International Journal of Grid and Distributed Computing
Vol. 9, No. 9 (2016), pp.293-306
http://dx.doi.org/10.14257/ijgdc.2016.9.9.25

Optimal Coverage Algorithm of Wireless Sensor Networks Based on Particle Swarm Optimization with Coherent Velocity

Chuanyun Wang[1], Enyan Sun[1] and Feng Tian[2]

1College of Computer Science, Shenyang Aerospace University, Shenyang, China
2College of Automation, Shenyang Aerospace University, Shenyang, China
wangcy0301@sau.edu.cn

Abstract

Optimal coverage of wireless sensor networks is one of the most fundamental problems for constructing efficient perception layer of the Internet of Things. On the basis of research on spatial neighborhood, the node coverage and area coverage models are analyzed, then an optimal coverage algorithm of wireless sensor networks is proposed based on particle swarm optimization with coherent velocity. Experimental results show that the algorithm can significantly improve the network coverage; in addition, the coherent velocity can effectively avoid network prematurely into a local optimal solution, so as to enhance the network coverage.

Keywords*: wireless sensor networks; optimal coverage; particle swarm optimization; coherent velocity*

1. Introduction

Because of the advantages of deployment flexibility, environmental adaptability, network self-organization and easy to extend, wireless sensor networks have attracted tremendous research interest in academia and industry recently. In the Internet of Things era, all objects and information access networks, so wireless sensor networks is particularly important as an ubiquitous information perception network. It is flexible and fast to construct perception layer include monitoring nodes and aggregation nodes, then a variety of environments and monitoring targets information is acquired and aggregated by multi-node collaboration to achieve seamless integration of the real physical world and the virtual information world [1]. It completely changes the interact manner between humans and nature.

Optimal coverage of wireless sensor networks is one of the most fundamental problems for constructing efficient perception layer network. It not only achieves network coverage maximization, but also avoids coverage of blind spots [2]. Sensor nodes are divided into static nodes and dynamic nodes. Static nodes are easy to achieve coverage optimization in the case of artificial deployment, while redundant deployment and sleep scheduling of nodes are adopted to achieve maximize coverage optimization in the case of random deployment. Mobile robot or simple activity node is generally used as dynamic node to take on environmental monitoring and data gathering in a special area, so the mobility is fully utilized to expand the monitoring range, reduce blind spots, and enhance the network coverage [3].

In this paper, on the basis of research on spatial neighborhood, the node coverage and area coverage models are analyzed, then an optimal coverage algorithm of wireless sensor networks based on particle swarm optimization with coherent velocity is proposed. The algorithm regards the network coverage maximization as fitness function of coverage optimization, and inertial velocity, individual acceleration, coherent velocity are comprehensively considered to achieve node moving to the optimal position for optimal coverage of the monitoring area [4].

ISSN: 2005-4262 IJGDC

The rest of this paper is organized as follows: In Section 2, we give an overview of the state of the art regarding deployment and optimize coverage algorithm of wireless sensor networks. Section 3 describes the preliminary of spatial neighborhood organization, node and area coverage model. Section 4 provides the overall architecture of our proposed coverage optimization algorithm. The implementation procedures are detail described in Section 5. In Section 6, the experimental results are presented and analyzed. Finally, we finish with conclusions in Section 7.

2. Related Works

The effectiveness of cluster-based distributed sensor networks depends on a large extent of the coverage provided by the sensor deployment. Zou *et al.* proposed a virtual force algorithm (VFA) as a sensor deployment strategy to enhance the coverage after an initial random placement of sensors [5]. Benyuan *et al.* defined three coverage measures to characterize the area coverage, node coverage and the capability to detect objects moving in the network, then approached the coverage problem from a theoretical perspective and exported the fundamental limits of the coverage of a large scale sensor networks [6]. Honghai *et al.* addressed the issues of maintaining sensing coverage and connectivity by keeping a minimum number of sensor nodes in the active mode in wireless sensor networks, and devised a decentralized optimal geographical density control (OGDC) algorithm for density control in large scale sensor networks [7]. Due to constraint of associated battery power, Demin *et al.* regarded coverage and lifetime as two paramount problems, and provided an analytical framework for the coverage and lifetime of wireless sensor networks with 2D Gaussian distribution [8].

Particle swarm optimization is widely used in wireless sensor networks. Jize *et al.* corrected situations by employing some mobile robots as mobile nodes in wireless sensor networks which can actively move to desired locations for repairing the broken networks. The algorithm named particle swarm genetic optimization (PSGO), which imported selection and mutation operators in the PSO to overcome the premature fault of classical PSO, was proposed to redeploy the mobile robots according to the node density for repairing the sensing coverage hole after their initial random deployment [9]. Aiming at the coverage problem of wireless sensor networks, Aziz *et al.* proposed an algorithm to optimize sensor coverage using PSO and Voronoi diagram, in which PSO was used to find the optimal deployment of the sensors that gave the best coverage and Voronoi diagram was used to evaluate the fitness of the solution [10-11]. Zhiming *et al.* provided a method of improved particle swarm optimization (IPSO) to solve the node deployment problem in wireless sensor networks which was always consist of stationary and mobile sensor nodes, and evaluated the coverage ratio achieved using the traditional VFA and IPSO [12]. Xingzhen *et al.* applied PSO to maximize the coverage of mobile sensor networks in limited mobility model, and reduced energy consumption during the node move process [13].

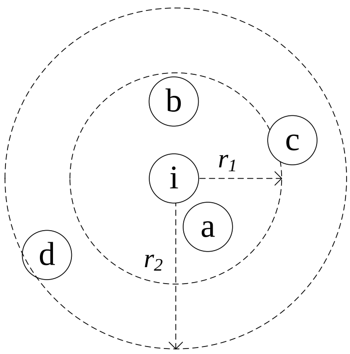
Due to the number of sensors, region of interest (ROI) and limited sensing range aspects, Wan *et al.* provided a solution by engaging grid diagram with PSO [14]. Xue *et al.* proposed a virtual force directed co-evolutionary particle swarm optimization (VFCPSO) algorithm with the collaboration of multiple sensor nodes, in which a combined objective function was used to achieve the tradeoff of coverage and energy consumption [15]. Salehizadeh *et al.* proposed a node deployment algorithm for mobile sensor networks based on individual particle optimization (IPO) for the purpose of maximum coverage in environment, and the mobile nodes would relocate themselves to find the best deployment under various kinds of situations in order to cover the largest area [16]. Aziz *et al.* took the energy consumption into account, and gave a two phase PSO algorithm towards coverage maximization and energy conservation for mobile wireless sensor networks. Both objectives were tackled in separate phases with coverage maximization in the first phase while energy conservation in the second phase [17].

For investigating the performance of different paradigms, Xue *et al.* extended the centralized VFCPSO to distributed VFCPSO, heterogeneous hierarchical VFCPSO and homogeneous hierarchical VFCPSO (Homo-H-VFCPSO), and the solution of preferential deployment in interested region was also analyzed [18]. Joon-Woo *et al.* proposed a new approach to solving the efficient-energy coverage problem using three pheromones ant colony optimization (TPACO) algorithm, in which the local pheromone helped an ant organize its coverage set with fewer sensors. The other two global pheromones were used to optimize the number of required active sensors per Point of Interest (PoI), and to form a sensor set that had as many sensors as an ant had selected the number of active sensors by using the former pheromone [19].

3. Preliminary Modeling

3.1. Spatial Neighborhood

In particle swarm optimization, mutual exchanges could be achieved through the exchange of successful experience between the particles in the same neighborhood. In wireless sensor networks, it is not suitable for global optimization by whole network communication in the condition of nodes with limited communication capability and energy. In order to get the trade-off between convergence speed and optimal solution, the nodes organize a certain number of neighbours to construct a spatial neighborhood, in other words, they decompose the global optimal solution into multiple local optimal solution of spatial neighborhood [20]. There are two methods of construction: (1) neighborhood with fixed radius, in which the number of nodes in each neighborhood may be different; (2) neighborhood with certain number, in which the node should dynamically adjust the communication radius to communicate with neighbours [21]. Figure 1 shows the spatial neighborhood U_i construction of node *i*, in which (b) and (c) indicate the different results by method (1) and (2), respectively.



(a) Neighbors deployment of *i*

R_c	U_i
r_1	{i, a, b}
r_2	{i, a, b, c, d}

(b) U_i with fixed radius

N_u	U_i
1	{i}
3	{i, a, b}
5	{i, a, b, c, d}

(c) U_i with certain number

Figure 1. Spatial Neighborhood of Sensor Node

3.2. Node Coverage Model

The cover range of sensor node s_i is the circular area of its own coordinates (x_i, y_i) as the centre and monitoring distance r as the radius. The Euclidean distance between the sensor node s_i and the monitoring target k with coordinates (x, y) is d_{ik}, and computed as follows:

$$d_{ik} = \sqrt{(x_i - x)^2 + (y_i - y)^2} \tag{1}$$

Due to environmental noise and signal strength attenuation with larger transmission distance, the monitoring ability of sensor nodes shows some uncertainty. p_{ik} is used to indicate the probability of monitoring target k covered by node s_i, and computed as follows:

$$p_{ik} = \begin{cases} 1 & \text{if } d_{ik} \le r - r_e \\ e^{-\alpha(d_{ik}-(r-r_e))} & \text{else if } r - r_e < d_{ik} < r + r_e \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where r is the monitoring radius, $r_e(0< r_e <r)$ is the uncertain monitoring radius error of node, and α is the uncertain monitoring coefficient which reflects the fitness of environment for monitoring. So the taking value of α is depend on uncertain environmental noise. Figure 2 shows the relationship between p_{ik} and d_{ik} with different uncertain monitoring coefficient. According to the Figure, we can select a bigger coefficient to reduce the coverage probability rapidly in a complex monitoring environment.

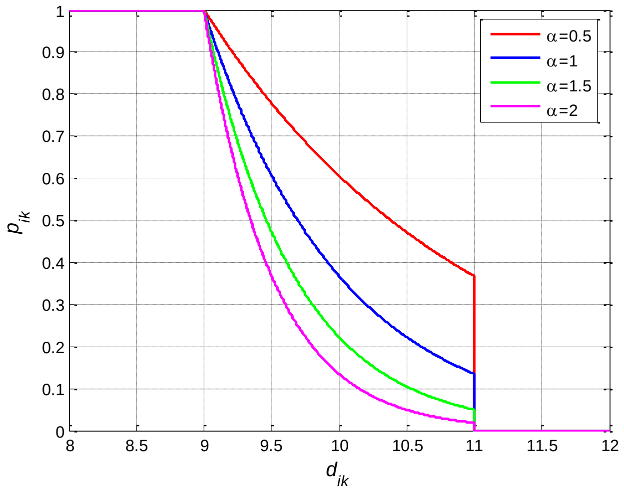


Figure 2. Probability of Monitoring Target *k* Covered by Node s_i

Any monitoring target k may be covered by multiple sensor nodes, so the combined coverage P_k is formed by all the nodes cover probability, and computed as follows:

$$P_k = 1 - \prod_i (1 - p_{ik}) \tag{3}$$

3.3. Area Coverage Model

Assume that the monitoring area A is divided into $m \times n$ pixel points in two-dimensional, so there are $m \times n$ monitoring targets. The number of sensor nodes deployed in the area is N, and the set of nodes is marked as $S=\{s_0,s_1,s_2,\ldots\ldots s_N\}$. The whole network coverage R_{area} is the ratio of combined coverage of nodes set S to monitoring area acreage, and computed as follows:

$$R_{area} = \frac{\sum_S P_k}{m \times n} \tag{4}$$

Because of the limited ability of sensor nodes, it is rather complex for optimal coverage of the whole network, and the coverage calculation within the local neighborhood of node i is more suitable. The local neighborhood coverage what is marked as R_i is the ratio of the combined coverage of nodes in the spatial neighborhood U_i to according proportion of monitoring area acreage, and computed as follows:

$$R_i = \frac{\sum_{U_i} P_k}{m \times n \times \frac{Nu}{N}} \tag{5}$$

4. Coverage Optimization Algorithm

4.1. Node Position Optimization

In this algorithm, the maximization neighborhood coverage R_i of node i is regarded as the optimization goal. It is a function of the number of rounds iteration t, marked as $f(t)$. Each sensor node is seen as a particle individual, and its coordinates change with t [22]. In order to facilitate the presentation, there does not distinguish between the dimension of node coordinates, and entirely mark the coordinates of node i at the number of rounds t as $L_i(t)$.

For each individual, the optimal position marked as $B_i(t)$ is the best position for node i located from the beginning to the number of rounds t, then the individual best position of node i at the number of rounds t+1 is computed as follows [23]:

$$B_i(t+1) = \begin{cases} L_i(t+1) & \text{if } f(L_i(t+1)) > f(B_i(t)) \\ B_i(t) & \text{otherwise} \end{cases} \tag{6}$$

The position movement of node i from the number of rounds t to t+1 is depend on a certain velocity marked as $V_i(t+1)$. It is formed by inertial velocity, acceleration and coherent velocity, and it is computed as follows:

$$V_i(t+1) = wV_i(t) + c_1 \times r_1(t) \times (B_i(t) - L_i(t)) + c_2 \times v_{c,i} \tag{7}$$

where w is the inertia weight of velocity; c_1 and c_2 are the individual acceleration and coherent velocity scale factor respectively; $r_1(t)$ is a random number on [0,1] at the number of rounds t; $v_{c,i}$ is the coherent velocity of node i.

After each movement, the coordinates of node i is need to be updated. The change of position from the number of rounds t to t+1 is the velocity, so it can be computed by following equation at the number of rounds t+1.

$$L_i(t+1) = L_i(t) + V_i(t+1) \tag{8}$$

In order to avoid the node moving out of the monitoring area, a reverse ricochet strategy is used after collision, and the distance of reverse ricochet is calculated as follows:

$$L_i(t) = \begin{cases} 2L_{\min} - L_i(t) & \text{if } L_i(t) < L_{\min} \\ 2L_{\max} - L_i(t) & \text{if } L_i(t) > L_{\max} \end{cases} \tag{9}$$

where $L_{\min}$ is the minimum coordinates of monitoring area, and $L_{\max}$ is the maximum coordinates of monitoring area.

4.2. Inertia Weight of Velocity

Both the local and global optimization will benefit solving some kinds of problems. There is a trade-off between the global and local optimization for different problems. Considering of this, an inertia weight of velocity w is introduced. The w, which reflects the memory of movement history, plays the role of balancing the global and local optimization, controls the influence of previous velocity on next movement. The larger value of w is conducive to exploration and increase the diversity of possible solutions, while the smaller value of *w* will enhance local development capabilities. It means that the inertia weight of velocity *w* can balance the local and global during the optimization process, and the values of *w* can be a fixed value, also can dynamically change with generation number. Three methods of value selection are described as follows:

(1) Random adjustment

In each round iteration, a randomly *w* within the range [0, 1] is selected, such as from the Gaussian distribution [24]:

$$w \sim N(\mu, \sigma) \tag{10}$$

In which, we can let μ =0.729， σ =0.5 inspired by predecessors' constriction factor concept.

(2) Linear decreasing

A linear approach for *w* can be used to decrease from the beginning of larger value to a smaller value [25-26]:

$$w(t) = \left(w(0) - w(n_t)\right)\frac{(n_t - t)}{n_t} + w(n_t) \tag{11}$$

where n_t is the maximum number of rounds iteration executed by the algorithm, and $w(0)$ and $w(n_t)$ are the initial and final inertia weight of velocity, as well as $w(t)$ is the value of inertia weight at the number of rounds *t*.

According to the previous work, the optimal solution can be improved by varying the value from 0.9 at the beginning to 0.4 at the end for most problems.

(3) Non-linear decreasing

On the other hand, a non-linear approach for *w* can be adopted to decrease from the beginning of larger value to a smaller value [27]:

$$w(t+1) = \frac{\left(w(t) - 0.4\right)(n_t - t)}{n_t + 0.4} \tag{12}$$

where n_t is the total number of generations for which the algorithm runs and *t* is the present generation number, and let $w(0)$=0.9 in general.

The inertia weight of velocity selection is flexible as described above. The random adjustment method is simple and effective, but ignores the influence of the number of rounds iterations, so this method reduces the speed of convergence; the linear or non-linear decreasing method significantly improves the speed of convergence, and the non-linear decreasing method is more considering the difference of different iterative phase, to improve the ability of the algorithm to explore a better solution.

4.3. Coherent Velocity

In order to avoid the neighborhood nodes at the same time moving towards a same position, resulting in a local optimum rather than search for a better solution, a definition of coherent velocity is introduced, and then it is used to explore better solution by the exclusion mechanism between the nodes to force the movement direction diverge out.

Definition 1: In wireless sensor networks, the velocity of sensor nodes is mutual influence and interference. It means that any change of node in direction or speed may cause all neighbors movement adjustment, and these adjustments, in turn, inhibit or encourage the change. This velocity in direction or speed is called coherent velocity [28].

A coherent velocity calculation method can be used [29-30], which is defined as follows:

$$v_{c,i} = f_{s,i}(C, \rho, \gamma)\bar{v}_i \xi_i \tag{13}$$

where ξ_i is random number sampled from standard Cauchy distribution. The Cauchy density function is shown as follows:

$$f(x) = \frac{1}{\pi\left(1 + x^2\right)} \tag{14}$$

In equation (13), $\overline{v}_i$ is the average velocity of local neighborhood nodes for node *i* with the number of n_s, and it can be computed as follows:

$$\overline{v}_i = \frac{\sum_{j=1}^{n_s} v_j}{n_s} \tag{15}$$

In equation (13), $f_{s,i}(C, \lambda, \gamma)$ is a S-shaped function, and *C* is the degree of coherence of neighborhood movement which is the ratio of neighborhood center speed v_s to node average speed $\overline{v}$, and these variables are computed by following equations:

$$f_{s,i}(C, \rho, \gamma) = \frac{1}{1 + e^{\rho(C-\gamma)}} \tag{16}$$

$$C = \frac{v_s}{\overline{v}} \tag{17}$$

$$v_s = \left\| \frac{\sum_{j=1}^{n_u} v_j}{n_u} \right\| \tag{18}$$

$$\overline{v} = \frac{\sum_{j=1}^{n_u} \left\| v_j \right\|}{n_u} \tag{19}$$

where ρ and γ are the width and offset of S-shaped function respectively.

5. Implementation Description

5.1. Node Attributes

In this algorithm, each node is an independent individual that records the information obtained and used for coverage optimization. These information include coordinate of x/y-axis, spatial neighborhood, local best coverage, velocity and coherent velocity *etc.*. The structure of node attributes is shown in Table 1.

Table 1. Structure of Node Attributes

Attribute of node s_i	*Description*
x	Coordinate of x-axis
y	Coordinate of y-axis
U	Set of spatial neighbourhood
Nu	Number of neighbourhood
BC	Local best coverage
Bx	Local best coordinate of x-axis
By	Local best coordinate of y-axis
Vx	Velocity of x-direction
Vy	Velocity of y-direction
VCx	Coherent velocity of x-direction
VCy	Coherent velocity of y-direction

5.2. Spatial Neighborhood Construction

Spatial neighborhood is a prerequisite for local coverage optimization. The neighborhood of a node changes constantly, so the neighborhood should be updated in each round iteration of coverage optimization. As described in Section 1.1, the construction of spatial neighborhood could be made by two methods, and algorithm 1 shows the procedure of construction.

Algorithm 1. Neighborhood with fixed radius *Rc*

Input: the coordinates (x_i, y_i) of sensor node s_i and the fixed radius Rc
Output: Set of spatial neighborhood U and the number of neighborhood Nu
Counter c=0;
For each node k=1,2,…,N **Do**
    compute distance d between nodes by equation (1)
    If d<=Rc **Then**
        c=c+1;
        $U(c)$=k;
    End
End
Nu=c;

Algorithm 2 describes the procedure of spatial neighborhood construction in the case of nodes with certain number of neighbors.

Algorithm 2. Neighborhood with certain number *Nu*

Input: the coordinates (x_i, y_i) of sensor node s_i and the certain number Nu
Output: Set of spatial neighborhood U
For each node k=1,2,…,N **Do**
    compute distance $D(k)$ between nodes by equation (1)
End
sort $D(k)$ in ascending order
For k=1,2,…,Nu **Do**
    construct spatial neighborhood U with nearest Nu nodes
End

5.3. Local Neighborhood Coverage Computation

Statistical analysis of the combined coverage of the discrete monitoring targets in local neighborhood monitoring area is made by the equation (3), and local neighborhood coverage is computed by the equation (5) what is regarded as fitness function of coverage optimization to maximize the network coverage. In fact, the coverage of the whole network is a special case of local neighborhood coverage when the node communication radius is large enough or the number of neighborhood nodes is sufficient. The algorithm 3 describes the local neighborhood coverage computation procedure.

Algorithm 3. Coverage of local neighborhood

Input: sensor node s_i and its set of spatial neighborhood U
Output: the local neighborhood coverage R_i
For each monitoring target k=1,2,…,M **Do**
    Initial cover probability Pm=1;
    For each node in neighborhood j=1,2,…,Nu **Do**
        compute distance d between node in neighborhood and monitoring target by equation (1)
        compute the node coverage p by equation (2)
        Pm= Pm×(1-p)
    End
    $P(k)$=1-Pm;
End
compute the coverage of local neighborhood by equation (5)

5.4. Procedure of Coverage Optimization

The optimal coverage algorithm of wireless sensor networks based on PSO is achieved by many rounds iteration. The conditions of iteration terminate can be pre-set maximum number of rounds iteration, and also a certain network coverage, as well as no change in the coverage within a certain number of rounds iteration. The first condition is selected in this algorithm. In the each round iteration of optimization procedure, multiple steps are operated, such as the inertia weight of velocity computation, spatial neighborhood construction, local neighborhood coverage computation, best position update, coherent velocity update, velocity and position of node update *etc.* The algorithm 4 describes the coverage optimization procedure.

Algorithm 4. Coverage optimization procedure

```
Input: the initial coordinates of all sensor nodes
Output: the optimized coordinates of all sensor nodes
For each round iteration r=1,2,…,rmax Do
    compute the inertia weight of velocity by equation (10) or (11) or (12)
    For each node i=1,2,…,N Do
        construct spatial neighborhood by algorithm 1 or 2
        compute local coverage fc of node s_i by algorithm 3
        If fc> s_i.BC Then
            update BC and Bx, By of s_i by equation (6)
        End
    End
    For each node i=1,2,…,N Do
        update VCx, VCy by equation (13)-(19)
    End
    For each node i=1,2,…,N Do
        update velocity Vx, Vy by equation (7)
        update coordinate x, y by equation (8)
        avoid the node out of monitoring area by equation (9)
    End
    compute the network coverage by equation (4)
End
```

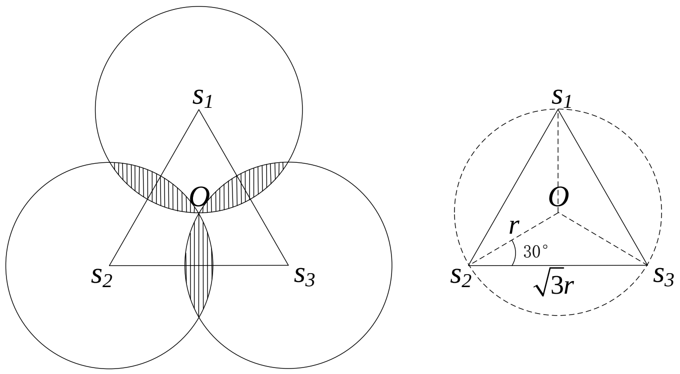
6. Experiment and Analysis

6.1. Experimental Settings

The experiment is performed in the environment of Matlab software. In view of this algorithm is a downright local optimization, the monitoring scale is unrestricted and without consideration. Therefore, a monitoring area of 100m×100m is used in this experiment with 50 sensor nodes randomly deployed. The monitoring radius and uncertain monitoring radius error of node are 10m and 1m respectively, while the uncertain monitoring coefficient is 0.5. The individual acceleration and coherent velocity scale factor are 1 and 0.1 respectively. The width and offset of S-shaped function are -0.5 and 0.5, respectively, and the maximum number of rounds iteration is 100.

6.2. Communication Radius Selection

In the ideal condition, the uncertain monitoring radius error r_e is 0, so the monitoring radius is a fixed value r. According to the equation (5), the local neighborhood coverage combined by s_1, s_2, s_3 is maximal in the case of seamless coverage topology, shown in Figure 3(a). In other words, the limit of zero for the overlapping coverage area O of three nodes s_1, s_2, s_3 leads to optimal coverage.



(a) Seamless coverage topology (b) Distance relationship

Figure 3. Communication Radius Analysis and Selection

The distance relationship between communication radius and monitoring radius is shown in Figure 3(b), and it could be computed as follows:

$$R_c = 2 \times r \times \cos(30°) = \sqrt{3}r \tag{20}$$

In the real condition, the uncertain monitoring radius error r_e must be considered. The monitoring radius is r-r_e, so the communication radius should be selected as follows:

$$R_c = 2 \times (r - r_e) \times \cos(30°) = \sqrt{3}(r - r_e) \tag{21}$$

6.3. Network Coverage Discussion

In order to verify the influence of network coverage in the case of spatial neighborhood constructed by different methods, different communication radius and different number of neighborhood nodes, the corresponding experiments are made, and the results are shown in Table 2 and Table 3.

From Table 2, we can see that the communication radius of the nodes has influence to the network coverage while using the spatial neighborhood constructed by fixed radius. When the communication radius is about $\sqrt{3}$ times of monitoring radius, the network coverage trends to achieve maximum, and the value is 0.91697 when communication radius is 18.

Table 3 tells us that the number of neighborhood nodes has influence to the network coverage while using the spatial neighborhood constructed by certain number neighbors. Too large numbers of neighborhood nodes not only reduce network efficiency, but also further reduce the network coverage.

Table 2. Coverage of Neighborhood with Fixed Radius

Rc	14	16	18	20	22
Coverage	0.85514	0.86685	0.91697	0.89392	0.86154

Table 3. Coverage of Neighborhood with Certain Number

Nu	3	5	7	9	11
Coverage	0.86465	0.85394	0.85873	0.78547	0.78709

The following experiments analyze and compare the network coverage of initialization and after 100 rounds iteration using the spatial neighborhood constructed by fixed radius while the value is 18m, and the results are shown in Figure 4.

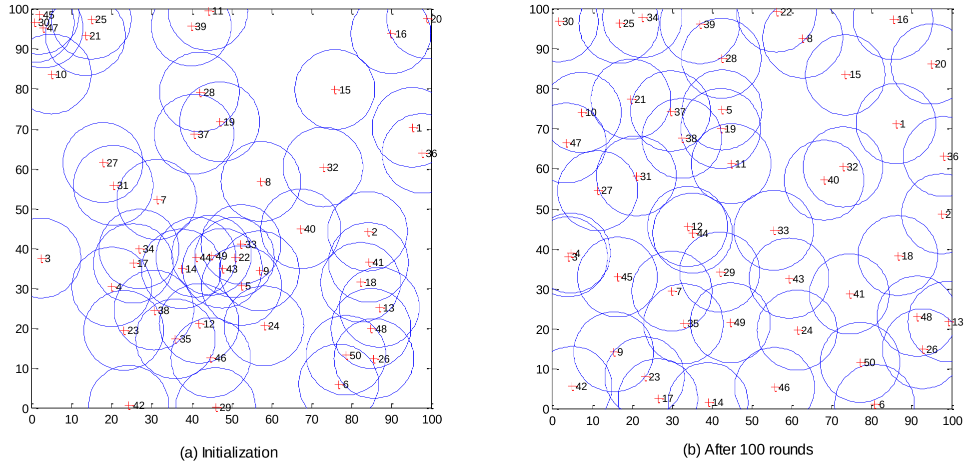


Figure 4. Network of Pre and Post Optimization

From Figure 4, we can see that the network nodes of initialization leading to low network coverage are randomly distributed, and there are a lot of monitoring blind spots and overlapping areas. The distribution of nodes after 100 rounds iteration changes uniform with less monitoring blind spots and overlapping areas, so the network coverage has been greatly improved.

6.4. Coherent Velocity Effects

Statistical analysis of the network coverage in the case of the algorithm with and without coherent velocity is made, and the results are shown in Figure 5.

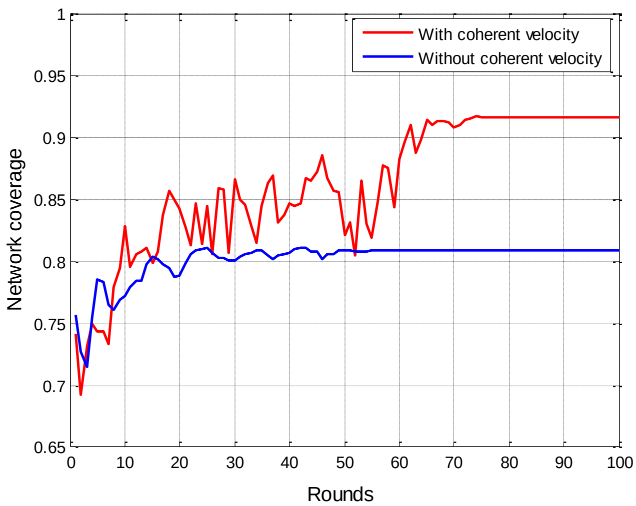


Figure 5. Network Coverage of the Proposed Algorithm

From the Figure, we can see that the network coverage gradually converge after 40 rounds iteration to stabilize with best coverage 0.8179 when using the algorithm without coherent velocity. On the other hand, the network coverage gradually converges after 70 rounds iteration to stabilize with best coverage 0.91697 when using the algorithm with coherent velocity. Therefore, the addition of the coherent velocity significantly avoids the network prematurely into a local optimal solution, and improves the network coverage.

6.5. Running Time Analysis

In this section, we analyze the running time of the algorithm in the case of the algorithm with and without coherent velocity, and the results are shown in Figure 6.

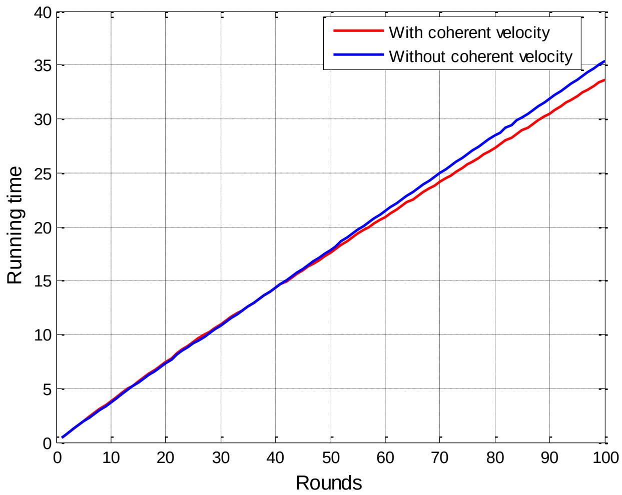


Figure 6. Running Time of the Proposed Algorithm

It is not difficult to recognize the approximate proportional relation between the rounds of iteration and the running time; and then the coherent velocity introduced into particle swarm optimization for network coverage not consumes more time but saves running time. It is because the coherent velocity improves the ability of a node to find the optimum position in local area.

7. Conclusions

In this paper, starting from the point of multi-node collaboration in the perception layer of Internet of Things, an optimal coverage algorithm of wireless sensor networks is proposed based on particle swarm optimization with coherent velocity. The experiments verify the effectiveness of the algorithm; in addition, the coherent velocity plays an important role to improve network coverage. The main contributions are as follows:

First, two construction of spatial neighborhood is presented to decompose the global optimal solution into multiple local optimal solution.

Second, the node and area coverage models are provided as optimal objective function.

Third, a pure local coverage optimization algorithm of wireless sensor networks is detailed description using particle swarm optimization.

Fourth, the coherent velocity is introduced to avoid falling into local optimum and to explore better solution.

Acknowledgement

This work was supported by Educational Department of Liaoning Province of China under Grant No.L2013067. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers which have improved the presentation.

References

[1] Luigi Atzori, Antonio Iera, Giacomo Morabito, "The Internet of Things: A survey", Computer Networks, vol. 54, no. 15, **(2010)**, pp. 2787-2805.

[2] Amitabha Ghosh, Sajal K. Das, "Coverage and connectivity issues in wireless sensor networks: A survey", Pervasive and Mobile Computing, vol. 4, no. 3, **(2008)**, pp. 303-334.

[3] Benyuan Liu, Peter Brass, Olivier Dousse, "Mobility improves coverage of sensor networks",. The 6th ACM international symposium on Mobile ad hoc networking and computing, **(2005)**, pp. 300-308.

[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, March, vol.41, no.2, **(2011)**, pp. 262-267.

[5] Zou Y., Krishnendu Chakrabarty, "Sensor deployment and target localization based on virtual forces", Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, vol.2, no. 30, **(2003)**, pp. 1293-1303.

[6] Benyuan Liu, Towsley D., "A study of the coverage of large-scale sensor networks", 2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems, **(2004)**, October 25-27, pp. 475- 483.

[7] Honghai Zhang, Jennifer C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks", Ad Hoc & Sensor Wireless Networks, vol. 1, **(2005)**, pp. 89-124.

[8] Demin Wang, Bin Xie, Agrawal D.P., "Coverage and Lifetime Optimization of Wireless Sensor Networks with Gaussian Distribution", IEEE Transactions on Mobile Computing, vol.7, no.12, **(2008)**, pp.1444-1458.

[9] Jize Li, Kejie Li, Wei Zhu, "Improving sensing coverage of wireless sensor networks by employing mobile robots", IEEE International Conference on Robotics and Biomimetics, **(2007)**, pp. 899-903.

[10] Aziz N.A.B.A., Mohemmed A.W., Daya Sagar B.S.,. "Particle Swarm Optimization and Voronoi diagram for Wireless Sensor Networks coverage optimization", International Conference on Intelligent and Advanced Systems, **(2008)**, pp. 961-956.

[11] Aziz N.A.B.A., Mohemmed A.W., Alias M.Y., "A wireless sensor network coverage optimization algorithm based on particle swarm optimization and Voronoi diagram", International Conference on Networking, Sensing and Control, **(2009)**, pp. 602-607.

[12] Zhiming Li, Lin Lei, "Sensor node deployment in wireless sensor networks based on improved particle swarm optimization", International Conference on Applied Superconductivity and Electromagnetic Devices, **(2009)**, pp. 215-217.

[13] Xingzhen Bai, Shu Li, Changjun Jiang, "Coverage Optimization in Wireless Mobile Sensor Networks", 2009 5th International Conference on Wireless Communications, Networking and Mobile Computing, **(2009)**, September 24-26, pp. 1-4.

[14] Wan Ismail W.Z., Manaf S.A., "Study on coverage in Wireless Sensor Network using grid based strategy and Particle Swarm Optimization", IEEE Asia Pacific Conference on Circuits and Systems (APCCAS) , **(2010)**, pp. 1175-1178.

[15] Xue Wang, Xinyao Sun, Sheng Wang, "Distributed strategy for sensing deployment in wireless sensor networks", International Symposium on Communications and Information Technologies (ISCIT), **(2010)**, pp. 64-69.

[16] Salehizadeh S., Dirafzoon A., Menhaj M.B., "Coverage in wireless sensor networks based on individual particle optimization". 2010 International Conference on Networking, Sensing and Control, **(2010)**, April 10-12, pp. 501-506.

[17] Aziz N.A.B.A., Mohemmed A.W., Alias M.Y., "Coverage Maximization and Energy Conservation for Mobile Wireless Sensor Networks: A Two Phase Particle Swarm Optimization Algorithm", Sixth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), **(2011)**, pp. 64-69.

[18] Xue Wang, Sheng Wang, "Hierarchical Deployment Optimization for Wireless Sensor Networks", IEEE Transactions on Mobile Computing, vol. 10, no. 7, **(2011)**, pp. 1028-1041.

[19] Joon-Woo Lee, Byoung-Suk Choi, Ju-Jang Lee, "Energy-Efficient Coverage of Wireless Sensor Networks Using Ant Colony Optimization With Three Types of Pheromones", IEEE Transactions on Industrial Informatics, **(2011)**, vol. 7, no .3, pp. 419-427.

[20] Ghosh P., Zafar H., Das S., "Hierarchical dynamic neighborhood based Particle Swarm Optimization for global optimization", Evolutionary Computation (CEC) , **(2011)**, pp. 757-764.

[21] Mohais A.S., Ward C., Posthoff C., "Randomized directed neighborhoods with edge migration in particle swarm optimization". Congress on Evolutionary Computation,, **(2004)**, June 19-23, pp. 548- 555.

[22] Haruna Matsushita, Yoshifumi Nishio, "Network-Structured Particle Swarm Optimizer Considering Neighborhood Relationships", Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA, **(2009)**, June 14-19, pp. 2038-2044.

[23] Min Han, Jianchao Fan, "Particle swarm optimization using dynamic neighborhood topology for large scale optimization", 2010 8th World Congress on Intelligent Control and Automation, **(2010)**, July 7-9, pp. 3138-3142.

[24] Clerc M. & Kennedy J., "The particle swarm explosion, stability, and convergence in a multidimensional complex space", IEEE Transactions on Evolutionary Computation, vol. 6, no. 1, **(2002)**, pp. 58–73.
[25] Shi Y. & Eberhart R., "A modified particle swarm optimizer", Proceedings IEEE International Conference Evolutionary Computation, **(1998)**, pp. 69–73.
[26] Ratnaweera A., Halgamuge S., and Watson H., "Self-organizing hierarchical Particle swarm optimizer with time-varying acceleration coefficients", IEEE Transactions on evolutionary computation, vol. 8, no. 3, **(2004)**, pp. 240-255.
[27] Peram T., Veeramachaneni K., and Mohan C.K., "Fittness-Distance-Ratio based particle swarm optimization", Proceedings of the IEEE swarm intelligence symposium, IEEE press, **(2003)**, pp. 174-181.
[28] Ross J, Badcock D R, Hayes A., "Coherent global motion in the absence of coherent velocity signals", Current Biology, vol. 10, no. 11, **(2000)**, pp: 679-682.
[29] Hendtlass T., Randall M., "A survey of ant colony and particle swarm metaheuristics and their application to discrete optimization problems", Proceeding of the inaugural workshop on artificial life, **(2001)**, pp. 15-25.
[30] Hendtlass T., "Preserving diversity in particle swarm optimization", Lecture Notes in Computer Science, **(2003)**, pp. 31-40.

