the proposed approach in general case, there are two obstacle areas in the 100×100 room, each is 30×20 , and there are 90 sensors to be placed in this room. The parameters for each sensor are the same as those in Section 4.1. Figure 3 shows the sensor distribution of initial and after 10000 iterations for both the proposed approach and the original ABC algorithm.

Because the performance of PSO algorithm is not good in Figure 2, only the comparison between FNF-BL version and original-ABC algorithm is provided here and shown in Figure 4.

From Figure 4 we can find that the proposed approach has higher coverage rate than original ABC. For this case with obstacles, the total moving distance of all sensors is 3817.8 for the proposed approach and 4870.4 for the original

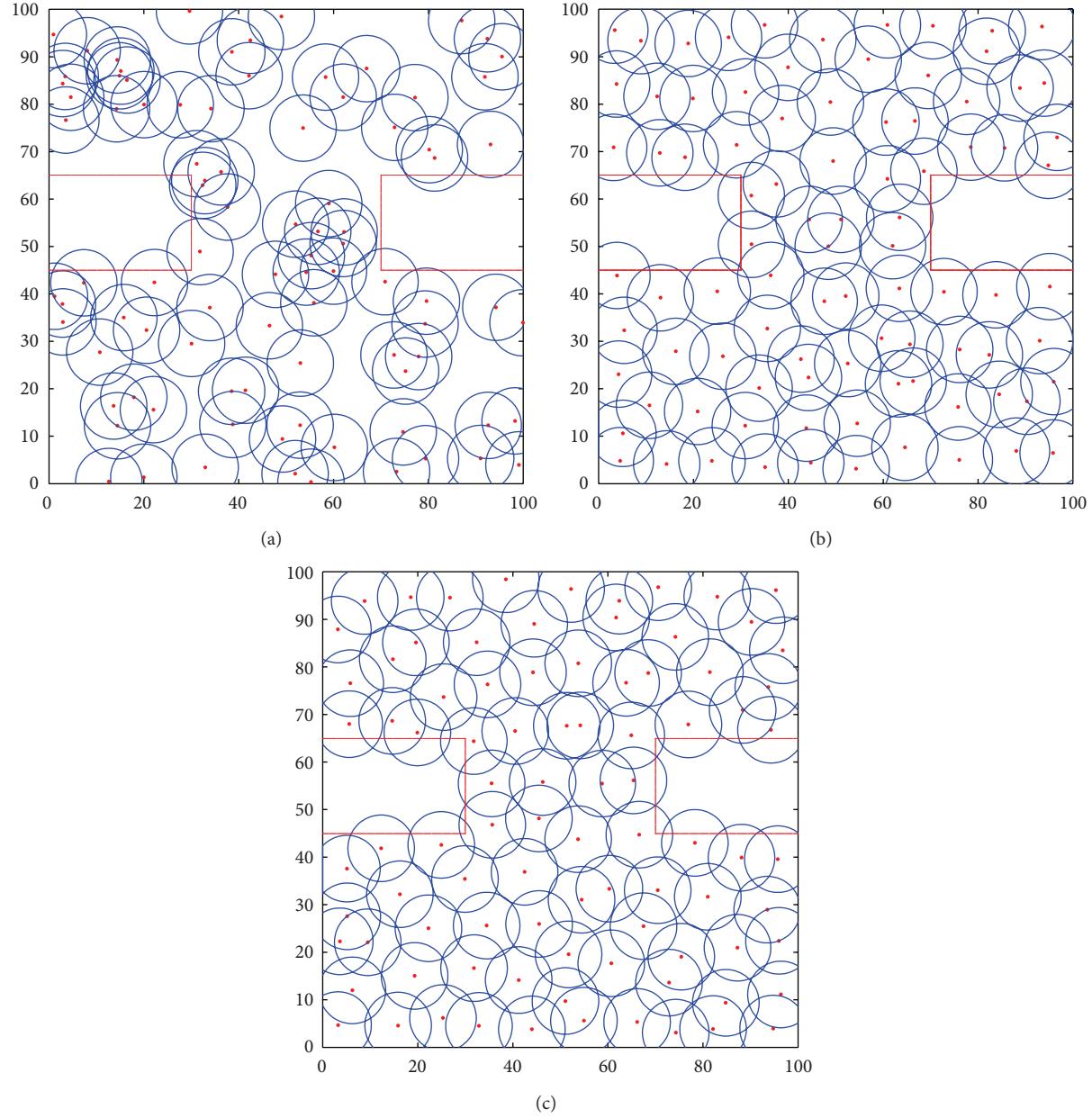


FIGURE 3: Sensor distribution: (a) initial, (b) original ABC algorithm after 10000 iterations, and (c) FNF-BL-ABC algorithm after 10000 iterations.

ABC algorithm. These again showed that, in the case with obstacles, the proposed approach has higher coverage rate and less energy consumption than original one.

5. Conclusion

In this paper, in order to optimize the deployment of wireless sensors, a modified ABC algorithm introducing forgetting and neighbor factor in the onlooker bee phase and backward learning in the scout bee phase is proposed. Simulation results showed that the proposed modified ABC algorithm has higher coverage rate and fast convergence along with less total

moving distance for sensor deployment in an area of interest. It has revealed a better performance in solving this kind of problem. In our future work, virtual force-based deployment algorithm will be introduced before the proposed approach is applied to make the initial distribution of sensor less random to accelerate the processing and to make it more practical.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

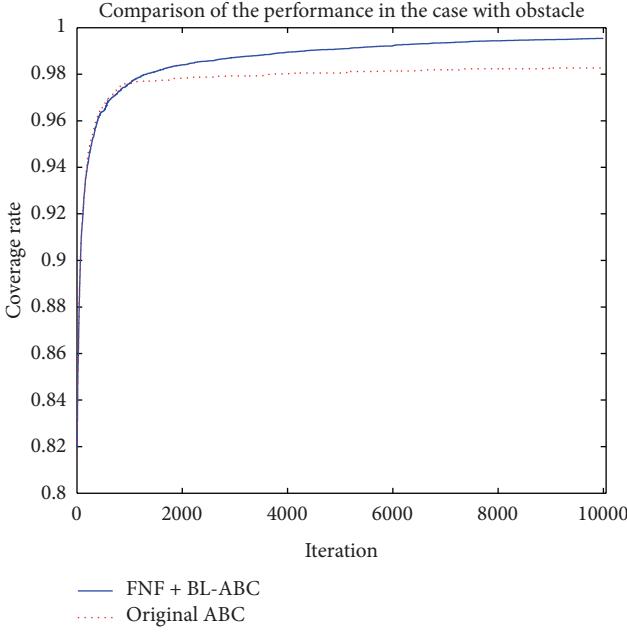


FIGURE 4: Coverage rate comparison between FNF-BL-ABC and original ABC for the room shown in Figure 3.

Acknowledgments

This work was supported by the Foundation for Distinguished Young Talents in Higher Education Guangdong, China (Grant no. LYM10011), National Natural Science Foundation of China (Grant no. 61201178), Scientific Project of Guangdong (no. 2010B010600019), and United Foundation Project of Guiyang University Guizhou (Grant no. QianKeHeJ-LKG[2013]36).

References

- [1] J. Zheng and A. Jamalipour, *Wireless Sensor Networks: A Network Perspective*, IEEE Press, New Jersey, NJ, USA, 2009.
- [2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [3] J. Chen, E. Shen, and Y. Sun, "The deployment algorithms in wireless sensor net works: a survey," *Information Technology Journal*, vol. 8, no. 3, pp. 293–301, 2009.
- [4] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, 2008.
- [5] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem," in *Proceedings of the 6th International Conference on Distributed Autonomous Robotic Systems*, pp. 299–308, Fukuoka, Japan, 2002.
- [6] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, pp. 1293–1303, New York, NY, USA, April 2003.
- [7] X. Yu, W. Huang, J. Lan, and X. Qian, "A novel virtual force approach for node deployment in wireless sensor network," in *Proceedings of IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '12)*, pp. 359–363, Hangzhou, China, 2012.
- [8] X. Yu, W. Huang, J. Lan, and X. Qian, "A van der Waals force-like node deployment algorithm for wireless sensor network," in *Proceedings of the 8th International Conference on Mobile Ad-Hoc and Sensor Networks*, pp. 218–221, Chengdu, China, 2012.
- [9] G. Wang, G. Cao, and T. F. La Porta, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 640–652, 2006.
- [10] D. Jourdan and O. L. de Weck, "Layout optimization for a wireless sensor network using a multi-objective genetic algorithm," in *Proceedings of the 59th IEEE Vehicular Technology Conference (VTC '04)*, vol. 5, pp. 2466–2470, May 2004.
- [11] X. Wang, S. Wang, and J. J. Ma, "An improved co-evolutionary particle swarm optimization for wireless sensor networks with dynamic deployment," *Sensors*, vol. 7, no. 3, pp. 354–370, 2007.
- [12] N. Kukunuru, B. Thella, and R. Davuluri, "Sensor deployment using particle swarm optimization," *International Journal of Engineering Science and Technology*, vol. 2, no. 10, pp. 5395–5401, 2010.
- [13] Z. Li and L. Lei, "Sensor node deployment in wireless sensor networks based on improved particle swarm optimization," in *Proceedings of IEEE International Conference on Applied Superconductivity and Electromagnetic Devices (ASEMD '09)*, pp. 215–217, Chengdu, China, September 2009.
- [14] W. H. Liao, Y. Kao, and Y. S. Li, "A sensor deployment approach using glowworm swarm optimization algorithm in wireless sensor networks," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12180–12188, 2011.
- [15] C. Ozturk, D. Karaboga, and B. Gorkemli, "Artificial bee colony algorithm for dynamic deployment of wireless sensor networks," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 20, no. 2, pp. 255–262, 2012.
- [16] C. Ozturk, D. Karaboga, and B. Gorkemli, "Probabilistic dynamic deployment of wireless sensor networks by artificial bee colony algorithm," *Sensors*, vol. 11, no. 6, pp. 6056–6065, 2011.
- [17] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, 2012.
- [18] W. Gao and S. Liu, "Improved artificial bee colony algorithm for global optimization," *Information Processing Letters*, vol. 111, no. 17, pp. 871–882, 2011.
- [19] W. Gao, S. Liu, and L. Huang, "A global best artificial bee colony algorithm for global optimization," *Journal of Computational and Applied Mathematics*, vol. 236, no. 11, pp. 2741–2753, 2012.
- [20] A. Rajasekhar, A. Abraham, and M. Pant, "Levy mutated artificial bee colony algorithm for global optimization," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC '11)*, pp. 655–662, Anchorage, Alaska, USA, October 2011.
- [21] H. Wang, "Improve artificial bee colony," *Computer Engineering and Design*, vol. 32, no. 11, pp. 3869–3872, 2011.
- [22] X. Bi and Y. Wang, "Artificial bee colony algorithm with fast convergence," *Systems Engineering and Electronics*, vol. 33, no. 12, pp. 2755–2761, 2011.
- [23] X. Yu and Z. Zhu, "A modified artificial bee colony algorithm with its applications in signal processing," *International Journal*

- of *Computer Applications in Technology*, vol. 47, no. 2-3, pp. 297–303, 2013.
- [24] Y. H. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *Proceedings of IEEE World Congress on Computational Intelligence (ICEC ’98)*, Evolutionary Computation Proceedings, pp. 69–73, Anchorage, Alaska, USA, May 1998.
 - [25] R. V. Kulkarni and G. K. Venayagamoorthy, “Particle swarm optimization in wireless-sensor networks: a brief survey,” *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 41, no. 2, pp. 262–267, 2011.
 - [26] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, “Opposition versus randomness in soft computing techniques,” *Applied Soft Computing*, vol. 8, no. 2, pp. 906–918, 2008.

