Deployment algorithm based on dynamic multi-populations particle swarm optimization for wireless sensor networks

Lei Hong*

School of Information Technology, Jinling Institute of Technology, Nanjing 211169, China

Received 1 March 2014, www.cmnt.lv

Abstract

Aiming at improving coverage rate and reducing coverage holes of wireless sensor networks, this paper proposes a deployment algorithm based on dynamic multi-populations particle swarm optimization. K-Means clustering algorithm is employed to divide the network into several sub-populations dynamically, which could weaken particles on the pursuit of local optima, realize the improvement of basic PSO (Particle Swarm Optimization) algorithm, and solve the "premature" problem of basic PSO algorithm effectively. In addition, it also accelerates the convergence of the algorithm. Simulation results show that this deployment algorithm can improve the network coverage rate effectively. Comparing with the conventional particle swarm optimization algorithm, its coverage rate is increased by 3.66%.

Keywords: deployment, particle swarm optimization, *k*-means, wireless sensor networks

1 Introduction

Wireless sensor network is constituted in ways of self-organization and multi-hop by large volumes of sensor nodes with communication and computation capability. Nodes in the network are able to collaboratively perceive, collect, process, and transmit the information of perceived objects within the coverage area of the network, as well as to report the information to users [1,2]. Wireless sensor network has great potential application value in military, transportation, medical care, and environment monitoring [3,4]. Node deployment is a key issue in wireless sensor network, which reflects the quality of awareness service provided by the network.

Currently, there are mainly two ways to deploy nodes for wireless sensor network: deterministic deployment and random deployment [5]. Deterministic deployment refers to the approach that, when the status of wireless sensor network is relatively fixed, or the size of deployment area is determined, network topology structure may be determined according to pre-set node position, or sensor node density in key areas may be increased. However, in practical natural environment, the environment of monitored area may be quite severe, or the network status is not determined in advance. As for this, random deployment may be adopted. However, this deployment method may easily lead to dead zones. Under certain deployment density, it may be hard to reach the required coverage rate. Hereby, how to reduce dead zones and to improve network coverage rate is an important problem urgently needing to be solved by wireless sensor network.

Based on parallel optimization and fast convergence of fish swarm algorithm, Literature [6] proposed WSNs coverage optimization strategy based on fish swarm algorithm. Literature [7] put forward a mobile sensor deployment algorithm based on virtual force. The algorithm integrates the concepts of field potential and disc packet. Yet, VFA didn't solve the problem of "dead zone". Literature [8] proposed a sensor node deployment algorithm based on virtual rhombic grid. The algorithm integrates deterministic deployment and self-organizing deployment into a unified platform. Particle Swarm Optimization (PSO) [9] is an intelligent optimization algorithm firstly proposed by Professor Kennedy and Eberhard from the US. Its ideology derived from artificial life and evolutionary computation theory, which was mainly inspired by birds flock's behaviour of looking for food. Literature [10] and Literature [11] presented a wireless sensor network deployment optimization method based on particle sward optimization algorithm. Although particle sward optimization algorithm is proved with the ability to optimize wireless sensor network deployment, in spatial searching, standard particle sward optimization algorithm is easy to take on "precocity" phenomenon, which limits the searching range of particle. In allusion to the problem of regional optimization that may be resulted by precocity of PSO algorithm, Literature [12] introduced the concept of disturbance factor into the basic particle sward optimization algorithm, and also applied disturbance factor in mixed wireless sensor networks. In Literature [13], Lin, et al. put forward multi-populations PSO algorithm, which was able to improve the optimazation performance of particles. It's running efficiency and precision was both superior to mono-population PSO algorithm. However, the literature divided sub-population by randomly selecting particles in populations. Such sub-population division strategy was blind to some extent.

* *Corresponding author's* e-mail: honglei@jit.edu.cn

With respect to the above issues, the paper puts forward a wireless sensor network node deployment optimization strategy based on dynamic multi-populations particle sward optimization (DMPSO) algorithm. In the optimization process, *k*-means clustering algorithm is introduced to divide the population into several sub-populations, so that optimization may be performed independently on these sub-populations. On the other hand, in order to enhance the information exchange between sub-populations, sub-populations are dynamically re-grouped, so as to reduce parties' pursuit for regional optimal point, and to effectively avoid particle "precocity" in basic PSO algorithm. This improves the coverage rate of network.

The rest part of this paper is organized as follows: in Section 2, we propose the deployment model for wireless sensor networks. In Section 3, we propose the DMPSO algorithm. In Section 4, we simulate the DMPSO optimazation scheme by using computer software and evaluate its performance. Finally, in Section 5, we reach the main conclusions.

2 Deployment model

2.1 PARTICLE SWARM OPTIMIZATION MODEL

Standard particle sward optimization algorithm [14] takes individuals as particles without weight and volume in *N*-dimensional space, which flies at a certain velocity in the searching space. The flying velocity is dynamically adjusted according to individual and population flying experience.

Assuming that $X_i=(X_{i1}, X_{i2}, \dots, X_{iN})$ is present position of Particle *i*; $V_i=(V_{i1}, V_{i2}, \dots, V_{iN})$ is present flying velocity of Particle *i*; $P_i=(P_{i1}, P_{i2}, \dots, P_{iN})$ is the best position experienced by Particle *i*, i.e. the position experienced by Particle *i* with the best adaptive value, which is denoted as P_{best}, also referred to as individual best position. Assuming $f(x)$ as minimized fitness function, so that the best position of Particle *i* may be determined by the below equation:

$$P_i(t+1)=\begin{cases} P_i(t) & f(X_i(t+1)) \geq f(P_i(t)) \\ X_i(t+1) & f(X_i(t+1)) < f(P_i(t)) \end{cases}. \quad (1)$$

Assuming the number of particles in the population is *s*, and the best position experienced by all particles in the population as P_g, also referred to as global best position g_{best}, so that:

$$\begin{aligned} &P_g(t) \in \{P_0(t), P_1(t), \dots, P_s(t)\} \mid f(P_g(t)) = \\ &\min\{f(P_0(t)), f(P_1(t)), \dots, f(P_s(t))\} \end{aligned}. \quad (2)$$

For the *t*-th iteration, motion of Particle *i* in *d*-dimensional space ($1 \leq d \leq D$) follows the below Equation:

$$\begin{aligned} v_{id}(t+1) = \omega \cdot v_{id}(t) + c_1 \cdot rand() \cdot (P_i - x_{id}(t)) \\ c_2 \cdot rand() \cdot (P_g - x_{id}(t)) \end{aligned}, \quad (3)$$

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

where, ω is inertia coefficient, which endows particles with the tendency to expand the searching space, so as to search new areas; c_1 and c_2 are acceleration constant; rand() refers to the random values within [0,1].

2.2 OPTIMIZATION MODEL FOR DMPSO

Particle swarm optimization algorithm based on dynamic multi-populations (DMPSO) is an improved multi-populations PSO algorithm. Basic ideology of DMPSO: In the algorithm, population is divided into several sub-populations by *k*-means clustering algorithm, so that sub-populations may be optimized independently. Meanwhile, after several generations of iteration, sub-populations are then re-divided to form new sub-populations, so as to enhance information exchange between sub-populations.

Assuming that there are m particles in Population *M*, the m particles are then divided into *K* sub-populations, which may be described as $M=(M_1, M_2, \dots, M_k)$. Each sub-population may also be represented as $M_i=(X_1, X_2 \dots X_p)$, ($i=1, 2 \dots K$; $0 \leq p \leq m$), and elements in sub-populations stand for individual particles, and the total number of particles in all sub-populations is *m*. For each Particle *i* in sub-populations, X_i is employed to indicate its position $X_i=(X_{i1}^1, X_{i1}^2, X_{i2}^1, X_{i2}^2, \dots, X_{iN}^1, X_{iN}^2)$, where elements in it separately refer to horizontal coordinate and vertical coordinate of sensor nodes in the particle. The flying velocity of Particle *i* is described as $V_i=(V_{i1}^1, V_{i1}^2, V_{i2}^1, V_{i2}^2, \dots, V_{iN}^1, V_{iN}^2)$, where elements in it the component velocity of sensor nodes in the particle along X axis and Y axis. Clustering center is taken as the best position passed by all particles in each sub-population, represented by P_{lg}, also referred to as Lg_{best}. P_{lg} is used to replace population global best position P_g in Equation (1). After the improvement, velocity evolution formula of Particle *i* in *d*-dimensional space ($1 \leq d \leq D$) is shown below:

$$\begin{aligned} v_{id}(t+1) = \omega \cdot v_{id}(t) + c_1 \cdot rand() \cdot (P_i - x_{id}(t)) \\ c_2 \cdot rand() \cdot (P_{lg} - x_{id}(t)) \end{aligned}. \quad (5)$$

In each sub-population, particles often perform searching with clustering center as the best position of the sub-population. If clustering center of the sub-population is coincidentally located at the position of regional optimal solution, the sub-population may take on "precocity" convergence. Here, after *R* generations' iteration, all particles in population will be considered as a whole, and will be divided into new sub-generations by *k*-means clustering algorithm. Dynamic re-combination of sub-population is designed to improve information exchange between sub-populations, so as to avoid "precocity" of particles.

Moreover, *k*-means clustering algorithm is adopted to partition the data of all node coordinates into *K* areas, i.e. the number of sub-populations. The number of sub-populations determines social information sharing degree between particles in each sub-population. If parties are lack of social information sharing, network coverage may be significantly reduced.

In order to enhance global searching performance of particles in searching process, inertia coefficient factor in velocity evolution Equation (5) will be appropriately adjusted:

$$\omega(t) = 0.9 - \frac{t}{\max Iterations} \cdot 0.5, \quad (6)$$

where, t refers to present iteration generation of particle, and maxIterations stands for the maximum iteration generation in the algorithm. It may be seen from Equation (6) that ω reduces linearly with the iteration generation increases. As for this, the algorithm is endowed with strong global searching ability in the beginning and strong regionnal searching ability in later phase.

3 DMPSO deployment algorithms

In the paper, random deployment method is employed. In the beginning when nodes are scattered into the monitoring area, dead zones may be easily caused in the network. As for this, secondary deployment shall be performed in allusion to such "problematic" area. In order to simplify the network model, assuming that Monitoring Area A is a two-dimensional plane, while N sensor nodes with the same parameters are scattered in the area. The perception radius of each node is denoted as R_s, and communication radius as R_C. In order to keep the connectivity of network, communication radius is set no less than two times of the perception radius [15], i.e. $R_c \geq 2R_s$. Sensor node set is denoted as $S=\{S_1, S_2, \ldots, S_N\}$, where, $S_i = \{x_i, y_i, R_s\}$, $i \in (1, 2, \ldots, N)$. The coverage model of each node may be considered as a circle with coordinates of the node as the center, and R_s as the radius.

Digitally discretizing Monitoring Area A as m × n pixels, with coordinates of pixels as (x,y), and the distance between sensor node S_i and a certain pixel p shall be:

$$d(S_i, p) = \sqrt{(x_i - x)^2 + (y_i - y)^2}. \quad (7)$$

Here, node Boolean coverage model, i.e. 0-1 coverage model, is employed. Assuming the event of a certain pixel's being covered by sensor nodes in the monitoring area as t_i, if the event is true, $p(t_i)=1$; or else $p(t_i)=0$. Being described in Equation:

$$p(t_i) = \begin{cases} 1, & d(S_i, p) \leq R_s \\ 0, & d(S_i, p) > R_s \end{cases}. \quad (8)$$

Monitoring Area A is a m × n rectangle, which is divided into m × n pixels of equal size, with area as 1. The discrete precision is 1. For pixel (x,y), as long as there is 1 node in Node Set S covers the pixel, the pixel shall be considered being covered by Node Set S. Otherwise, the pixel (x,y) shall be considered uncovered. Denoting the rate of pixel (x,y)'s being covered by Node Set S as p(x,y,S), so that:

$$p(x, y, S) = p\left(\bigcup_{i=1}^{N} t_i\right) = 1 - \prod_{i=1}^{N} (1 - p(t_i)). \quad (9)$$

As for this, the total area covered by Sensor Node Set S is just the union set of all pixels covered by all nodes in the node set, denoted as S_{area}, then:

$$S_{area} = \int_0^m \int_0^n p(x, y, S)\, dxdy. \quad (10)$$

The optimization goal of wireless sensor network node deployment is to maximum the coverage rate of the network. Here, coverage rate refers to the specific value between coverage area of Node Set S and the total area of the monitoring area, i.e.:

$$\sigma = \frac{S_{area}}{m \times n}. \quad (11)$$

Denoting the fitness function as follows:

$$f(X) = \max \sigma = \max\left(\frac{S_{area}}{m \times n}\right). \quad (12)$$

When f(X) obtains it maximum value, node position information is the best deployment position of node in wireless sensor network.

Each particle in monitoring area represents one sensor node deployment method. When Equation (9) is taken as the fitness function, node deployment optimization algorithm based on DMPSO is shown as Algorithm 1:

ALGORITHM 1: DMPSO algorithm

Algorithm: DMPSO

a) Initializing m particles, i.e. randomly generating Position X_i and Velocity V_i of each particle;
b) Dividing the population into K sub-populations with k-means clustering algorithm;
c) Updating the velocity and position of each particle in each sub-population with Formula (2)~(4);
d) Calculating the coverage rate of each particle according to fitness function;
e) Comparing the coverage rate of particle with its best position P_{best}; if the result is better, re-setting P_{best};
f) Comparing the coverage rate of each particle in each sub-population with the best position of the sub-population Lg_{best}; if the result is better, re-setting Lg_{best};
g) Judging if R iteration generations have been reach; if so, returning to b); or else, executing h);
h) If the pre-set maximum iteration generation or satisfactory coverage rate has been reached, stopping the process; optimal individual position X_i of the population shall be taken as the result; or else, returning to c) and repeating the steps.

4 Experiments and result analysis

4.1 COVERAGE AND UNIFORMITY

Generally, coverage can be considered as the measure of quality of service of a sensor network. Gage invented the concept of coverage in the research of multi-robot systems [16]. We define it as the ratio between sum of the coverage area of all the nodes and the area of the entire target region, shown in Equation (13). The definition of sum of the coverage area is taken from the concept of union in the Set Theory, thus the coverage is usually less than or equal to 1.

$$Coverage = \frac{\bigcup_{i=1,\dots,N} A_i}{A}. \quad (13)$$

The uniformity of coverage is a well-defined standard to measure the service life of a network. Article [17] describes the concept as the standard deviation of distance between nodes. Smaller standard deviation means better coverage uniformity of the network, as shown in Equation (14).

$$\begin{cases} U_i = \left[\frac{1}{n}\sum_{j=1}^{n}(d_{i,j} - \bar{d})^2\right]^{\frac{1}{2}} \\ U = \frac{1}{N}\sum_{i=1}^{N} U_i \end{cases}, \quad (14)$$

where in Equation (14), U is the Uniformity, N is the total number of nodes, U_i is the standard deviation of distance between the i-th node and its adjacent nodes, n is the number of neighbors of the ith node, $d_{i,j}$ is the distance between i-th and j-th nodes, $\bar{d}$ is the mean of internal distances between the ith node and its neighbors. So far, we have discussed the relation between communication and coverage. Article [16] has proved that when the communication range of node is twice or larger than the sensing range, coverage will contain pure connections. In practical deployment, we only have to consider the coverage so as to ensure the connection. At the moment, coverage contains connection problems.

4.2 SIMULATION

Assuming that the monitoring area of wireless sensor network is a 50m×50m square, perception radius of each sensor node R_s=5m, and communication radius $R_c=2R_s$=10m. 35 nodes are randomly deployed in the monitoring area. As particles are diversified, when the number of particles is larger, distribution of nodes may be relatively even. However, with the number of particles increases, the calculation duration may increase exponenttially, largely reducing the calculation speed. Taking into consideration the above factors, the number of particles in the population is assumed to be 30. The flying velocity of particles is limited within -3-3m/s, parameter $c1$=0.9, $c2$=0.9, maximum iteration generation I=500.

The number of sub-populations divided with k-means clustering algorithm may eventually affect the coverage rate of the monitoring area. 50 experiments are conducted under situations with K=2 to 30. On this basis, the average value is figured out, leading to the experimental data shown in Figure 1.

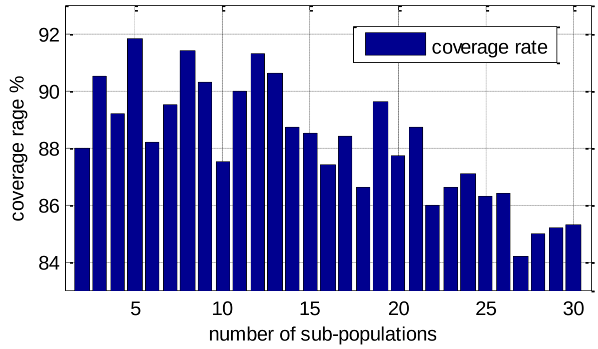

FIGURE 1 Coverage vs. various sub-populations

It can be seen from Figure 1 that, when the number of sub-populations is less than half of the population scale, the coverage rate is generally higher than that when the number of sub-populations is more than half of the population scale. As for this, the number of sub-populations is controlled within 3-8, and the effect would be better. The reason is that, when the number of sub-populations is relatively less, a certain amount of particles for each sub-population could be guaranteed. In essence, PSO algorithm is a sort of swarm intelligence algorithm. The velocity of particle is co-determined by its flying experience and companions' flying experience. Population diversifycation of each sub-population shall be guaranteed, so that particles in each sub-population may be able to interact and to exchange information. On the contrary, when the number of divided sub-populations is larger, there might be fewer particles for each sub-population. As for this, particles may be lack of social information sharing, and the probability of reaching optimal solution would be lower. According to the experiment, when the population is divided into 5 sub-populations, wireless sensor network deployment optimization effect is the best, and the coverage rate at the moment is 91.89%.

In order to test the effectiveness of DMPSO algorithm, simulation experiment based on two models is performed. In other words, standard PSO algorithm and improved DMPSO algorithm are separately applied to optimize node deployment for wireless sensor network. Figure 2a shows the initial status of node random deployment; Figure 2b is wireless sensor network deployment result based on the optimized PSO algorithm; Figure 3c is wireless sensor network deployment result based on the improved DMPSO algorithm.

In the Figure 2, solid black dots indicate the position of sensor nodes, while circles refer to the perception range of nodes. After 500 times' iteration, the coverage rate of wireless sensor network deployed with standard PSO algorithm is 88.23%; the coverage rate of wireless sensor network deployed with DMPSO algorithm is 91.89%. Thus, it can be seen that, the coverage rate of the improved algorithm is increased by 3.66%, reaching the effect of deployment optimization. Shown by Figure 2, node distribution in (c) is more even than that in (b), so that the rate of "dead zone" and repeat coverage is relatively lower. DMPSO algorithm divides particles into several sub-populations via clustering. On this basis, particles in each sub-population will be able to adjust the flying direction according to their respective flying experience and the "global optimum" of their sub-population. When the frequency of worst fitness of a certain particle reaches the pre-set value, the particle will consequently be regarded as being unfit for present searching environment, which needs to be optimized. As for this, the particle will be removed from regional optimal value for optimal solution. In this way, particles may be able to get rid of regional optimal value, to expand the searching range, and to effectively solve the problem of "precocity".

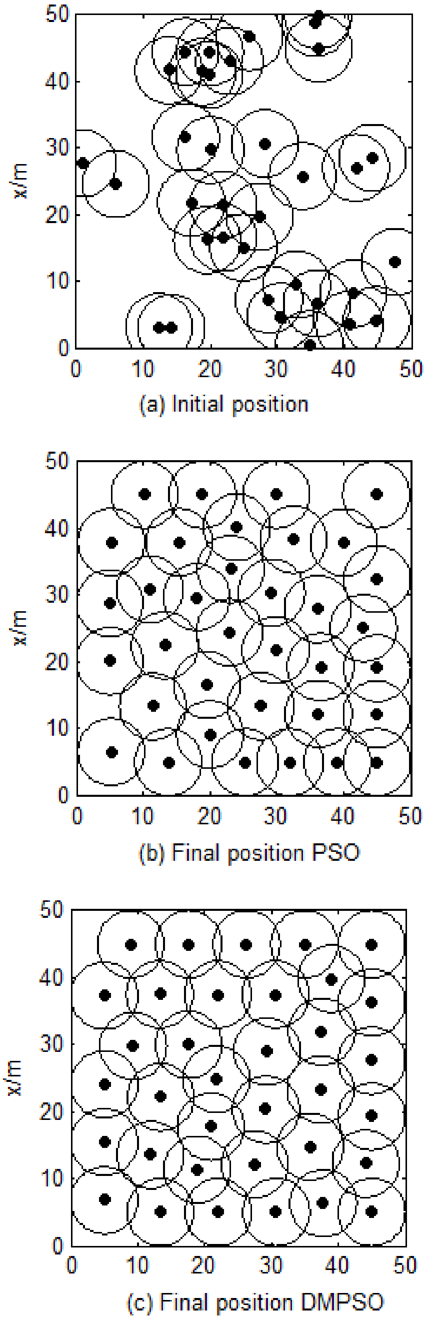

FIGURE 2 Deployment of sensor nodes

Figure 3 shows coverage and uniformity changing curve of DMPSO algorithm. According to the figure, the initial coverage rate of network is approximately 35%. After 500 iterations, the coverage rate is significantly improved. Seeing from the growth slope of the curve, in the first 100 generations of iteration, the slope of curve is relatively higher, and the coverage rate increases sharply. After 150 iterations, the slope reduces obviously, and the coverage rate grows gently, eventually being stabilized to a constant. The reason for this is that, under the influence of inertia coefficient, the algorithm is easy to converge to a global best position in the early stage. With the number of iteration increases, particles begin to oscillate around the best position, so that the result obtained tends to be stabilized to the best result 91.89%. Compared with the initial coverage rate, the coverage rate at the moment is improved by approximately 55%. Thus, it may be seen that, the algorithm shows more obvious effect in improving network coverage rate.

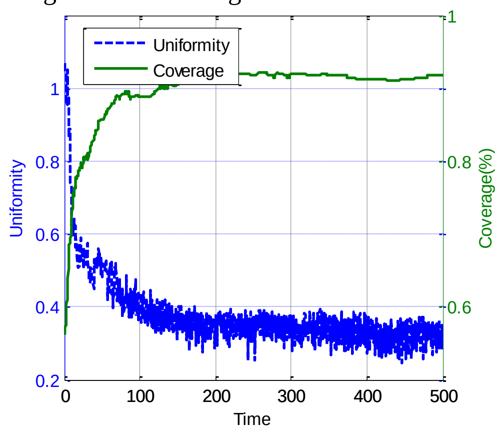

FIGURE 3 Coverage and Uniformity of DMPSO

Figure 4 shows the coverage rate of standard PSO algorithm and DMPSO algorithm. According to Figure 4, the number of iteration of standard PSO algorithm when reaching convergence is around 350, and for DMPSO algorithm, the number is around 300. The convergence speed of the improved algorithm is improved by approximately 14%.

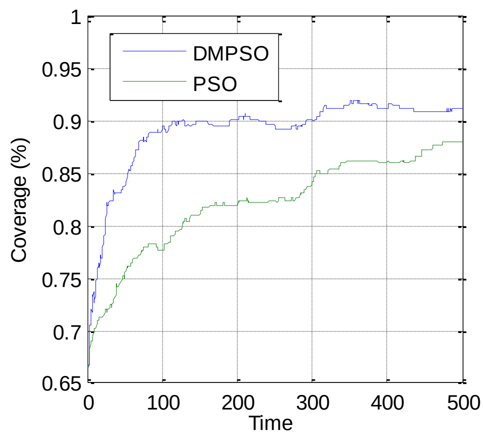

FIGURE 4 Comparison of PSO and DMPSO coverage

Thus, it may be seen that, in wireless sensor network deployment optimization, DMPSO algorithm has better convergence performance than that of PSO algorithm.

In order to further test the feasibility of DMPSO algorithm, common genetic algorithm, and bee colony algorithm are employed to compare with DMPSO algorithm proposed in this paper. The number of iteration of all the three algorithms is 500. Table 1 shows simulation result comparison of the 3 algorithms after 50 times of statistics.

TABLE 1 Comparison of various algorithms

Algorithm	Coverage %	Iteration
Genetic Algorithm	80.3	485
Bee Colony Algorithm	85.2	426
DMPSO	91.89	308

Shown by Table 1, compared with genetic algorithm and bee colony algorithm, DMPSO algorithm improves network coverage rate separately by 11.59% and 6.65%. Moreover, DMPSO algorithm reaches the best solution within less generation. As for this, it is feasible to apply DMPSO algorithm to optimize wireless sensor network node deployment.

4 Conclusions

In this paper k-means clustering method is introduced to propose a particle swarm model for dynamically dividing sub-populations. The algorithm effectively solves the "precocity" problem of standard PSO algorithm. Moreover, the simulation result also shows that, the proposed DMPSO algorithm is able to optimize the deployment of WSN node, and to improve the network coverage. How to reduce the repeat coverage of node, so as to further improve network coverage rate is a key problem to be further studied in future.

References

[1] Losilla F, Garcia-Sanchez A J, Garcia-Sanchez F, et al.2011 A comprehensive approach to WSN-based ITS applications: A survey *Sensors* **11**(11) 10220-65

[2] Kulkarni R V, Venayagamoorthy G K. 2011 Particle swarm optimization in wireless-sensor networks: A brief survey *Systems, Man, and Cybernetics Part C: Applications and Reviews IEEE Transactions on* **41**(2) 262-7

[3] Alippi C, Camplani R, Galperti C, et al. 2011 *Sensors Journal IEEE* **11**(1) 45-55

[4] Zhang J, Varadharajan V 2010 Wireless sensor network key management survey and taxonomy *Journal of Network and Computer Applications* **33**(2) 63-75

[5] Ren Y, Zhang SD, Zhang HK 2006 Theories and algorithms of coverage control for wireless sensor networks *Journal of Software* **17**(3) 422-33

[6] Yiyue W, Hongmei L, Hengyang H 2012 Wireless sensor network deployment using an optimized artificial fish swarm algorithm *Computer Science and Electronics Engineering (ICCSEE) International Conference on. IEEE* 2 90-4

[7] Zou Y, Chakrabarty K 2003 Sensor deployment and target localization based on virtual forces *Proceedings of the 22ndAnnual Joint Conference of the IEEE Computer and Communications SanFrancisco USA IEEE* 1293-303

[8] Wang XQ, Yang YT 2006 Sensor deployment algorithm based on virtual rhomb grid *Computer Applications* **26**(7) 1554-6

[9] Kennedy J, Eberhar R 1995 Particle swarm optimization *in: Proceedings of the 4thIEEE International Conference on Neural Networks Piscatawya:IEEE Service Center* 1942-8

[10] Burne R A,Buczak A L,JIN Yao-chu 1999 Self-organization cooperative sensor network for remote surveillance: Current result

[11] Wang X, Wang S, Ma J J 2007 Parallel Particle Swarm Optimization Based Mobile Sensor Node Deployment in Wireless Sensor Networks **30**(4) 563-9

[12] Xiang XX, Huang HG, Li YD 2010 Hybrid sensor networks coverage-enhancing approach based on particle swarm optimization *Application Research of Computers* **27**(6) 2273-5

[13] Lin ZL, Feng YJ 2009 Research on strategy for optimizing coverage of WSNs based on multi-particle PSO *Application Research of Computers* **26**(12) 4701-3

[14] Clerc M 2010 Particle swarm optimization *John Wiley & Sons*

[15] Zhang H-h, Hou J C 2005 Maintaining sensing coverage and connectivity in large sensor networks *Wireless Ad Hoc and Sensor Networks* **1**(1) 89-124

[16] Gage, Douglas W 1992 Command control for many-robot systems. *Naval command control and ocean surveillance center rdt and e div San Diego CA*

[17] Nojeong H, Varshney P K 2003 A distributed self spreading algorithm for mobile wireless sensor networks *Wireless Communications and Networking WCNC IEEE* 3

Author

Lei Hong, born in Mart, 1976, Jiangsu, China.

Current position: a lecturer at the School of Information Technology, Jinling Institute of Technology, China.
University studies: Bachelor Degree from Anhui Normal University in 1998, the Master. Degree from Hohai University in 2005.
Scientific interest: software engineering and wireless sensor network.
Publications: 3 papers.