

Sensor Node Deployment in Wireless Sensor Networks based on Ionic Bond-Directed Particle Swarm Optimization

Haiping Huang^{1,2,3,*}, Junqing Zhang^{1,2}, Ruchuan Wang^{1,2,3} and Yisheng Qian¹

¹ College of Computer, Nanjing University of Posts and Telecommunications, 210003, Nanjing, China

² Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, 210003, Nanjing, China

³ Key Lab of Broadband Wireless Communication and Sensor Network Technology of Ministry of Education, Nanjing University of Posts and Telecommunications, 210003, Nanjing, China

Received: 21 Mar. 2013, Revised: 22 Jul. 2013, Accepted: 24 Jul. 2013

Published online: 1 Mar. 2014

Abstract: Sensor node deployment is one of the critical topics addressed in wireless sensor networks (WSNs) research, which determines coverage efficiency of WSNs. This paper proposes a self-organizing algorithm for enhancing the coverage for WSNs, which is so-called Ionic bond-directed particle swarm optimization (IBPSO). The proposed algorithm combines the ionic bond method with particle swarm optimization (PSO), where ionic bond method uses a judicious ionic bond between two sensor nodes to determine which node needs to move and also the path and direction of the movement and PSO is suitable for solving multi-dimension function optimization in continuous space. Simulation results demonstrate that IBPSO has more satisfactory performance on regional convergence and global searching than PSO algorithm and can implement dynamic deployment of WSNs more efficiently and rapidly.

Keywords: Wireless sensor networks, Sensor node deployment, Ionic bond, Particle swarm optimization.

1. Introduction

The sensory ability of WSNs to physical world is embodied in coverage which is often used to describe the monitoring standard of Quality of Service (QoS) [1,2]. Two key issues in mobile Wireless Sensor Networks (WSNs) are coverage and energy conservation. A high coverage rate ensures a high quality of service of the WSNs. These two issues are correlated, as coverage improvement in mobile WSNs requires the sensors to move, which is one of the main factors of energy consumption. Therefore sensor node deployment optimization in mobile WSNs has become a critical problem in wireless sensor network applications.

Some previous works model the mobile sensors as the electrons [3,4] or molecules [5,6] to avoid uneven deployment where sensor nodes modeled as cluster architecture. Virtual force algorithm recently emerges as one of main approaches for dynamic deployment [7]. The received signal strength of this message is treated as the force which pushes each other. The deploying procedure

finishes when the forces work on every sensor are balanced. Sensors in this model may need oscillation moving that sensors move back and forth over a small region to adjust their positions before the force trends balance. It is not energy efficiency for those energy limited sensors. The authors of article [8] modeled the deploying procedure as that of building the ionic bonds between ions. Sensors are ions, and the links between them are the ionic bonds. Sensors do not need to have their respective position information. They are only required the abilities to identify the directions of incoming signals and accurately estimate their distances to neighbors. These are two essential abilities in general self-deploying methods. One satisfactory deployment method can effectively maximize the coverage and minimize the deploying time. However, the Ion-6 method in [8] needs to fix the nodes' positions continually in order to form the hexagon topology, which would influence the performance on global optimization.

PSO is a search algorithm which can be used to look for optimal solution in a given search space. It is based on

* Corresponding author e-mail: hhp@njupt.edu.cn

how a flock of birds work together to find food in an area. These birds, directed by the results of their own searches and other birds' successes, will move around the search space to find food. The birds are represented in PSO algorithm by a swarm of particles. Wu used PSO to optimize coverage in a mobile WSNs and reduce the communication energy consumption in cluster based sensor networks by electing the best set of cluster heads in [9]. The coverage is evaluated by grid-based fitness function. Another algorithm known as virtual force directed co-evolutionary PSO (VFPSO) is introduced in [10], however, consideration on minimizing energy consumption is not taken. In [11], a multi-objective problem is considered, where the objectives include maximizing coverage and minimizing energy consumption on sensor communications and sensor movements. However, the search space of PSO algorithm expands exponentially along with the increase of the optimized vector dimensions. Therefore, calculation time of PSO algorithm is still a bottleneck for WSNs optimization.

Based on the above problems, this paper combines PSO algorithm and the ionic bond method, and proposes a sensor node deployment algorithm based on ionic bond-directed particle swarm optimization (IBPSO). On one hand, IBPSO algorithm adopts ionic bond to guide the evolution directions of particles and promote the update speed of PSO algorithm. On the other hand, IBPSO algorithm avoids the defects of the Ion-6 method that fixed the node positions for hexagon topology. IBPSO algorithm has stronger searching ability and faster convergence speed to obtain the optimal deployment compared with PSO algorithm and the Ion-6 Method.

This paper is organized as follows: section 2 gives the problem description and related definitions. Section 3 describes the node deployment algorithm based on ionic bond-directed particle swarm optimization (IBPSO). Section 4 verifies the validity of the algorithm via simulation experiments. At the end of this paper, we come to a conclusion and introduce the future research plan.

2. Description of the problem and related definitions

Assumption 1: Lots of sensor nodes are randomly distributed in a given target area to monitor the interested events, and there exists one sink node as the processing center to implement the IBPSO algorithm.

Assumption 2: Every sensor node has a unique identity.

Assumption 3: Every sensor node has the basic orientation function (perhaps GPS and antenna array) and it can calculate the current position and direction.

Assumption 4: All sensor nodes have the same communication ranges. The coverage area of each sensor node is a circular disk. The sensing range is equal to the

communication range. Every sensor node can communicate with others without losing data.

Assumption 5: Sensor node can accurately finish the position migration and node energy is sufficient to support the node deployment process.

Assumption 6: Sensor node can precisely estimate the distance to the sender by the received signal strength of incoming packets.

Assumption 7: Every sensor node installs a precise antenna array, which can identify the angle of every incoming packet. Each sensor also has a precise compass to determine its moving direction.

Assume that in the target area A, the locations of randomly deployed sensor nodes are all meet the form of uniform distribution model, and any two sensor nodes is not in the same location. The relevant definitions are as follows:

Definition 1. Distance between Node and Target:

Node N_i is in (x_i, y_i) and target N_j is in (x_j, y_j) , then the distance between target N_j and node N_i is defined as $D(N_i, N_j)$, shown as equation (1):

$$D(N_i, N_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

Definition 2. Distance between two Nodes:

Distance d_{AB} of node $A(x_A, y_A)$ to node $B(x_B, y_B)$ is defined as equation (2):

$$d_{AB} = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} \quad (2)$$

3. A node deployment algorithm based on IBPSO optimization

3.1. Particle swarm optimization

Particle swarm is a population based optimization tool inspired by the natural social behavior of certain organisms like bird flocking and fish schooling as developed by Kennedy and Eberhart [12]. This behavior is imitated in PSO where particles fly over the search domain influenced by the experience of their own and the surrounding neighbors. The algorithmic flow in PSO starts with a population of particles whose positions and velocities are randomly initialized in the search space, where the former represents the potential solutions for the current problem, and the latter determines the next movement. The search for optimal position is performed by updating particle velocities (v_{ij}) and positions (x_{ij}) through equation (3) and equation (4) respectively:

$$v_{ij}(t+1) = w \times v_{ij}(t) + c_1 \times r_{1j}(t) \times (p_{ij}(t) - x_{ij}(t)) + c_2 \times r_{2j}(t) \times (p_{gj}(t) - x_{ij}(t)) \quad (3)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t) \quad (4)$$

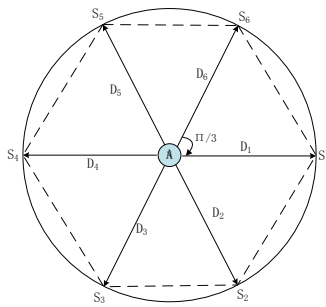


Figure 1 The six ionic bonds and stable slots of sensor node A

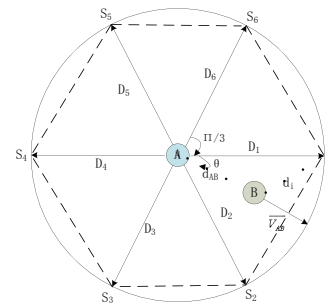


Figure 2 The path and direction of sensor node B's movement

where w is inertia weight used to control the effect of the previous velocity on the current velocity. Decreasing inertia weight over time encourages higher exploration at the beginning and better tuning at the end of one search. c_1 and c_2 are the learning factors to control the effect of the "best" factors of particles. $r_{1j}(t)$ and $r_{2j}(t)$ are two independent random numbers in the range of $[0.0, 1.0]$. The velocity of the particle is influenced directly by two factors: the best position found so far by the particle ($p_{ij}(t)$ i.e. *pbest*) and the best position found by the neighboring particles ($p_{gj}(t)$ i.e. *gbest*). The quality of the solution is evaluated by a fitness function, which is a problem-dependent function. If the current solution is better than the fitness of $p_{ij}(t)$ or $p_{gj}(t)$, the best value will be replaced by current solution accordingly. This update process will continue until stopping criterion is met, usually when either maximum iteration is achieved or target solution is attained.

3.2. Ionic bond based method

Before discussing the ionic bond based method, we made an assumption that sensor nodes are modeled as ions, and the links between them are treated as ionic bonds which can be seen as a force between every two nodes. The number of ionic bonds of a sensor node is limited, in order to organize the deploying topology as the hexagonal shape, the number of the ionic bonds of every sensor node are set to six. When the number comes to six, the sensor node will expel others out of its field. Sensor nodes organize themselves as the hexagonal shape to maximize the network's coverage area, retain the network connectivity and prevent from introducing the coverage holes. As shown in Fig.1, assume that node A is the first sensor node to start deploying, then node A will determine the direction of each ionic bond. The six ionic bond directions are just divided the coverage area of A into six slots which would form the hexagon. All the nodes during the deployment will select their six directions according to that of the first node A.

Now we define some variables, S_i represents the stable neighbor of A, I_i represents the stable ionic bond

between A and S_i , then \vec{D}_i represents the direction of I_i , the distance between A and S_i is equal to the sensing radius R . So according to the defined variables, the stable neighbors of A are S_1, S_2, \dots, S_6 and the directions of the six ionic bonds of sensor node A are $\vec{D}_1, \vec{D}_2, \dots, \vec{D}_6$.

At the beginning, all sensor nodes have free ionic bonds and are waiting for combining with other nodes whose states are unsteady. We randomly choose a sensor node A to enter the active mode and start the deploying procedure. Node A sets the default directions of the six ionic bonds and broadcasts its six directions which are $\vec{D}_1, \vec{D}_2, \dots, \vec{D}_6$ to all neighbors S_1, S_2, \dots, S_6 .

Seen in Fig.2, assume B is an unsteady sensor node that can directly receive the bond packet from A, and its distance to A is d_{AB} which can be calculated through equation (2). Define \vec{V}_{AB} is the incoming direction of the bond packet. For each free ionic bond I_i in the bond packet, we assume the distance from node B to S_i as d_i and the direction as \vec{V}_i , B will compute d_i and \vec{V}_i to the corresponding S_i . Then, sensor node B sends the results to A. We can use trigonometric function as the equation (5) to compute d_i :

$$d_i = \sqrt{d_{AB}^2 + R^2 - 2d_{AB} \times R \times \cos \theta_i} \quad (5)$$

where R is the sensing radius, \vec{D}_i is the direction from sensor A to one of A's six ionic bonds of sensor node S_i . θ_i is the included angle of \vec{V}_{AB} and \vec{D}_i . It can be obtained from the inner product of \vec{V}_{AB} and \vec{D}_i shown in equation (6).

$$\theta_i = \cos^{-1} \left(\frac{|\vec{V}_{AB} \cdot \vec{D}_i|}{|\vec{V}_{AB}| |\vec{D}_i|} \right) \quad (6)$$

The moving direction \vec{V}_i can be computed from equation (7) as follows:

$$\vec{V}_i = R \times \vec{D}_i - d_{AB} \times \vec{V}_{AB} \quad (7)$$

Sensor node A instructs the sensor node to move to each S_i with the minimal d_i after it collects the results from all neighbor sensor nodes S_1, S_2, \dots, S_6 . These instructed sensor nodes will switch to active mode. If $S_1,$

S_2, \dots, S_6 are not occupied by any other node, they are ready for moving to the stable slots and park at the corresponding locations. After these six nodes complete their respective movement procedure, they will notify sensor node A. Then A will expel all passive mode sensors out its sensing field. After that, node A transforms to lock state. Those lock state sensors will no longer move. The six neighbor sensor nodes S_1, S_2, \dots, S_6 will repeat A's work and instruct their own neighbor nodes to achieve lock states. It is worth noting that one node can only execute this process at one time. Finally, all the nodes are steady and the whole procedure is completed.

3.3. Proposed optimization algorithm for node deployment

In this section, we propose a sensor node deployment algorithm based on ionic bond-directed particle swarm optimization (IBPSO) by combining PSO algorithm and the ionic bond method. During optimization, each particle changes its velocity toward $pbest$ and $gbest$ position with the bounded random acceleration. Velocity and position of particle are updated according to equation (3) and (4) in section 3.1. $pbest$ and $gbest$ are updated according to equation (8) and (9) respectively:

$$pbest = \begin{cases} pbest & f(pnow) \geq f(pbest) \\ pnow & f(pnow) < f(pbest) \end{cases} \quad (8)$$

$$pbest = \min\{pbest_1, pbest_2, \dots, pbest_n\} \quad (9)$$

where $pbest$ is the best location of a particle, $gbest$ is the global optimal solution and $pnow$ is the current location. In original PSO, the initialized positions and velocities of particles are generated by a random condition, so the convergence speed is partially determined by the initialized parameters of particles. Moreover, the $pbest$ and $gbest$ positions may not be the optimal results, especially in the forefront of optimization, which will impact the convergence of optimization. Hence, if some other appropriate factors can be introduced to direct the particles flying to the optimal positions, the convergence speed and searching ability of PSO can be improved. It is also the key motivation for combining with the ionic bond method.

We can abstract the issue of sensor node deployment in wireless sensor networks to a problem of effective network coverage of target area optimization where input parameters are integer vectors of the nodes moving positions. Assume that the wireless sensor network is made up by N sensor nodes, the velocity of each particle is updated according to not only the historical optimal solutions but also the ionic bonds of sensor nodes. Updating is expressed by equation (10) and (11).

$$v_{ij}(t+1) = w(t) \times v_{ij}(t)$$

$$\begin{aligned} &+c_1 \times r_{1j}(t) \times (p_{ij}(t) - x_{ij}(t)) \\ &+c_2 \times r_{2j}(t) \times (p_{gj}(t) - x_{ij}(t)) \\ &+c_3 \times r_{3j}(t) \times g_{ij}(t) \end{aligned} \quad (10)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t) \quad (11)$$

where the meaning of $c_1, c_2, p_{ij}(t), p_{gj}(t), r_{1j}(t)$ and $r_{2j}(t)$ are the same as those in equation (3), c_3 is an acceleration constant, $r_{3j}(t)$ is also a random function in the range [0,1] which is independent to $r_{1j}(t)$ and $r_{2j}(t)$. $w(t)$ starts with a value 0.9 and linearly decreases to 0.4 [13] in terms of equation (12). $g_{ij}(t)$ is the proleptic motion suggested by ionic bond method of the i^{th} particle in the j^{th} dimension, which is computed by equation (13).

$$w(t) = 0.9 - \frac{t}{MaxIterations} \times 0.5 \quad (12)$$

where $MaxIterations$ is the number of maximum iterations.

$$g_{ij}(t) = d_i \times \frac{|\vec{V}_i \bullet \vec{J}|}{|\vec{V}_i|} \quad (13)$$

where \vec{J} is a unit vector in the j^{th} dimension. According to equation (5), (6) and (7), we obtain equation (14) as follows:

$$\begin{aligned} g_{ij}(t) = &\sqrt{d_{AB}^2 + R^2 - 2 \times d_{AB} \times R \times \frac{|\vec{V}_{AB} \bullet \vec{D}_i|}{|\vec{V}_{AB}| |\vec{D}_i|}} \\ &\times \frac{|(R \times \vec{D}_i - d_{AB} \times \vec{V}_{AB}) \bullet \vec{J}|}{|R \times \vec{D}_i - d_{AB} \times \vec{V}_{AB}|} \end{aligned} \quad (14)$$

where $g_{ij}(t)$ is the proleptic motion suggested by ionic bond method of the i^{th} particle in the j^{th} dimension.

With the guidance of ionic bond, the IBPSO algorithm can evolve to global optimization purposefully.

The detailed procedure of IBPSO algorithm is described as follows:

1. Initialize a population of particles with random positions, velocities and granularities. Obtain the effective detection area formed by stationary nodes.
2. Evaluate the effective coverage performance. Compare and update the optimal $pbest$ value of each particle and the global optimal $gbest$ of the whole population.
3. Change velocity and position of a particle according to equation (10) and (11) respectively.
4. Halve the granularity when $gbest$ is not evolved in recent 30 iterations, renew the velocities randomly, and re-analyze the fitness.
5. Loop to step 1 until a criterion is met, usually represented by a sufficiently small granularity, a sufficiently good fitness or a maximum number of iterations ($MaxIterations$).

// The process of IBPSO algorithm

Initialize particles population with random positions, velocities and granularities;

$T_{max} = MaxIterations$;

```

t = 1;
while (t ≤ Tmax or ideal fitness is not attained)
do {
    calculate fitness value of each particle using fitness
    function;
    update pij(t) if the current fitness value is better than
    pij(t - 1);
    determine pgj(t): choose the particle position with the
    best fitness value of all the neighbors as the pgj(t);
    for each particle {
        calculate particle velocity according to equation
        (10);
        update particle position according to equation (11);
    }
    t++;
}
    
```

However, as the assumptions mentioned in section 2, compared IBPSO algorithm with PSO algorithm and the Ion-6 Method, a stronger search ability and a faster convergence speed to obtain the optimal deployment require that all the sensor nodes have accurate orientation abilities and spend more energy consumption. We sacrifice some hardware conditions to achieve a higher efficiency and a faster speed. Fortunately, with the development of microelectronics technology, the cost of the sensor nodes and orientation devices will lower and lower, so the algorithm we proposed is feasible.

4. Simulation and analysis

4.1. Performance of the IBPSO algorithm

We use Visual Studio 2010 to develop a simulation software which is appropriate for the deployment of wireless sensor network in order to verify the effectiveness of the algorithm of IBPSO.

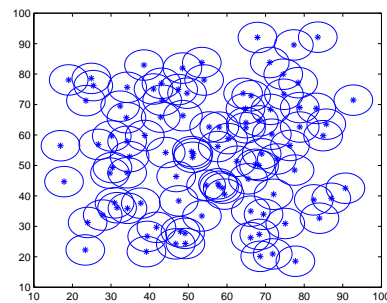
Values of specific simulation parameters are shown in Table 1. We assume that local optimum value c_1 , global optimal value c_2 and the value c_3 of ionic bond oriented to particles have the same influence during the particles evolution processso we set all the three learning factors $c_1=c_2=c_3=1$.

In this section, simulation experiments are carried out to investigate the performance of IBPSO. Sensor nodes are considered to be randomly deployed in a square region with area of $100 \times 100m^2$, $300 \times 300m^2$ and $400 \times 400m^2$ respectively. The detailed parameters values are shown in Table.1.

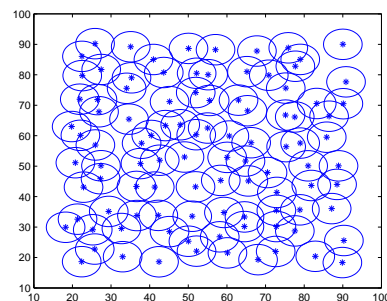
According to Fig.3, Fig.4 and Fig.5, the results with 100 times average in experiments show that the excellent performance on coverage carried out by IBPSO, the distribution of sensor nodes determined by IBPSO is symmetrical and effective, and the effective coverage determined by IBPSO are 96.12% ($T = 100 \times 100m^2, N = 100$), 97.49% ($T = 300 \times 300m^2, N = 250$) and 98.76% ($T = 400 \times 400m^2, N = 400$) respectively.

Table 1 Experiment parameters

Parameter Name	Parameter Values
Target region T	$100 \times 100m^2$
	$300 \times 300m^2$
	$400 \times 400m^2$
Distribution mode	random distribution
Number of nodes N	100, 250, 400
Communication radius of node R	5m
Learning factor c_1	1
Learning factor c_2	1
Learning factor c_3	1
$MaxIterations$	300



(a) Random placement in 100×100 area

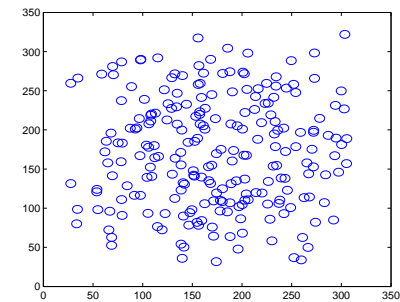


(b) Deployment after execution of IBPSO in 100×100 area

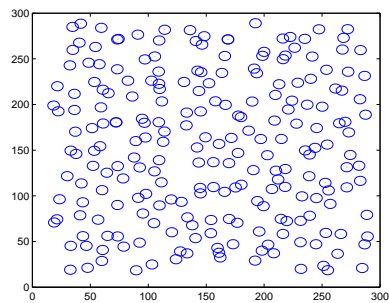
Figure 3 Coverage comparison between IBPSO and Random deployment in 100×100 area

4.2. Comparative analysis of algorithms

In this section, a series of simulation experiments are executed to illustrate the effect on the performance of IBPSO algorithm by comparing with three existing algorithms. Article [12] proposed an algorithm based on particle swarm optimization called PSO algorithm, article [8] put forwards a position-less self-deploying method for wireless sensor networks based on the ion-6 method, and article [14] uses unattended random node deployment and

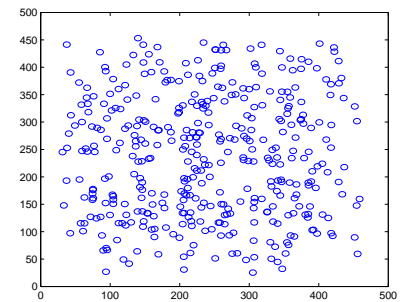


(a) Random placement in 300×300 area

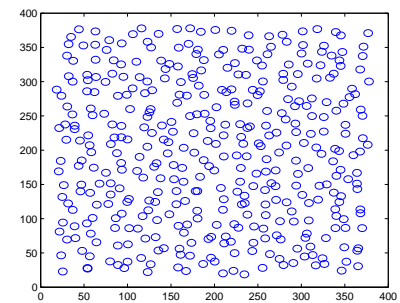


(b) Deployment after execution of IBPSO in 300×300 area

Figure 4 Coverage comparison between IBPSO and Random deployment in 300×300 area



(a) Random placement in 400×400 area



(b) Deployment after execution of IBPSO in 400×400 area

Figure 5 Coverage comparison between IBPSO and Random deployment in 400×400 area

partial coverage in wireless sensor networks for long-lasting surveillance of areas of interest. We compare these three algorithms with IBPSO algorithm proposed in this paper and analyze their respective performances from the following three aspects: effective moving ratio (EMR), deploying time and coverage rate of the given area.

Fig.6 displays the effective moving ratio. In the PSO process, sensor node always moves a small-step to adjust the moving direction. The average EMR of a sensor node ranges from 1.15 to 1.36. By computing and selecting the suitable candidates, each node can achieve to the ideal position by almost one step with the Ion-6 method, which introduces nearly zero redundant moving distance when network scale is 19. When the network scale is 127, the EMR of Ion-6 method is only 1.2. The EMR of Random deployment is ranged from 1.1 to 1.28 while the EMR of IBPSO is ranged from 0.8 to 1.17. It implies that IBPSO causes less redundant moving compared of the former three algorithms.

Fig.7 shows the time to complete the deployment. Sensors in the PSO process use small and uncertain moving steps to adjust their positions. When the number of sensor nodes grows, time to complete the deployment increases rapidly. On the contrary, sensor nodes in the

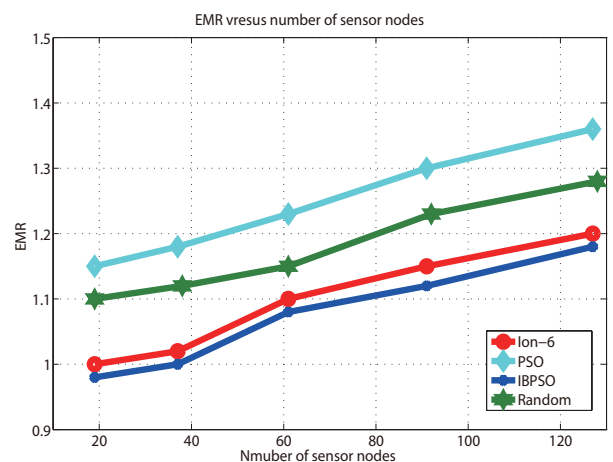


Figure 6 The effect moving ratio (EMR)

Ino-6 method use large moving steps to adjust their positions when sensors are crowded. The adjusting step gradually shrinks when sensors spread out. IBPSO improves the two algorithms and the deploying time is the

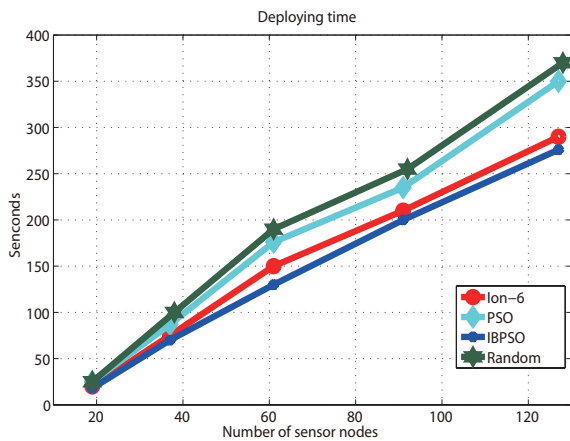


Figure 7 Deploying time

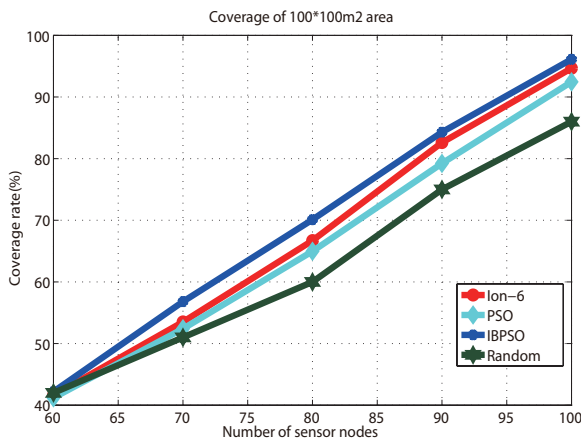


Figure 8 Coverage of the given area

shortest in the four algorithms which is confirmed by curve changes in Fig.7.

Then we do lots of experiments in the $100 \times 100m^2$ area to test the coverage rate of the given area when the number of sensor nodes is 60, 70, 80, 90 and 100 separately. Curves in Fig.8 show that along with the increase of sensor nodes, coverage rate of the four algorithms all become larger and IBPSO algorithm we proposed can lead to the largest coverage rate in the sensor node network.

For detailing the performance of the proposed IBPSO, we compared IBPSO with PSO in the aspect of the improvement in coverage. As shown in Fig. 9, obviously, the IBPSO can converge more rapidly, where it can achieve global optimal searching with only 110 iterations. The PSO can only complete the global searching after 250 iterations. This performance also confirmed the advantages of IBPSO.

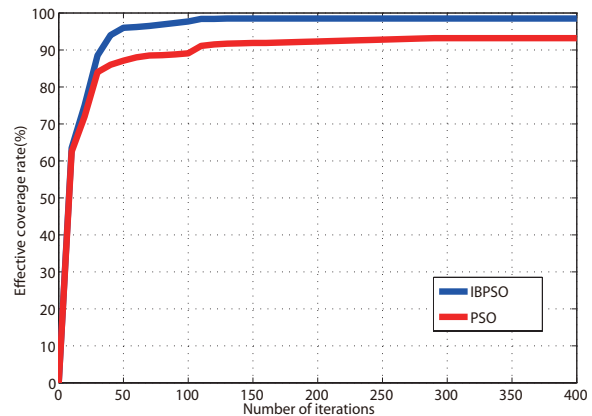


Figure 9 The improvement in coverage during the execution of the IBPSO and PSO

4.3. Discussion

So far, we have analyzed the performance of the proposed node deployment algorithm and got a satisfied conclusion that the IBPSO algorithm has the advantage in three aspects: the effect moving ratio, deploying time and coverage of the given area. In this section, we address one practical issue to discuss several important parameters of the IBPSO algorithm.

Before discussion, we assume that the sensor node itself has the basic orientation function and it can calculate the current position.

We remark that IBPSO integrated the advantages of particle swarm optimization and ionic bond based method, which can be discussed in equation (15):

$$\begin{cases} v_{ij}(t+1) = w \times v_{ij}(t) \\ \quad + c_1 \times r_{1j}(t) \times (p_{ij}(t) - x_{ij}(t)) \\ \quad + c_2 \times r_{2j}(t) \times (p_{gj}(t) - x_{ij}(t)) \\ \vec{V}_i = R \times \vec{D}_i - d_{AB} \times \vec{V}_{AB} \\ v_{ij}(t+1) = w(t) \times v_{ij}(t) \\ \quad + c_1 \times r_{1j}(t) \times (p_{ij}(t) - x_{ij}(t)) \\ \quad + c_2 \times r_{2j}(t) \times (p_{gj}(t) - x_{ij}(t)) \\ \quad + c_3 \times r_{3j}(t) \times g_{ij}(t) \end{cases} \quad (15)$$

where w is inertia weight used to control the effect of the previous velocity on the current velocity. c_1 and c_2 are the learning factors to control the effect of the "best" factors of particles; the definitions of other parameters can be seen in equation (3), (7) and (10).

We pay attention to parameters c_1 , c_2 and c_3 (where $c_1+c_2+c_3=1$), c_1 , c_2 reflect the effect of the PSO algorithm and c_3 reflects the proleptic motion suggested by ionic bond method of the i^{th} particle in the j^{th} dimension. We consider the values of $v_{ij}(t+1)$ in two situations where c_1 , c_2 and c_3 take different values. The given issue becomes a linear algebra problem.

Situation 1. $c_1 + c_2 = 1, c_3 = 0$

In this situation, $c_3 = 0$ means that there is no influence of the proleptic motion suggested by ionic bond method of the i^{th} particle in the j^{th} dimension, then $g_{ij}(t) = 0$, IBPSO algorithm degenerates into ordinary PSO algorithm.

Situation 2. $c_1 + c_2 = 0, c_3 = 1$

Similarly, in this situation, $c_1 + c_2 = 0$ means that there is no effect of the motion suggested by PSO algorithm. Therefore IBPSO algorithm will degenerate into ordinary ionic bond method, where the movements of nodes rely on the moving direction \vec{V}_i used in ionic bond method.

5. Conclusion

In this paper, the ionic bond-directed particle swarm optimization has been proposed as a practical approach for sensor node deployment in wireless sensor networks. The proposed IBPSO algorithm uses PSO to search the optimal deployment strategy and determines the velocities of particles in PSO by the tradeoff between optimal solutions and ionic bond between sensor nodes. Compared to Ion-6, IBPSO algorithm avoids the defects of Ion-6 that fixed the node positions in order to form the hexagon shape. Furthermore, IBPSO uses ionic bond method to direct the movements of particles, so the global searching and regional convergence abilities are better than PSO. The simulation results demonstrate that IBPSO can implement sensor node deployment much more efficiently than Ion-6 and PSO, since it reduces the computation time more than 15% with Ion-6 and 40% with PSO respectively, and it also performs better on improving the effective coverage area of WSNs, which verify that IBPSO is competent for sensor node deployment in wireless sensor networks. However the IBPSO algorithm we proposed need some specific assumptions such as the basic orientation function etc. Therefore, our future work is to minimize these assumptions and apply the proposed algorithm to the actual scene.

Acknowledgement

The subject is sponsored by the National Natural Science Foundation of P. R. China (No.61373138, 61171053, 61003039), Scientific & Technological Support Project (Industry) of Jiangsu Province (No.BE2012183), Natural Science Key Fund for Colleges and Universities in Jiangsu Province (No.12KJA520002), Postdoctoral Foundation (No.2013T60536, 2012M511753, 1101011B), Science & Technology Innovation Fund for higher education institutions of Jiangsu Province (No.CXLX13.467, CXZZ11.0409), Foundation of Nanjing University of Posts and Telecommunications (No.NY212047), Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (No.yx002001).

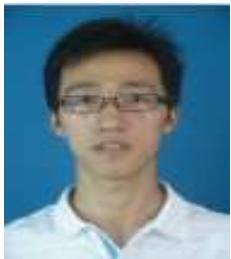
References

- [1] S. C Huang, APPLIED MATHEMATICS & INFORMATION SCIENCES, **6**, 331-337 (2012).
- [2] Stefano Bistarelli, Ugo Montanari, Francesca Rossi, Semiring-based constraint satisfaction and optimization, Journal of the ACM, **44**, 201-236 (1997).
- [3] A. M Cheng, A. V Savkin, CAMBRIDGE UNIV PRESS, **30**, 661-669 (2012).
- [4] C. F Cheng, K. T Tsai, IEEE SENSORS JOURNAL, **12**, 1726-1735 (2012).
- [5] T. Li, C. C Yu, M. H Yang, Proceedings of 6th International Conference on Wireless Communications, Networking and Mobile Computing (WICOM), **2010**, 1-4 (2010).
- [6] Y. F Niu, L. Peng, W. Zhang, Proceedings of 22nd Chinese Control and Decision Conference, **2005**, 4134-4138 (2010).
- [7] J. Zhao, J. C Zeng, IEEE SENSORS JOURNAL, **10**, 1328-1334 (2010).
- [8] S. C Huang, International Journal of Distributed Sensor Networks, **2012**, 1-10 (2012).
- [9] X. Wu, L. Shu, J. Yang, H. Xu, J. Cho, S. Lee, Swarm Based Sensor Deployment Optimization in Ad hoc Sensor Networks, Texts in Second International Conference on Embedded Software and Systems, **3820**, 34-39 (2005).
- [10] X. Wang, S. Wang, J. J Ma, Sensors, **7**, 354-370 (2007).
- [11] X. Wang, J. J Ma, S. Wang, D. W Bi, Sensors, **7**, 628-648 (2007).
- [12] J. Kennedy, R. C Eberhart, Proceedings of IEEE International Conference on Neural Networks, **4**, 1942-1948 (1995).
- [13] V. C Mariani, A. R. K Duck, F. A Guerra, L. D Coelho, R. V Rao, APPLIED THERMAL ENGINEERING, **42**, 119-128 (2012).
- [14] P. Medagliani, J. Leguay, G. Ferrari, R. M Lopez, PERVASIVE AND MOBILE COMPUTING, **8**, 429-447 (2012).



Haiping Huang received both his B.E degree and M.S degree of computer software & theory from Nanjing University of Posts and Telecommunications in 2002 and 2005, respectively in Nanjing city of China, and Ph. D degree of computer application technology from

Suzhou University in 2009, in Suzhou city of China. His research addresses wireless sensor networks, information security, mobile agent, and Internet of Things. He is now vice-professor & master tutor in College of Computer Science and Technology, Nanjing University of Posts and Telecommunications (a.b. NUPT) and postdoctoral candidate in College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. Dr. Huang serves as vice secretary general of Information Security Special Interest Committee, Jiangsu Institute of Electronics, vice secretary general of Jiangsu committee, National Computer Continuing Education Research Association and the member of ACM & CCF.



Junqing Zhang received his Bachelor's degree of computer science and technology from Nanjing University of Posts and Telecommunications, in Nanjing city of China, and he is now a postgraduate of computer software and technology from Nanjing

University of Posts and Telecommunications, in Nanjing city of China . His research addresses coverage and topology in wireless sensor networks.



Ruchuan Wang was born in Hefei, Anhui Province, China, on August 21, 1943. He received his B.S degree of computational mathematics from The PLA Information Engineering University in Zhengzhou city of China in 1968. His research interests include intelligent agent,

information security, wireless networking and distributed computing. He was with Bremen University, Germany, Munich University, Germany, and Max-Planck Institute during 1984-1992. And now he is a professor and a Ph.D supervisor in Computer Science at Nanjing University of Posts and Telecommunications, China.



Yisheng Qian is now an undergraduate student of electrical communication engineering from Nanjing University of Posts and Telecommunications (Nanjing, Jiangsu, China) and New York Institute of Technology (New York, U.S.A) respectively.