Paper:

An Improved Particle Swarm Optimization Deployment for Wireless Sensor Networks

Shuxin Ding, Chen Chen, Jie Chen, and Bin Xin

School of Automation, Beijing Institute of Technology
Key Laboratory of Intelligent Control and Decision of Complex Systems
5 Zhong Guan Cun South Street, Haidian District, Beijing 100081, China
E-mail: xiaofan@bit.edu.cn

[Received May 22, 2013; accepted December 24, 2013]

This paper addresses the issues associated with deployment of sensors, which are critical in wireless sensor networks. This paper provides an improved particle swarm optimization (PSO) algorithm by changing the basic form of PSO and introducing disturbance (d-PSO). By comparing with other PSO-based algorithms, simulation results show that the d-PSO algorithm provides a good-coverage solution with a satisfying coverage rate in a short time. This feature is especially useful for the rapid deployment of sensors.

Keywords: wireless sensor networks, deployment, particle swarm optimization, disturbance

1. Introduction

Wireless Sensor Networks (WSNs) are formed by small, inexpensive, low-powered sensors. The sensors communicate with each other wirelessly over a short distance [1].

WSNs have recently become a popular research area because of their promising application in numerous fields, especially in district monitoring. Each sensor has a sensing range, and with some sensors combined as WSNs, they can then detect an extended area. Therefore, WSNs are widely used in environment monitoring [2, 3].

However, there are some challenges in using WSNs because of their properties. For example, each sensor has a limited communication range and lifetime [4]. Therefore, the sensors need to be placed within a certain range for communication. For area monitoring, coverage is a problem. The sensors in WSNs are used for monitoring a region of interest (ROI). Therefore, an increase in the number of points detected in the ROI ensures better coverage effect of the sensors deployment. Meanwhile, the deployment algorithm should have an appropriate convergence speed considering about the computation time.

Numerous studies have been conducted to optimize sensor deployment. PSO algorithms are frequently used as an optimization algorithm to solve the deployment of WSNs [5]. Parallel particle swarm optimization (PPSO), which divides the ROI and the sensors equally into several parts, is proposed in [6], and it is used when there are large numbers of sensors to be deployed. Thus, the dimension of the searching space is partitioned to save time. In [7], a PSO-LA algorithm is proposed, and the velocity is changed by using learning automata (LA). In [8], an improved co-evolutionary PSO algorithm is proposed that combines virtual force and PSO with a co-evolutionary mechanism.

There are some computational geometry methods based on Delaunay triangulations and Voronoi diagrams [2, 4, 9]. In [10], a grid deployment algorithm is proposed with environmental factors in order to reach a minimum number of mobile nodes.

In this paper, the coverage problem is discussed, and an improved algorithm based on PSO is proposed. Some important issues investigated are coverage rate and convergence speed. The remainder of this paper is organized as follows. Section 2 contains the problem formulation, which gives the basic detection models, basic PSO, and other PSO-based algorithms. The d-PSO algorithm is introduced in Section 3. Simulations are introduced in Section 4. Finally, a conclusion of this paper and future work is discussed in Section 5.

2. Problem Formulation

2.1. Coverage Problem

In this paper, coverage rate is used as a way to evaluate the performance of WSN deployment. Therefore, the position of sensors is an important factor in determining the quality of the WSNs. Sensors should be placed reasonably according to the ROI such that the detecting ranges of the WSNs are fully utilized. The purpose of the coverage problem is to maximize the sensors coverage rate for a given ROI. In this paper, the ROI is an area described by a two-dimensional square.

We assume that there are n sensors deployed in the ROI at points $s_i(x_i, y_i)$, with detecting range r_i. In WSNs, there are two sensor detection models, the binary detection model [11] and the probabilistic detection model [12]. The probabilistic detection model is used in this paper. The detection model of the i-th sensor for the point $P(x, y)$ can be described as a probability function by distance. It is based on radio signal propagation models in which the signal strength decays with distance [13].

$$c_{x,y}(s_i) = \begin{cases} 0 & \text{if} \quad r_i + r_e \le d \\ e^{\left(-\alpha_1 \frac{\lambda_1^{\beta_1}}{\lambda_2^{\beta_2}} + \alpha_2\right)} & \text{if} \quad r_i - r_e < d < r_i + r_e \\ 1 & \text{if} \quad r_i - r_e \ge d \end{cases} \quad . \quad (1)$$

where d denotes the Euclidean distance between the point P and the location of the sensor. Thus, $d = \sqrt{(x-x_i)^2 + (y-y_i)^2}$. $r_e (r_e < r_i)$ measure the uncertainty of the detection. $\lambda_1 = r_e - r_i + d$, $\lambda_2 = r_e + r_i - d$. α_1, α_2, β_1, and β_2 are detection probability parameters. These values vary depending on the sensors types and characteristics.

In order to determine whether the point P is covered, it is better to calculate the probability of the point $P(x,y)$ from all the sensors in the ROI. Therefore, an overlapping sensor-detecting area increases the detection probability. The joint detection probability can be easily deduced as follows:

$$c_{x,y}\left(\sum_{i=1}^{n} s_i\right) = 1 - \prod_{i=1}^{n} (1 - c_{x,y}(s_i)). \quad . \quad . \quad . \quad (2)$$

The coverage rate is determined by calculating the effective detection probability of all the points in the ROI. However, there are an infinite number of points in the ROI. Therefore, the ROI can be expressed as a grid. The points in the ROI are sampled by two-dimensional uniformly distributed grids. The distance between two adjacent grids determines the number of points we consider in the ROI. This is a key factor of the calculation time and the accuracy of the coverage.

According to **Fig. 1**, the coverage rate is affected by the grid size. Therefore, the grid size should be chosen carefully in order to balance the calculation time and accuracy.

The coverage rate can be determined as follows [1]:

$$R = \frac{\sum_{j=1}^{n_p} c_{x_j,y_j}\left(\sum_{i=1}^{n} s_i\right)}{n_p}, \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (3)$$

where n_p is the number of grid points in the ROI.

There are several factors to consider in the coverage problem [2]. There may not be enough sensors to cover all the ROI. Since the sensors are cheap and limited in terms of their sensing range, they cannot easily cover the whole ROI. Some of them will die out because of limited access to power. Large-sensing-range sensors are very expensive to use. If this kind of sensor is somehow destroyed, the coverage of the ROI will decrease substantially. Meanwhile, it is easy for sensors to be deployed randomly in the ROI because of their mobility. For some dangerous and otherwise unreachable regions, they can initially be deployed remotely from the ROI. Therefore, the time for deployment may be an important issue as well.

The objective of the optimization in the paper is to maximize the coverage rate of the WSNs within a limited time.

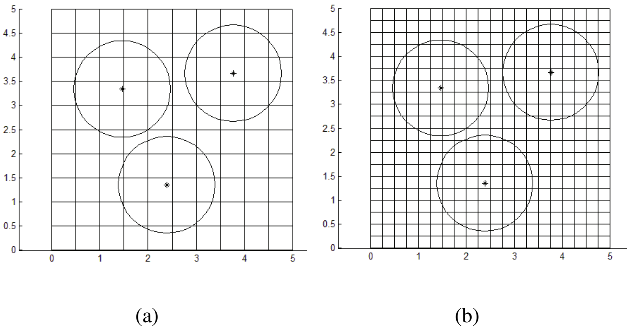

Fig. 1. Three sensor nodes deployed in gridded ROI: (a) grid size: 0.5×0.5; (b) grid size: 0.25×0.25.

2.2. Particle Swarm Optimization

The Particle Swarm Optimization (PSO) algorithm is an algorithm based on the social behavior of a flock of birds that was developed by Kennedy and Eberhart [14]. The motion of the particles is interpreted as birds flying. The particles move in the searching space according to their former speed, their experience, and the experience of their surrounding neighbors [2]. The d^{th} dimension i-th particle (x_{id}) represents a potential solution of the optimization. The dimension of the particle represents the number of variables that need to be optimized. The number of the particle n is set in advance, and the initial position and velocity (v_{id}) of the particles are randomly set according to certain constraints.

In the process of the PSO algorithm, the position and the velocity of each particle evolve as follows:

$$v_{id}(t+1) = w \times v_{id}(t) + c_1 \times rand() \times (p_{ibest} - x_{id}) + c_2 \times rand() \times (p_{gbest} - x_{id}) \quad . \quad . \quad (4)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (5)$$

where w is an inertia factor which ensures that the direction of motion of a particle is affected by its former velocity. It must be smaller than one and usually linearly decreases from 0.9 to 0.4 with respect to the time t. $rand()$ is an independent random number from 0 to 1. p_{ibest} represents the best position ever found for the i-th particle and p_{gbest} represents the global best position. c_1 and c_2 are respectively the cognitive factor and social factor which control the motion of the particle to its personal best position and global best position. The position of the particle is renewed by the velocity. The PSO algorithm will stop when the maximum number of iterations is met.

The best position is defined by a fitness function, which evaluates the position quality of a particle, and p_{ibest} and p_{gbest} are replaced according to it. For the coverage problem, the fitness function is the coverage rate. It should be noted that all the particles have the ability to memorize their personal best positions and the best positions of their neighbors.

Any WSN will constitute of numerous sensors. The searching space for this optimization problem will increase rapidly. For a high-dimensional optimization problem, the calculation time will increase as well. However, because all the components in one particle in the PSO algorithm are moving, there may be arise a situation where some components move closer to the optimal position while others move away from the optimal; the algorithm provides a better solution for a larger coverage rate. This is the so-called “two steps forward, one step back” process [15]. In this process, a local convergence situation is met.

Meanwhile, the velocity of the particle is affected by its earlier motion; this may speed up the local convergence situation because its earlier motion may not have been optimal. Therefore, an improved algorithm is proposed in the next section.

2.3. Other PSO-Based Algorithms

The virtual force PSO algorithm [8] is used for sensor deployment. For each sensor, we consider the force exerted on it by other sensors, which eventually forces the sensors to depart from each other. The force can be an attracting or a repelling force. The total force on sensor s_i can be expressed as

$$\vec{F}_i = \sum_{j=1, j\neq i}^{n} \vec{F}_{ij}, \quad \ldots \ldots \ldots \quad (6)$$

where n is the number of sensors.

The virtual force exerted on sensor s_i by sensor s_j can be expressed as

$$\vec{F}_{ij} = \begin{cases} 0 & \text{if} \quad d_{ij} \geq C \\ \left(w_A\left(d_{ij} - d_{th}\right), \alpha_{ij}\right) & \text{if} \quad C > d_{ij} > d_{th} \\ 0 & \text{if} \quad d_{ij} = d_{th} \\ \left(w_R\left(\frac{1}{d_{ij}} - \frac{1}{d_{th}}\right), \alpha_{ij} + \pi\right) & \text{if} \quad d_{ij} < d_{th} \end{cases} \quad (7)$$

where C is the communication range, α_{ij} is the orientation of the line from s_i to s_j, and w_A (w_R) describes the effect of the attractive (repulsive) force.

For virtual force particle swarm optimization (VFPSO), the velocity-renewing equation is given as

$$v_{id}(t+1) = w \times v_{id}(t) + c_1 \times rand() \times (p_{ibest} - x_{id}) + c_2 \times rand() \times (p_{gbest} - x_{id}) + q(d) \quad (8)$$

$$q(d) = \begin{cases} \dfrac{F_x\left(\frac{j+1}{2}\right)}{F_{xy}\left(\frac{j+1}{2}\right)} \times MaxStep \times e^{\frac{-1}{F_{xy}\left(\frac{j+1}{2}\right)}} \\ j = 1, 3, \ldots, 2n-1 \\ \dfrac{F_y\left(\frac{j}{2}\right)}{F_{xy}\left(\frac{j}{2}\right)} \times MaxStep \times e^{\frac{-1}{F_{xy}\left(\frac{j}{2}\right)}} \\ j = 2, 4, \ldots, 2n \end{cases} \quad (9)$$

The co-evolutionary PSO (CPSO) [15] algorithm improves the searching ability of PSO in high-dimensional problems. In CPSO, the searching space is partitioned into several one-dimensional subspaces. The solution vector is split into several values. For an n-dimensional problem, CPSO converts it into n one-dimensional problems. Further, it can combine with VF to create VFCPSO [8].

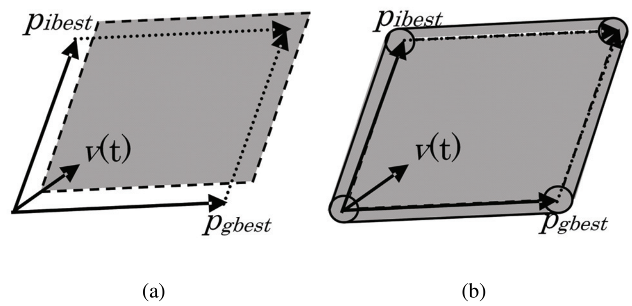

Fig. 2. Searching space between (a) PSO algorithm and (b) d-PSO algorithm.

3. Proposed Algorithm

In this section, a deployment algorithm called d-PSO is proposed to overcome the disadvantages of the PSO algorithm with local convergence and time consumption. This algorithm is based on PSO, but has a faster and more global solution.

In WSNs, we assume that there are n sensors in a two-dimensional ROI. The position for one sensor can be described using a coordinate system as (x_i, y_i). Therefore, a particle for n sensors can be represented as $(x_1, y_1, x_2, y_2, x_3, y_3, \ldots, x_n, y_n)$. The particles are $2n$-dimensional for n sensors.

Because of the drawbacks of the PSO algorithm, this algorithm changes the velocity form of the canonical PSO. It deletes the former velocity part and adds a disturbance to the velocity, which is given as follows:

$$v_{id}(t+1) = c_0 \times randn() + c_1 \times rand() \times (p_{ibest} - x_{id}) + c_2 \times rand() \times (p_{gbest} - x_{id}), \quad \ldots \quad (10)$$

where c_0 is the amplitude of the disturbance, and the function $randn()$ is a standard normal distribution with average zero and standard deviation one. The update formula of the position is the same with PSO.

Figure 2 describes the difference of the position tendency between the PSO algorithm and the d-PSO algorithm. Here, c_1 and c_2 are set to 1, c_0 is set according to the number of sensors, the sensing range, and the space size.

From **Fig. 2**, it is obvious that the searching space in the PSO algorithm is smaller than the d-PSO algorithm. The shadow area in **Fig. 2(a)** only considers more about the direction of its velocity. However, in **Fig. 2(b)**, the shadow area contains parts of the shadow in **Fig. 2(a)**, but has spaces away from its original direction as well.

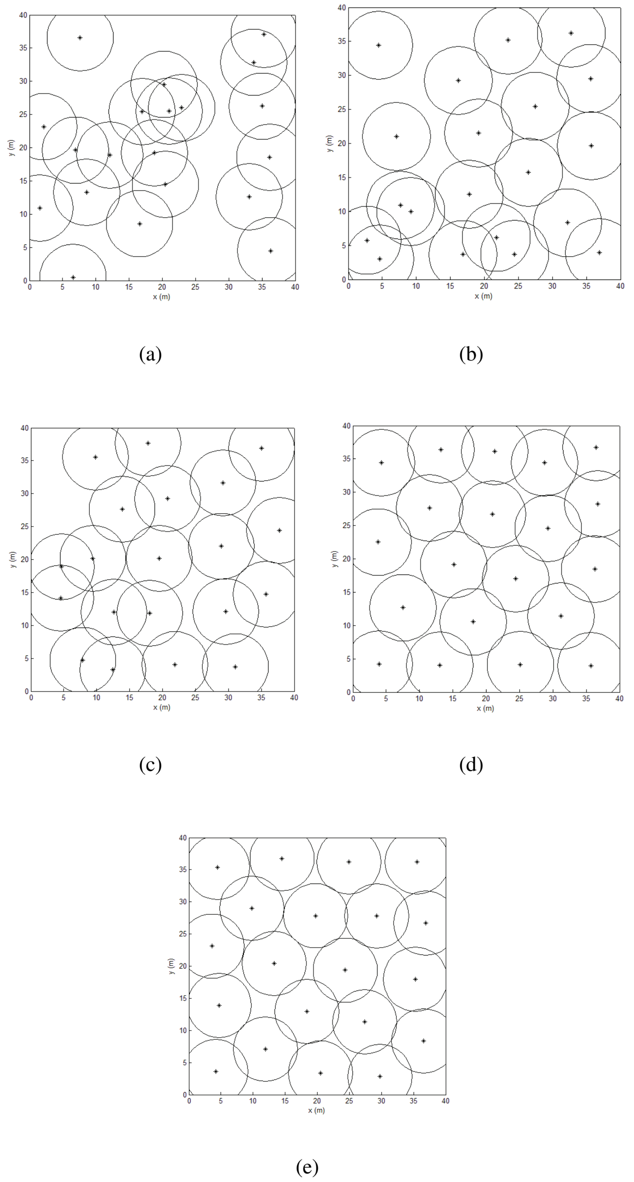

Fig. 3. Deployment after (a) initial random placement, (b) PSO, (c) VFPSO, (d) VFCPSO, and (e) d-PSO.

This unique feature ensures that the result will not lead to a local optimal, as in PSO algorithm, because the personal best and global best position may be a suboptimal position. The disturbance in **Fig. 2(b)** makes it possible for the particle to jump away from the local optimal position. It should be noted that the searching probability of the outer shadow introduced by the disturbance is not uniformly distributed, since the disturbance is normally distributed. The outer shadow is $3c_0$ in width, since for $N(0,1)$, $P(-3 < x < 3) = 2\Phi(3) - 1 = 99.7\%$.

The factor c_0 plays an important role in influencing particles to converge to the global optimized solution. If this factor is extremely large or small, this algorithm will perform badly.

This paper presents four algorithms related to PSO. The PSO algorithm is a simple and effective algorithm, as are both VFPSO and VFCPSO. However, the random component in d-PSO brings a nonzero velocity, which provides the possibility of improving a suboptimal solution to acquire the global optimal solution. For VFCPSO, its high computation time may be a big problem, especially when there are a large number of WSNs.

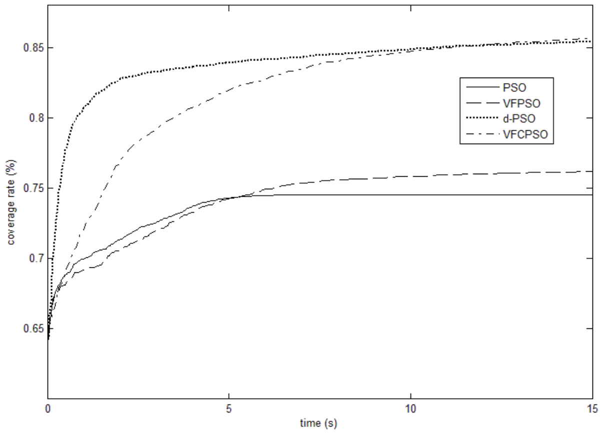

Fig. 4. The coverage rate of d-PSO and other algorithms during the iterations with random initial position.

4. Simulation Results

In order to test the performance of the d-PSO algorithm, several simulations are conducted as follows. The simulation is implemented on an Intel Core i5-3470 CPU (3.2 GHz) PC using MATLAB R2013a.

4.1. Deployment Performance

The object of this experiment is to test the performance of the d-PSO algorithm with other PSO-based algorithms (PSO, VFPSO, and VFCPSO). First, the situation is considered where there are $n = 20$ sensors in a WSN, so the dimension is $d = 2 \times n = 40$. The area of the ROI is 40×40 m^2. There are 20 particles in the entire algorithm. The detection range is $r = 5$ m. $w = 0.9 - 0.4$ (linearly decreasing with t) for PSO. $c_0 = 0.4$, $c_1 = c_2 = 1.4962$, and the $maximum\ \ iteration = 1000$. For the probabilistic detecting model, $r_e = 0.1\ r$, $\alpha_1 = 1$, $\alpha_2 = 0$, $\beta_1 = 1$, and $\beta_2 = 0.5$. The parameters for virtual force are set as $w_A = 1$, $w_R = 5$, $d_{th} = 2\ r$, $C = 3\ r$, and $MaxStep = 0.5\ r$. The size of grid is set to 1×1 m^2, so there are 1600 grids to determine coverage rate.

Figure 3 shows the result of the d-PSO algorithm and the other PSO-based algorithms with the same initial placement. The coverage rate for the initial placement in **Fig. 3(a)** is 64.04%, and the result by PSO, VFPSO, VFCPSO, and d-PSO in **Figs. 3(b), (c), (d)**, and **(e)** are 75.44%, 76.95%, 86.17%, and 86.87%.

It is obvious that d-PSO presents a better deployment solution than PSO and VFPSO, and is similar to VFCPSO. 50 experiments are conducted independently with random initial states are given in **Fig. 4** and **Table 1**.

Table 1. The average coverage rate and its standard deviation with random initial position.

Algorithm	PSO	VFPSO	d-PSO	VFCPSO
Average coverage rate (%)	74.56	76.30	85.49	86.70
Standard Deviation (%)	1.95	1.68	0.77	0.57

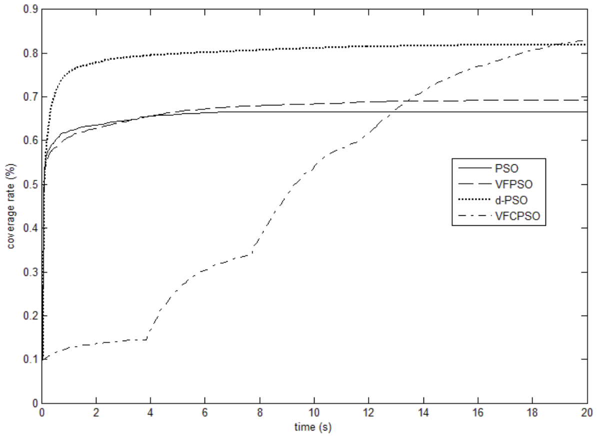

Fig. 5. The coverage rate of d-PSO and other algorithms during the iterations with bad initial position.

Figure 4 and **Table 1** shows that the d-PSO algorithm converges faster than the other PSO-based algorithms, but the coverage is less than that of VFCPSO. However, d-PSO demonstrates the ability to converge to a satisfying result in a short time. This ability enables it to be used in fast deployment applications.

4.2. The Impact of Initial Position

In Section 4.1, d-PSO shows a faster convergence speed than other algorithms. In Section 4.2, bad initial positioning of the algorithm is discussed.

The initial position of the sensors is not random. For example, some dangerous or unreachable region for humans. Or for dynamic deployment, the ROI is changing.

With the same parameters in Section 4.1, the initial coverage is lower than random deployment.

Figure 5 and **Table 2** shows that the d-PSO algorithm still has a good convergence speed compared to other algorithms. However, VFCPSO takes enormous time to converge. It takes VFCPSO 19 s to catch up with d-PSO. Although VFCPSO shows a good coverage rate, it takes too much time to converge. Therefore, d-PSO is better when time is limited.

Table 2. The average coverage rate and its standard deviation with bad initial position.

Algorithm	PSO	VFPSO	d-PSO	VFCPSO
Average coverage rate (%)	66.65	69.34	81.82	82.51
Standard Deviation (%)	3.02	3.30	4.46	1.59

5. Conclusion and Future Work

This paper presented an improved deployment algorithm called d-PSO. It is used to solve the deployment problem of WSNs. This algorithm delivers a better coverage rate within a short span of time, which is especially important in the rapid deployment of sensors.

In future studies, the convergence property of the algorithm and the factor of the disturbance will be studied thoroughly, and the description of the ROI and sensors may be more detailed.

Acknowledgements

This work was supported by the National Nature Science Foundation of China under Grant 61203181 and by the Beijing Outstanding Ph.D. Program Mentor under Grant 20131000704. The authors thank the anonymous reviewers and editors for their constructive comments on this paper.

References:

[1] A. S. Majid, and E. Joelianto, "Optimal sensor deployment in non-convex region using Discrete Particle Swarm Optimization algorithm," IEEE Conf. on Control, Systems & Industrial Informatics, pp. 109-113, 2012.

[2] N. Aziz, A. W. Mohemmed, and M. Y. Alias, "A wireless sensor network coverage optimization algorithm based on particle swarm optimization and Voronoi diagram," IEEE Int. Conf. on Networking, Sensing and Control, pp. 602-607, 2009.

[3] A. M. Elmogy, F. Karray, and A. M. Khamis, "Auction-Based Consensus Mechanism for Cooperative Tracking in Multi-Sensor Surveillance Systems," J. of Advanced Computational Intelligence and Intelligent Informatics, Vol.14, No.1, pp. 13-20, 2010.

[4] N. Aziz, A. W. Mohemmed, and D. Sagar, "Particle swarm optimization and Voronoi diagram for wireless sensor networks coverage optimization," IEEE Int. Conf. on Intelligent and Advanced Systems, pp. 961-965, 2007.

[5] R. V. Kulkarni and G. K. Venayagamoorthy, "Particle swarm optimization in wireless-sensor networks: A brief survey," IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol.41, No.2, pp. 262-267, 2011.

[6] X. Wang, S. Wang, and J. Ma, "Dynamic deployment optimization in wireless sensor networks," Intelligent Control and Automation. Springer Berlin Heidelberg, pp. 182-187, 2006.

[7] R. Soleimanzadeh, B. J. Farahani, and M. Fathy, "PSO based deployment algorithms in hybrid sensor networks," Int. J. Comput. Sci. Netw. Secur, Vol.10, pp. 167-171, 2010.

[8] X. Wang, S. Wang, and 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.

[9] F. Jing, W. Qiong, and H. Jun Feng, "Optimal deployment of wireless mesh sensor networks based on Delaunay triangulations," IEEE Int. Conf. on Information Networking and Automation, Vol.1, pp. 370-374, 2010.

[10] R. C. Luo and O. Chen. "Mobile sensor node deployment and asynchronous power management for wireless sensor networks," IEEE Trans. on Industrial Electronics, Vol.59, No.5, pp. 2377-2385, 2012.

[11] K. Chakrabarty, S. S. Iyengar, H. Qi et al, "Grid coverage for surveillance and target location in distributed sensor networks," IEEE Trans. on Computers, Vol.51, No.12, pp. 1448-1453, 2002.

[12] X. Wang and S. Wang, "Hierarchical deployment optimization for wireless sensor networks," IEEE Trans. on Mobile Computing, Vol.10, No.7, pp. 1028-1041, 2011.

[13] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks," ACM Trans. on Embedded Computing Systems (TECS), Vol.8, No.1, pp. 61-91, 2004.

[14] J. Kennedy and R. Eberhart, "Particle swarm optimization," IEEE Int. Conf. on Neural Networks, pp. 1942-1948, 1995.

[15] F. Van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," IEEE Trans. on Evolutionary Computing, Vol.8, No.3, pp. 225-239, 2004.

Name:
Shuxin Ding

Affiliation:
Ph.D. Student, School of Automation, Beijing Institute of Technology

Address:
5 Zhong Guan Cun South Street, Haidian District, Beijing 100081, China

Brief Biographical History:
2012 Received B.E. degree from Beijing Institute of Technology
2012- Ph.D. Student, Beijing Institute of Technology

Main Works:
• Optimization and its applications

Name:
Chen Chen

Affiliation:
Lecturer, School of Automation, Beijing Institute of Technology

Address:
5 Zhong Guan Cun South Street, Haidian District, Beijing 100081, China

Brief Biographical History:
2004 Received B.E. degree from Beijing Institute of Technology
2009 Received Ph.D. degree from Beijing Institute of Technology

Main Works:
• Complicated system multi-object optimization and distributed simulation

Name:
Jie Chen

Affiliation:
Professor, Beijing Institute of Technology

Address:
206 Lab., No.5 Yard, Zhong Guan Cun South Street, Haidian District, Beijing 100081, China

Brief Biographical History:
1986 Received B.E. degree in control theory and control engineering from Beijing Institute of Technology
1989-1990 Visiting scholar in California State University
1996 Received M.E. degree in control theory and control engineering from Beijing Institute of Technology
1996-1997 Research fellow in school of E&E, the University of Birmingham
1998- Professor of control science and engineering, Beijing Institute of Technology
2000 Received Ph.D. degree in control theory and control engineering from Beijing Institute of Technology

Main Works:
• Complex system multi-objective optimization and decision, intelligent control, constrained nonlinear control, and optimization methods

Membership in Academic Societies:
• Senior Member, The Institute of Electrical and Electronics Engineers (IEEE)
• Member, The 6th Review Panel on Control Science and Engineering
• Chinese State Council Academic Degrees Committee
• Executive Member and Deputy Director, The Technical Committee on Control Theory, Chinese Association of Automation
• Executive Member, The Chinese Association for Artificial Intelligence
• Member, The Academic Committee for the State Key Laboratory of Industrial Control Technology

Name:
Bin Xin

Affiliation:
Lecturer, School of Automation, Beijing Institute of Technology

Address:
5 Zhong Guan Cun South Street, Haidian District, Beijing 100081, China

Brief Biographical History:
2004 Received B.E. degree from Beijing Institute of Technology
2012 Received Ph.D. degree from Beijing Institute of Technology

Main Works:
• Intelligent optimization, combinatorial optimization, and evolutionary computation