Proceedings of the Ninth China-Japan International Workshop
on Internet Technology and Control Applications
Beijing, China, 28 June-1 July 2013

An improved deployment algorithm for wireless sensor networks based on Particle Swarm Optimization

Shuxin Ding[1,2], Jie Chen[1,2], Chen Chen[1,2], Bin Xin[1,2]
[1] Department of Automatic Control, Beijing Institute of Technology
Beijing, 100081, China
[2] Key Laboratory of Complex System Intelligent Control and Decision, Ministry of Education, Beijing 100081

Abstract

The main issue concerned in this paper is the coverage problem, which is a critical issue in wireless sensor networks deployment. Particle Swarm Optimization (PSO) algorithm is frequently used for deployment, but it may lead to local optimal solution. In this paper, an improved PSO algorithm is given by changing the basic form of PSO and introducing disturbance (d-PSO), which brings a better coverage result. Simulation results show that the d-PSO algorithm preforms a better coverage solution than PSO algorithm.

Keywords: Wireless Sensor Networks, Deployment, Particle Swarm Optimization, Disturbance.

1. INTRODUCTION

Wireless Sensor Networks (WSNs) are formed by some sensors, which are small in size, low in power and cost. The sensors are communicating with each other without wires over a short distance [1].

WSNs have recently become a popular research area, since their promising applications in many fields, especially district monitoring. A sensor has a sensing range, with some sensors combined as WSNs, they can detect an area together. Therefore, WSNs are widely used in environment monitoring [2].

However, there are some challenges in WSNs, due to their properties. For example, for a sensor, it has limited communication range, and lifetime [3]. Therefore, the sensors should be placed within certain range for communication. For the area monitoring, a challenge is the coverage problem. The sensors in WSNs are used for monitoring a region of interest (ROI). Therefore, the more points in the ROI are detected, the better coverage effect of the sensors deployment.

In this paper, the coverage problem is mainly discussed and an improved algorithm is proposed. This paper is organized as follows. Section 2 is a review of the related works of the coverage problems in WSNs. Section 3 contains the problems formation, which gives the basic detection models, basic PSO algorithm. The d-PSO algorithm is introduced in Section 4. Simulations are introduced in Section 5. Finally, a conclusion and future work of this paper is discussed in Section 6.

2. RELATED WORKS

A lot of studies have done to analysis the optimization of the sensors deployment. PSO algorithms are frequency used as an optimization algorithm to solve WSNs deployment [4]. When there are large numbers of sensors, [5] proposed a parallel particle swarm optimization (PPSO) which divided the ROI and the sensors equally into several parts, so the dimensional of the searching space is cut down to save time. In [6], a PSO-LA algorithm is proposed, the velocity is changed using some knowledge. In [7], an improved co-evolutionary PSO algorithm is proposed which combines virtual force and PSO with a co-evolutionary mechanism.

There are some geometry methods based on Delaunay Triangulations and Voroni Diagram [8]-[10]. In [11], a grid deployment algorithm is proposed with environmental factors to reach a minimum of the mobile node.

3. PROBLEMS FORMATION

3.1 Coverage Problem

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

We can assume that there are n sensors deployed in the ROI at point $s_i(x_i,y_i)$, the detecting range is r_i. The detection model of the i^{th} sensor for the point $P(x,y)$ can be described as a probability function by distance [1].

$$c_{x,y}(s_i)=\begin{cases}0 & \text{if } r_i<d\\ 1 & \text{if } r_i\geq d\end{cases} \tag{1}$$

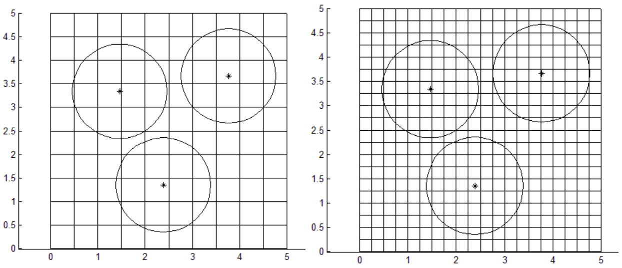
where d is the distance between the point P and the location of the sensor. So, $d = \sqrt{(x - x_i)^2 + (y - y_i)^2}$.

In order to determine the point P_j is covered or not, it is better to calculate the probability of the point $P(x,y)$ from all the sensors in the ROI. It can be easily deduced as follows:

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

The coverage rate is determined by calculating the detection probability of all the point in the ROI. However, there are infinite numbers of points in the ROI. Therefore, the ROI can be expressed in a gridded way. The points in the ROI are sampled by some two-dimensional uniformly distributed grids. The distance between two adjacent grids determine the number of points we consider in the ROI, which is a key factor of the calculation time and the accuracy of the coverage.

The following is a figure which shows the difference of the size of grids.



(a) (b)

Fig.1 3 sensor nodes deployed in gridded ROI: (a) grid size: 0.5×0.5; (b) grid size: 0.25×0.25

According to the Fig.1, the coverage rate is determined in different grid size. Therefore, the grid size should be chosen carefully which balances the calculation time and accuracy.

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

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

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

The objective of the optimization in the paper is to maximize the coverage rate of the WSNs.

3.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) algorithm is a algorithm based on the social behavior of a flock of birds developed by Kennedy and Eberhart [12]. The motions of the particles are regarded as the birds' flying. The particles are moving in the searching space according to their former speed, their experience and the experience of their neighbors around [2]. For the dth dimension i^{th} particle (x_{id}) represents a potential solution of the optimization. The dimension of the particle represents the number of the objectives needed 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 within some restrictions.

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

$$\begin{aligned} v_{id}(t+1) = w \times v_{id}(t) + c_1 \times rand() \times (p_{ibest} - x_{id}) + \\ c_2 \times rand() \times (p_{gbest} - x_{id}) \end{aligned} \qquad (4)$$

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

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

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

w is a inertia factor which ensure that the motion direction of a particle is affected by its former velocity. It must be smaller than 1 and usually linearly decreasing from 0.9 to 0.4 with respect to 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 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 renews by the velocity. The PSO algorithm will stop when a maximum 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 neighbors' best positions.

In fact, there will be a number of sensors in WSNs. The searching space for this optimization problem will increase rapidly. For a high-dimensional optimization problem, the calculation time will increase as well. However, since all the components in one particle in PSO algorithm is moving, there may be bad situation that some components move closer to the optimal position while others may move away from the optimal, while it gives a better solution for larger coverage rate. This situation is the so-called "two step forward, one step back" [13]. In this process, a local convergence situation is met.

Meanwhile, since the velocity of the particle is affected by its former one, it may speed up the local convergence situation since the former motion may not be optimal. Therefore, an improved algorithm is proposed.

4. PROPOSED ALGORITHM

In this section, a deployment algorithm called d-PSO is proposed to overcome the disadvantages of PSO algorithm with local convergence and time-consuming.

This algorithm is based on PSO, and has a more global and faster solution.

In WSNs, assuming that there are n sensors in a two-dimensional ROI. The position for one sensor can be described in the coordinate 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.

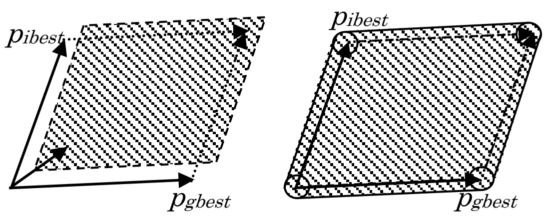
The fitness function has been mentioned before, which is the coverage rate of the n sensors.

Due to the drawbacks of the PSO algorithm, the algorithm proposed here 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:

$$\begin{aligned} v_{id}(t+1) = c_0 \times randn() + c_1 \times rand() \times (p_{ibest} - x_{id}) + \\ c_2 \times rand() \times (p_{gbest} - x_{id}) \end{aligned} \quad (6)$$

where c_0 is a factor to describe the amplitude of the disturbance, and the function *randn*() is a standard normal distributed with average 0 and standard deviation 1. The update formula of the position is the same with PSO.

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



(a) (b)

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

From Fig. 2, it is obviously 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 also space away from its original direction. This unique feature ensures that the result will not lead to somehow local optimal as PSO algorithm. Since the personal best and global best position may be the suboptimal position. The disturbance in Fig. 2(b) makes it possible that the particle is able 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 normal distributed. The outer shadow is $3c_0$ in width, since for $N(0,1)$, $P(-3<x<3)=2\Phi(3)-1=99.7\%$.

For c_0, it can be set as follows:

$$c_0 = \frac{cL}{nr} \quad (7)$$

where c is a constant number set as 1, ROI is a $L \times L$ square area. n is the number of the sensors, r is the detection range of one sensor.

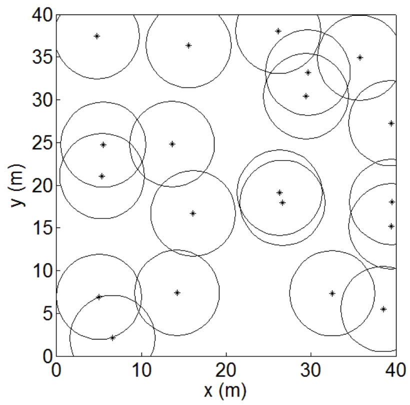
This factor plays an important role for particles to converge to the global optimized solution. If this factor is too big or too small, this algorithm will perform badly.

5. SIMULATION RESULTS

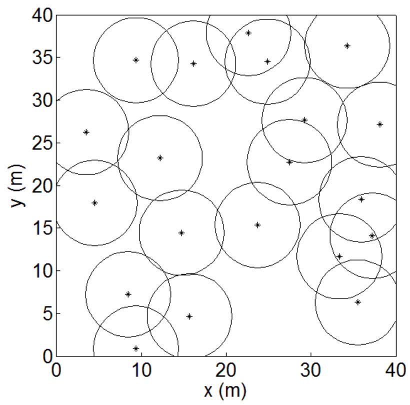
In order to test the performance of the d-PSO algorithm, several simulation results are conducted as follows. The simulation is implemented on an Intel Core i5-3470 CPU (3.2GHz) PC using MATLAB R2013a.

5.1 Deployment performance

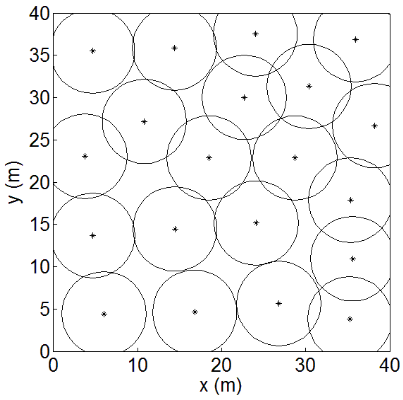
The object of this experiment is to test the performance of the d-PSO algorithm with the PSO algorithm. First, a situation is considered that there are n=20 sensors in WSNs, so the dimension is $d=2\times n=40$. The area of the ROI is $40\times 40\text{m}^2$. There are 20 particles in the entire algorithm. The detection range is r=5m. w=0.9 - 0.4 (linearly decreasing with t) for PSO algorithm. c=0.75, $c_1=c_2$=1.4962, *maximum iteration* = 600. The size of grid is set 1×1, so there are 1600 grids to determine coverage rate.



(a)



(b)



(c)

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

Fig. 3 shows the result of the d-PSO algorithm and PSO algorithm with the same initial placement. The coverage rate for the initial placement in Fig. 3(a) is 65.97%, and the result by PSO and d-PSO in Fig. 3(b) and 3(c) are 74.30% and 84.95%. The execution time for PSO and d-PSO are 7.31s and 6.82s.

It is obviously that d-PSO presents a better deployment solution than PSO. The following figure shows the coverage rate with respect to iteration times.

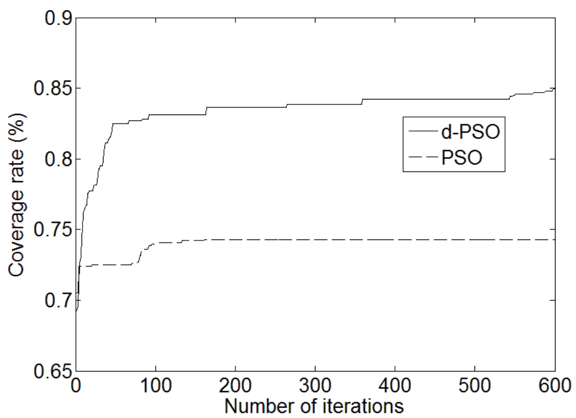


Fig.4 The coverage rate of PSO and d-PSO during the iterations.

The result shows that the PSO algorithm converges faster than the d-PSO algorithm. But the d-PSO algorithm seems to find better solution when the iterations are big. It is because there is no mechanism like the inertia factor *w*, which limits the velocity to vary. And the disturbance always allows the particle to find better solutions.

In order to confirm the robustness of the PSO and d-PSO algorithm, 50 experiments are conducted independently with random initial states. The average coverage rate, its standard deviation and average execution time are shown in the following table.

Table 1. The average coverage rate, its standard deviation and execution time

Algorithm	PSO	d-PSO
Average coverage rate (%)	74.67	84.88
Standard Deviation (%)	2.04	1.08
Average execution time (s)	7.13	6.98

The result shows that the d-PSO algorithm seems to be better in effectiveness and robustness.

5.2 Analysis of the particle numbers

The PSO algorithm will perform better when there a large numbers of particles. The following simulations provides information that the effectiveness of the d-PSO algorithm with different amount of particles.

The simulation has the same condition with Section 5.1 with different amount of particles from 1 to 40, and 50 independent experiments are conduction for each particle.

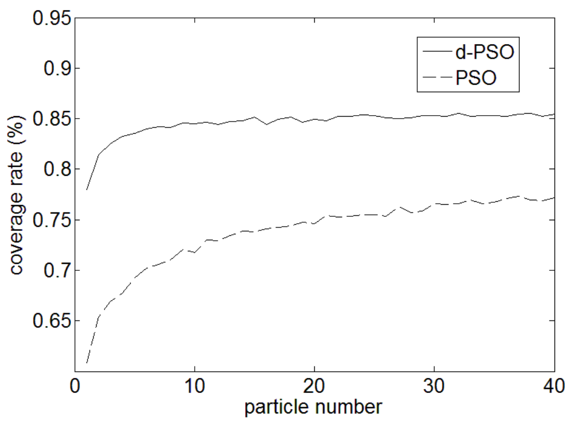


Fig.5 The coverage rate of PSO and d-PSO with different amount of particles

Fig. 5 shows that the d-PSO algorithm has an acceptable solution when there are few particles, while PSO algorithm seems to be better when particle number is growing. It is obviously that the execution time will cut down with fewer particles using d-PSO, and the solution will be acceptable. Therefore, the d-PSO algorithm is time-saving than PSO.

6. CONCLUSION & FUTURE WORK

This paper presented an improved deployment algorithm called d-PSO algorithm. It is used to solve the coverage problem of WSNs. This algorithm performs a better coverage rate with less swarm nodes than PSO algorithm. Rather than PSO, d-PSO performs a better global searching ability. In the future studies, the convergence property of the algorithm and the factor of the disturbance can be studied thoroughly, and the description of the ROI and sensors may be more complicated.

References

[1] Majid, Agung S., and Endra Joelianto. “Optimal sensor deployment in non-convex region using Discrete Particle Swarm Optimization algorithm”, IEEE Conference on Control, Systems & Industrial Informatics, 2012. pp. 109-113.

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

[3] Aziz N, Mohemmed A W, Sagar D. “Particle swarm optimization and Voronoi diagram for wireless sensor networks coverage optimization”, IEEE International Conference on Intelligent and Advanced Systems, 2007. pp. 961-965.

[4] Kulkarni R V, Venayagamoorthy G K. “Particle swarm optimization in wireless-sensor networks: A brief survey”, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2011, Vol.41, Issue.2. pp. 262-267.

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

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

[7] Wang X, Wang S, Ma, J. “An Improved Co-evolutionary Particle Swarm Optimization for Wireless Sensor Networks with Dynamic Deployment”, *Sensors* 7, 2007, no. 3, pp. 354-370.

[8] Jing F, Qiong W, JunFeng H. “Optimal deployment of wireless mesh sensor networks based on Delaunay triangulations”, IEEE International Conference on Information Networking and Automation, 2010, Vol.1, pp. V1-370-V1-374.

[9] Aziz N, Mohemmed A W, Sagar D. “Particle swarm optimization and Voronoi diagram for wireless sensor networks coverage optimization”, IEEE International Conference on Intelligent and Advanced Systems., 2007, pp. 961-965.

[10] Aziz N, Mohemmed A W, Alias M Y. “A wireless sensor network coverage optimization algorithm based on particle swarm optimization and Voronoi diagram.” IEEE International Conference on Networking, Sensing and Control, 2009, pp. 602-607.

[11] Luo R C, Chen O. “Mobile sensor node deployment and asynchronous power management for wireless sensor networks”, IEEE Transactions on Industrial Electronics, 2012, Vol.59, Issue.5, pp. 2377-2385.

[12] Kennedy J, Eberhart R. “Particle swarm optimization”, IEEE International Conference on Neural Networks, 1995, Vol.4, pp. 1942-1948.

[13] Van den Bergh F., Engelbrecht A. P. “A cooperative approach to particle swarm optimization”, IEEE Trans. on Evolut. Comput. 2004, Vol.8, Issue.3, pp. 225-239.