

Full Paper

Distributed Particle Swarm Optimization and Simulated Annealing for Energy-efficient Coverage in Wireless Sensor Networks

Xue Wang *, Jun-Jie Ma , Sheng Wang and Dao-Wei Bi

State Key Laboratory of Precision Measurement Technology and Instrument, Department of Precision Instruments, Tsinghua University, Beijing 100084, P. R. China; E-mails: {mjj, wang_sheng00, bdw02}@mails.tsinghua.edu.cn.

* Author to whom correspondence should be addressed; E-mail: wangxue@mail.tsinghua.edu.cn.

Received: 25 April 2007 / Accepted: 8 May 2007 / Published: 10 May 2007

Abstract: The limited energy supply of wireless sensor networks poses a great challenge for the deployment of wireless sensor nodes. In this paper, we focus on energy-efficient coverage with distributed particle swarm optimization and simulated annealing. First, the energy-efficient coverage problem is formulated with sensing coverage and energy consumption models. We consider the network composed of stationary and mobile nodes. Second, coverage and energy metrics are presented to evaluate the coverage rate and energy consumption of a wireless sensor network, where a grid exclusion algorithm extracts the coverage state and Dijkstra's algorithm calculates the lowest cost path for communication. Then, a hybrid algorithm optimizes the energy consumption, in which particle swarm optimization and simulated annealing are combined to find the optimal deployment solution in a distributed manner. Simulated annealing is performed on multiple wireless sensor nodes, results of which are employed to correct the local and global best solution of particle swarm optimization. Simulations of wireless sensor node deployment verify that coverage performance can be guaranteed, energy consumption of communication is conserved after deployment optimization and the optimization performance is boosted by the distributed algorithm. Moreover, it is demonstrated that energy efficiency of wireless sensor networks is enhanced by the proposed optimization algorithm in target tracking applications.

Keywords: Wireless sensor network, deployment optimization, energy efficiency, particle swarm optimization, simulated annealing.

1. Introduction

Wireless sensor networks (WSNs) can implement various complicated tasks in the sensing field via a large number of smart wireless sensor nodes which have sensing, storage, processing and communication capabilities. All the wireless sensor nodes work collaboratively to leverage their individual efforts for the entire application. Since battery-powered wireless sensor nodes are greatly constrained with regards to energy supply, energy efficiency becomes a critical problem in WSNs. As an essential requirement, sensing coverage has been investigated in a few literature reports [1,2]. The coverage problem is defined from several points of view, including deterministic, statistical, worst and best case in [3]. In particular, efficient network deployment considering coverage as well as connectivity is discussed in [4,5]. Target tracking is a typical application for WSNs and poses a great challenge to achieve both high reliability and long lifetime [6]. For WSNs which implement target tracking applications, the efficiency of energy usage should be taken into account in the deployment. Generally, wireless communication spends much more energy than sensing and computation, so it should be the primary consideration [7]. In addition, the potential processing capability of multiple wireless sensor nodes may contribute to better optimization performance [8].

Due to the above-mentioned requirements of deployment in WSNs, we propose distributed particle swarm optimization and simulated annealing (DPSOSA) for energy-efficient coverage. This method takes the energy consumption of target tracking into account to optimize the energy efficiency of WSN coverage with distributed computing. Sensing coverage and energy consumption models for WSNs are introduced first. The purpose of optimization is to find the best deployment of mobile wireless sensor nodes so that the sensing coverage requirement is satisfied and communication energy consumption can be minimized. Then the grid exclusion algorithm is exploited to calculate the coverage rate of specific network deployment, which has minimized computational cost and scalable granularity. We adopt Dijkstra's algorithm to search the lowest cost paths for data collection, which will be regarded as packet transmission paths in target tracking applications. The sensing coverage rate and total energy consumption of data collection are defined as coverage and energy metrics, respectively. The DPSOSA algorithm is then employed to optimize the communication energy consumption under a given sensing coverage requirement. It is executed over a number of nodes, in which the particle swarm optimization (PSO) procedure is aided by the optimization results of simulated annealing (SA) for the global optimal solution. In DPSOSA, a number of particles are given a better view to search for better solutions in their vicinity, by which the PSO procedure can be corrected. Meanwhile, as multiple particles need to be optimized, the optimization task is assigned among wireless sensor nodes to boost up the computational capability. With simulations of deployment optimization and target tracking, the energy efficiency of the proposed distributed optimization algorithm is verified.

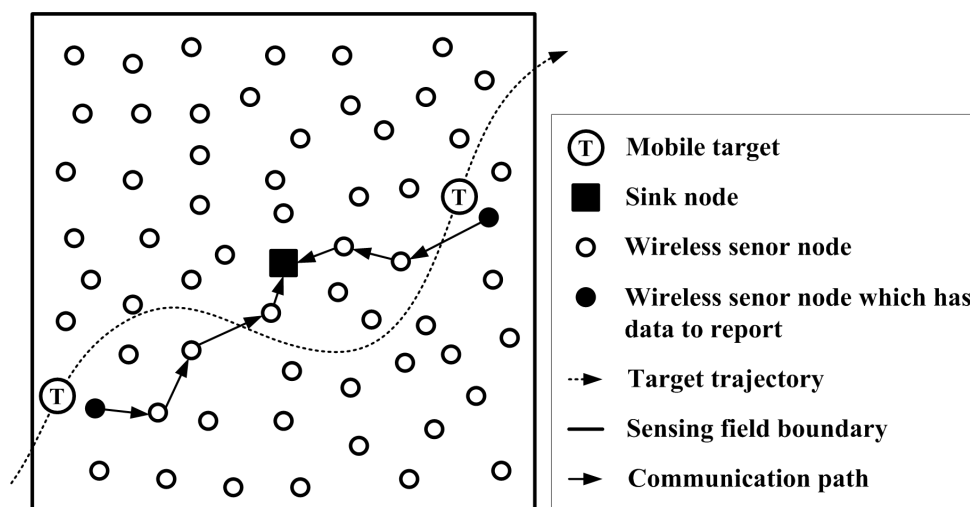
The rest of this paper is organized as follows: section 2 formulates the energy-efficient coverage problem with stationary and mobile wireless sensor nodes in WSNs, where the sensing coverage and energy consumption models are presented. In Section 3, two important metrics, coverage and energy, are defined for network deployment evaluation according to the fundamental model, where the grid exclusion and Dijkstra' algorithm are introduced. Then Section 4 presents the DPSOSA algorithm for energy-efficient coverage in WSNs. In Section 5, we simulate the deployment optimization algorithm

for target tracking application and analyze energy-efficiency of WSNs. We conclude the paper in Section 6.

2. Preliminaries

We assume a WSN composed of two types of wireless sensor nodes: stationary and mobile nodes. In the sensing field, the stationary nodes are deployed randomly, while the mobile ones can adjust their positions adaptively against the environment. With the mobile nodes located at their proper positions, WSN can implement target tracking applications. As shown in Figure 1, wireless sensor nodes which are close to the mobile target trajectory may acquire data. A sink node is located in the centre of sensing field, to which the observations will be forwarded hop by hop [9,10]. It is assumed that the positions of nodes can be obtained by global positioning system (GPS) [11]. In this section, we will describe the sensing coverage model for reliability detection. Considering the energy efficiency problem, energy consumption model of communication will be discussed.

Figure 1. In the target tracking application of WSNs, the mobile target moves through the sensing field and wireless sensor nodes around it will report their data to the sink node in a multi-hop manner.



2.1. Sensing coverage model

Each wireless sensor node integrates three radar sensors with the same sensing radius R , oriented at 120° intervals. Azimuth coverage of radar sensor is $-60^\circ \sim +60^\circ$ [12]. For each wireless sensor node, it is assumed that the strength of received detection signal varies exponentially with the distance from the target. If the coordinates of wireless sensor node i and a target are (x_i, y_i) and (x_{target}, y_{target}) , respectively, the received signal strength reflected off the target is:

$$G_i = G_0 e^{-\beta d_{target,i}} \quad (1)$$

where G_0 is a constant which denotes the strength of emission signal, β is the attenuation constant. And $d_{target,i}$ is the Euclidean distance between the target and sensor:

$$d_{target,i} = \sqrt{(x_{target} - x_i)^2 + (y_{target} - y_i)^2} \quad (2)$$

According to the sensitivity and reliability of sensor, we can define a threshold of received signal strength G_{th} , and calculate the detection reliability as:

$$r_i = \begin{cases} 0 & G_i < G_{th} \\ r_0 & G_i \geq G_{th} \end{cases} \quad (3)$$

where r_0 is the reliability of sensor when the received signal strength exceeds G_{th} , $0 < r_0 < 1$. Thus, wireless sensor node i can physically cover a plate area with radius R_a , where the centre locates at (x_i, y_i) . The sensing radius R_a can be calculated as:

$$R_a = -\frac{1}{\beta} \ln \frac{G_{th}}{G_0} \quad (4)$$

Considering the inherent redundancy of WSNs, we discuss the k -coverage problem, that is, certain area is covered by k or more wireless sensor nodes at the same instant. In this case, synthesis detection reliability of the area is at least:

$$R = 1 - \prod_{i=1}^k (1 - r_i) \quad (5)$$

Therefore, we can acquire high synthesis detection reliability even though the detection reliability of individual wireless sensor node is limited.

2.2. Energy consumption model

During target tracking, wireless sensor nodes have the functions of data acquisition, processing and reporting. The related sensing, computation and communication operations will lead to energy depletion. Out of all the energy consumption sources in WSNs, wireless communication is the largest portion. Thereby, it is the main one taken into account here. As radio signal attenuation in the air is related with the propagation distance, we adopt the free space propagation model [13], which is expressed as:

$$L_p = \left(\frac{\lambda_s}{4\pi d_{i,j}} \right)^2 \quad (6)$$

where L_p is the path loss, λ_s is the wavelength of signal, and $d_{i,j}$ is the propagation distance. If radio signal propagates between wireless sensor node i and j , which are located at (x_i, y_i) and (x_j, y_j) , the corresponding propagation distance can be calculated as:

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (7)$$

Accordingly, a model of wireless communication is assumed to analyze energy consumption of communication. Here, the power consumption of data transmission between wireless sensor node i and j is calculated as [14]:

$$\Psi = \alpha_1 r_d + \alpha_2 d_{i,j}^2 r_d \quad (8)$$

where r_d denotes the data rate, α_1 denotes the electronics energy expended in transmitting one bit of data, $\alpha_2 > 0$ is a constant related to the radio energy. Given the transmission tasks through the network, the energy consumption feature of WSNs can be obtained.

3. Evaluation Metrics of Energy-efficient Coverage

To achieve reliable detection and energy conservation in target tracking application, WSNs should apply an energy-efficient coverage scheme. Coverage and energy performance is concerned in potential mobile node deployment. For specified detection reliability, sensing area needs to be covered by certain number of wireless sensor nodes. The area which can satisfy this reliability requirement in the whole sensing field can reflect the coverage performance. On the other hand, data packet transmission from wireless sensor nodes to the sink node results in energy consumption. An energy-efficient communication framework can be established by the lowest cost paths. This framework indicates the lowest energy consumption level which can be provided by different deployment of WSNs.

It is assumed that there are n stationary nodes and m mobile nodes in a $L \times L$ square sensing field. The coordinates of sink node are $(L/2, L/2)$. In a possible network deployment, the coordinates of all wireless sensor nodes (x_i, y_i) ($i=1, 2, \dots, n+m$) can be obtained, where the indices of the stationary and mobile nodes are $(i=1, 2, \dots, n)$ and $(i=n+1, n+2, \dots, n+m)$, respectively. Accordingly, coverage and energy metrics and related algorithms will be presented to evaluate the network deployment in this section.

3.1. Coverage metrics

Typically, certain detection reliability R^{req} is required for specific target tracking application. Based on Equation (5), the required number of wireless sensor nodes, which can detect the target with reliability r_0 at the same time, can be calculated as:

$$k^{req} = \log_{1-r_0} (1 - R^{req}) \quad (9)$$

The area which is covered by k^{req} or more wireless sensor nodes is regarded as the reliable detection area. To provide integrated and continuous detection of targets in the sensing field, the reliable detection area should be as large as possible. Therefore, we define the proportion of reliable detection area in the whole sensing field as the coverage metric.

As discussed in Section 2.1, each wireless sensor node covers a plate area with radius R_a . Due to the irregular network deployment, the coverage state problem is too complicated for geometric analysis. Thus, we exploit a numerical method, the grid exclusion algorithm, to extract the coverage state information. The pseudo-code for grid exclusion algorithm is outlined in Algorithm 1.

Algorithm 1

1. Initialization

Divide the square sensing field into $l \times l$ uniform grids. Each grid is a $(L/l) \times (L/l)$ square area. Simplify the grids into points, then each grid can be denoted by its centre point. The coordinates of points are:

$$\{(x^g, y^g) \mid x^g, y^g = L/2l, 3L/2l, \dots, (2l-1)L/2l\} \quad (10)$$

Initialize the coverage state matrix $\{\text{cov}(i, j)\}$:

$$\text{cov}(i, j) = 0 \quad i, j = 1, 2, \dots, l \quad (11)$$

Set the number of reliable detection point $num = 0$ and set the number of unreliable detection points $n_r = 0$.

2. Coverage state for stationary nodes on the whole sensing field

Check the detection reliability point by point.

For $x^g = L/2l, 3L/2l, \dots, (2l-1)L/2l$

For $y^g = L/2l, 3L/2l, \dots, (2l-1)L/2l$

This point is related to the element $\text{cov}(i^g, j^g)$ of the coverage state matrix:

$$i^g = x^g l / L + 1/2, \quad j^g = y^g l / L + 1/2 \quad (12)$$

Check whether stationary node i covers this point.

For $i = 1, 2, \dots, n$

Calculate the distance between stationary node i and the point:

$$d^g = \sqrt{(x_i - x^g)^2 + (y_i - y^g)^2} \quad (13)$$

Update the coverage state matrix as:

$$\text{cov}(i^g, j^g) = \begin{cases} \text{cov}(i^g, j^g) + 1 & d^g < Ra \\ \text{cov}(i^g, j^g) & d^g \geq Ra \end{cases} \quad (14)$$

End

If $\text{cov}(i^g, j^g) > k^{req}$

Update the number of reliable detection point:

$$num = num + 1 \quad (15)$$

Else

Record the unreliable detection point:

$$n_r = n_r + 1, \quad (x'_{n_r}, y'_{n_r}) = (x^g, y^g), \quad (i'_{n_r}, j'_{n_r}) = (i^g, j^g) \quad (16)$$

End

End

End

The coverage state matrix of stationary nodes is obtained.

3. Coverage state for mobile nodes on the unreliable detection area

Check the detection reliability excluding the reliable detection point.

For $j= 1, 2, \dots, n_r$

Check whether mobile node i covers this point.

For $i=n+1, n+2, \dots, n+m$

Calculate the distance d^g between (x'_j, y'_j) and (x_i, y_i) .

Update the coverage state matrix adopting Equation (14).

End

If $\text{cov}(i'_j, j'_j) > k^{req}$

Update the number of reliable detection points as Equation (15).

End

End

Finally, the coverage metric C can be calculated as:

$$C = \frac{\text{num}}{l \times l} \quad (17)$$

Instead of calculating the coverage state of all wireless sensor nodes at one time, the coverage state matrix of stationary nodes is first extracted in the grid exclusion algorithm. Excluding the reliable area, the coverage state of mobile nodes is then calculated on the remaining area. In this way, only Step 3 of the algorithm needs to be implemented repeatedly when a different deployment of mobile nodes is evaluated, thereby computational costs could be reduced. Moreover, only the recorded information of unreliable detection area is necessary for multiple wireless sensor nodes in distributed optimization algorithms, such as DPSOSA, to be covered in Section 4, so that the distributed optimization structure can be simplified and its communication costs will be low. Notice that granularity of computation is scalable by easily adjusting the division parameter l . The tradeoff can be made between computational cost and coverage evaluation

3.2. Energy metrics

During target tracking in WSNs, wireless sensor nodes spend significant energy reporting their observations. With the model presented in Section 2.2, we analyze the energy consumption of wireless communication.

According to Equation (8), the wireless sensor nodes which are far away from the sink node would spend too much energy when their data packets are transmitted directly. These nodes may find a number of other nodes for data forwarding and such a multi-hop manner will potentially conserve energy. Thus, the lowest cost path to sink node should be found for each wireless sensor node.

Here, Dijkstra's algorithm is introduced to solve this lowest path problem, which can accomplish breadth-first path search between one single destination vertex and all the other vertexes on the connected graph [15,16]. Since any vertex that has shorter path to the destination vertex is traversed, the optimal solution can always be found.

For any given WSN deployment, the sink node is regarded as the destination vertex and denoted by u_0 , while wireless sensor nodes are taken as all the other vertexes and denoted by $U = \{u_1, u_2, \dots, u_{n+m}\}$. The edge weight between vertex u_i and u_j is defined according to Equations (7) and (8):

$$\omega_{i,j} = \alpha_1 + \alpha_2 d_{i,j}^2 \quad i, j = 0, 1, 2, \dots, n+m, i \neq j \quad (18)$$

Then the pseudo-code for the lowest cost path search is outlined in Algorithm 2.

Algorithm 2

1. Initialization

Adopt variable D_i to represents estimate of the lowest cost from u_i to u_0 . It converges to the real value after iterations. Initialize the connected graph as:

$$D_0 = 0, D_i = \omega_{i,0} \quad i = 1, 2, \dots, n \quad (19)$$

The set of vertexes which have found the lowest cost paths is denoted by Q , set $Q = \emptyset$.

2. Iteration

While $Q \neq U$

Find the next vertex with the lowest cost path to u_0 .

For any vertex $u_i \notin Q$

If D_i satisfies:

$$D_i = \min_{k=1}^{n+m} D_k \quad (20)$$

The lowest cost path from vertex u_i to u_0 is found.

Update Q :

$$Q = Q \cup \{u_i\} \quad (21)$$

Record the vertex u_i , set $i_0 = i$.

End

End

For any vertex $u_j \notin Q$

Update D_j :

$$D_j = \min\{D_j, \omega_{j,i_0} + D_{i_0}\} \quad (22)$$

End

End

After iteration, D_i denotes transmission energy consumption per bit from vertex u_i to u_0 adopting the lowest cost path, where $i = 1, 2, \dots, n+m$.

Thus, the lowest cost paths from all wireless sensor nodes to sink node are available, which form an energy-efficient communication framework. This framework reflects the lowest energy consumption

level which can be provide by the given network deployment. Since each wireless sensor node has the opportunity to detect a target and report its data, we can evaluate the energy consumption by the total cost of all the reporting paths. Therefore, the energy metric E of network deployment is calculated as:

$$E = \sum_{i=1}^{n+m} D_i \quad (23)$$

Generally, different deployment of mobile nodes corresponds to different energy metric values. The proposed coverage and energy metric will be used to evaluate different network deployment in the optimization algorithm.

4. Distributed Optimization Algorithm for Energy-efficient Coverage

When WSNs are initially organized, proper deployment of mobile nodes is desirable to achieve energy-efficient coverage. Also, the environment may cause changes in WSNs, such as the appearance of node failures. Therefore, position adjusting of mobile nodes is necessary for resource re-allocation. With the proposed coverage and energy metrics, deployment optimization should be implemented to provide adaptability for WSNs in these cases. Then, the optimization results are broadcasted over the network so that WSNs can be self-organized.

Following the previous assumption, there are n stationary nodes and m mobile nodes available in the deployment problem. The coordinates of mobile nodes are taken as non-integral input vectors for optimization. As described in Section 3.1, certain coverage ratio C_0 , namely the optimized coverage metric, is demanded under the detection reliability requirement. Thus, the objective of optimization is to decrease the energy consumption level of WSNs in target tracking applications under the condition that the required coverage metric is satisfied.

Kennedy *et al.* developed particle swarm optimization in 1995 based on the analogy of swarms of birds and fish schools. PSO is an efficient optimization tool for solving combinatorial optimization and dynamic optimization problems in multi-dimensional space, which implements fast convergence and good robustness [17]. Here, it is considered as a deployment optimization algorithm in WSNs. Like other evolutionary algorithms, PSO uses a fitness function as criterion to evolve the behavior of the solution population. In the algorithm, potential solutions, namely particles, fly through the search space. Each particle keeps track of the best position it has achieved so far, which represents a particle experiment. Another kind of experiment is the best position which has been achieved by the companion of particle so far. The particle velocity is constantly adjusted according to the two kinds of experiences.

PSO has a strong ability for finding the most optimal result. However, it has a disadvantage in local minima. Thus, simulated annealing [18] which has a strong ability for finding the local optimal result is introduced to avoid the problem of local minima. SA mainly consists of the repeating of two steps: a generation mechanism and an acceptance criterion. It starts off at an initial random state with a high temperature, and then a sequence of iterations is generated. A perturbation mechanism transforms the current state into a next state selected from the neighborhood of the current state. If this neighboring state has better fitness, the neighboring state is accepted as the current state. If this neighboring state has worse fitness, the neighboring state is accepted with a certain probability determined by the

acceptance criterion [19]. After sufficient times of acceptance, the temperature is decreased. This process is repeated until the final temperature is reached.

We propose distributed particle swarm optimization and simulated annealing here. SA is applied on the global best position of PSO. Then the vicinity of the global best position is searched to obtain a local optimal result. Thereby, the procedure of PSO is corrected by the result. In the same way, the best position achieved by individual particle can be corrected by SA. Since SA maintains only one solution, this extended optimization tasks can be assigned simply to a number of wireless sensor nodes utilizing the distributed computing capacity of WSNs. The pseudo-code for DPSOSA is outlined in Algorithm 3.

Algorithm 3

The sink node performs main part of the algorithm.

1. Initialization

The population of particles is set as pop .

For $i = 1, 2, \dots, pop$

For particle i , X_i represents the current position, where the elements present the coordinates of all mobile nodes:

$$X_i = (x_{n+1}, y_{n+1}, x_{n+2}, y_{n+2}, \dots, x_{n+m}, y_{n+m}) = \{x_j^i \mid j = 1, 2, \dots, 2m\} \quad (24)$$

V_i represents the current velocity it has achieved so far:

$$V_i = \{v_j^i \mid j = 1, 2, \dots, 2m\} \quad (25)$$

P_i represents the best position it has achieved so far:

$$P_i = \{p_j^i \mid j = 1, 2, \dots, 2m\} \quad (26)$$

Initialize X_i as a random position $X_i(1)$ in the search space.

Initialize V_i as a random velocity $V_i(1)$.

Set the initial P_i as:

$$P_i(1) = X_i(1) \quad (27)$$

End

According to the purpose of energy-efficient coverage in WSN, the minimization objective function $f(X)$ is defined for the position X of any given particle as:

$$f(X) = \begin{cases} \rho E_0 - C & C < C_0 \\ \rho E - 1 & C \geq C_0 \end{cases} \quad (28)$$

where E and C are the metrics defined in Section 3.1 and 3.2 respectively. E_0 is a constant which denotes the upper bound of energy metric, while C_0 is the demanded coverage ratio. ρ is a constant for balancing the two metrics.

2. PSO iterations

For $t = 1, 2, \dots, PSO_ITER$

The global best position of particle is calculated as:

$$P_g(t) = \min\{f[P_1(t)], f[P_2(t)], \dots, f[P_{pop}(t)]\} = \{p_j^g \mid j = 1, 2, \dots, 2m\} \quad (29)$$

For $i = 1, 2, \dots, pop$

The velocity of particle is updated as:

$$v_j^i(t+1) = \eta(t)v_j^i(t) + c_1r_j^1[p_j^i(t) - x_j^i(t)] + c_2r_j^2[p_j^g(t) - x_j^i(t)] \quad (30)$$

$\Gamma_1 = \{r_j^1\}$ and $\Gamma_2 = \{r_j^2\}$ are two separate random sequences, where $j = 1, 2, \dots, 2m$, c_1 and c_2 are acceleration constants, representing the weight of acceleration terms that pull each particle toward the local best position and global best position and $\eta(t)$ is the inertia weight for balancing the global and local search ability. It is defined as:

$$\eta(t) = 0.9 - \frac{t}{PSO_ITER} \times 0.5 \quad (31)$$

The position of particle is updated as:

$$x_j^i(t+1) = x_j^i(t) + v_j^i(t+1) \quad (32)$$

The best position of particle is calculated as:

$$P_i(t+1) = \begin{cases} P_i(t) & f[X_i(t+1)] \geq f[P_i(t)] \\ X_i(t+1) & f[X_i(t+1)] < f[P_i(t)] \end{cases} \quad (33)$$

End

The sink node sorts $P_i(t+1)$ by their fitness. Select the best SA_NUM positions $\{P_i^s \mid i = 1, 2, \dots, SA_NUM\}$, which are to be optimized with SA. The optimization of global best position will be performed by the sink node, while $SA_NUM - 1$ wireless sensor nodes are randomly selected to optimize the other positions.

The sink node transmits each particle to the related node. Then sink node and these nodes perform parallel SA optimization.

For $i = 1, 2, \dots, SA_NUM$

Perform SA iterations taking the initial state as:

$$A = P_i^s = \{a_j \mid j = 1, 2, \dots, 2m\} \quad (34)$$

Set an initial temperature T .

For $k = 1, 2, \dots, SA_ITER$

The cooling condition is that the best state remains unchanged for K times.

While the cooling condition is not satisfied

Use a perturbation mechanism to generate a new state A' :

$$A' = A + rand_norm \cdot \Delta \quad (35)$$

where $rand_norm$ is a normally distributed random number, $\Delta = \{\delta_j | j = 1, 2, \dots, 2m\}$ and $\Delta = \{\delta_j | j = 1, 2, \dots, 2m\}$ and δ_j is defined with a random integer j_0 in $[1, 2m]$:

$$\delta_j = \begin{cases} 1 & j = j_0 \\ 0 & j \neq j_0 \end{cases} \quad (36)$$

The decrease of fitness is:

$$df = f(A') - f(A) \quad (37)$$

Check whether the new state should be accepted according to Metropolis criteria.

If $df < 0$

Accept the new state: $A = A'$.

Else if $e^{-df/(\gamma T)} > rand$, where γ is Boltzmann constant and $rand$ is a random number in $[0,1]$

Accept the new state: $A = A'$.

Else

The new state A' cannot be accepted.

End

End

Cool down with a parameter λ :

$$T = \lambda T \quad (38)$$

End

End

If any result is better the initial state, the wireless sensor node sends it back to the sink node, where the former position will be replaced.

End

Finally, the global best position presents the optimized deployment of WSN.

In PSOSA, the sink node performs PSO_ITER iterations of PSO, where the inertia weight $\eta(t)$ linearly decreases through the course of the run. A large inertia weight facilitates a global search while a small inertia weight facilitates a local search. Accordingly, the optimization process can converge to the neighborhood of the global optimal solution smoothly at the prophase and converge to the global optimal solution quickly at the anaphase. SA_NUM local best positions are optimized by SA on the sink node and $SA_NUM - 1$ other wireless sensor nodes. After SA_ITER iterations of SA, the optimized results are utilized to correct the former positions. In this way, the algorithms have good potential to obtain the optimal deployment of WSNs.

5. Simulation Experiments

In this section, we will analyze the efficiency of DPSOSA algorithm with simulation experiments. The simulation environment will be specified. Then the simulation and comparison of algorithms will be given. Finally, the network simulations will be present for target tracking application.

5.1. Simulation environment

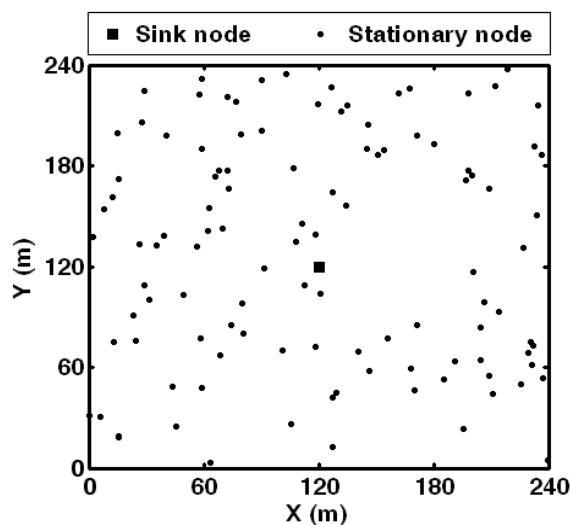
The fundamental parameters of WSN are presented in Table 1. The stationary nodes are placed randomly in the square sensing field as shown in Figure 2(a). With the specified sensing radius of wireless sensor node, we can calculate the initial coverage state of stationary nodes according to Section 3.1. Figure 2(b) shows the initial coverage state, and the area with darker grey levels means that there is coverage by more nodes.

Table 1. Fundamental parameters of WSN.

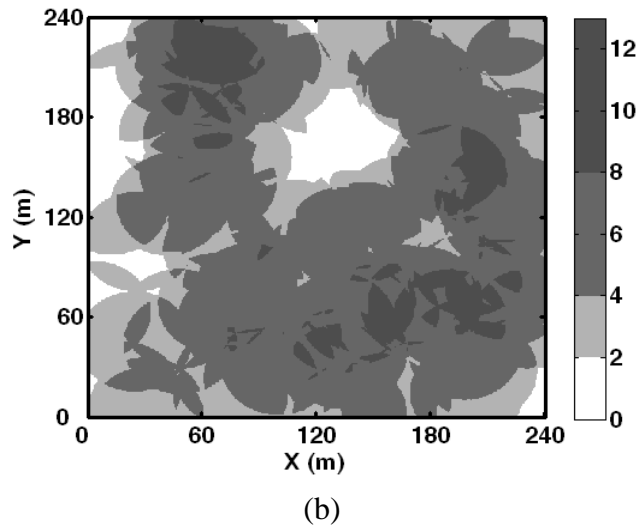
Parameter	Value
Sensing field dimension $L \times L$	$240m \times 240m$
Stationary node number n	108
Mobile node number m	20
Sink node coordinates	(120,120)
Sensing radius R_a	30m
Sensor reliability r_0	0.6

In the energy consumption model, we set $\alpha_1 = 50nJ/bit$ and $\alpha_2 = 100pJ/bit/m^2$. When we calculate the coverage metric, the sensing field is divided into 100×100 uniform grids. Accordingly, the initial coverage metric of k -coverage area is presented in Table 2, where k changes from 1 to 6. As more covering nodes are required to satisfy the reliability, the initial coverage metric turns lower rapidly. Based on Equation (5), Figure 3 illustrates the detection reliability with different covering node number. The detection reliability grows exponentially and exceeds 0.99 when the covering node number is 6.

Figure 2. Initial deployment and coverage state of WSN: (a) Placement of sink node and stationary nodes; (b) Coverage state of stationary nodes in the sensing field.



(a)

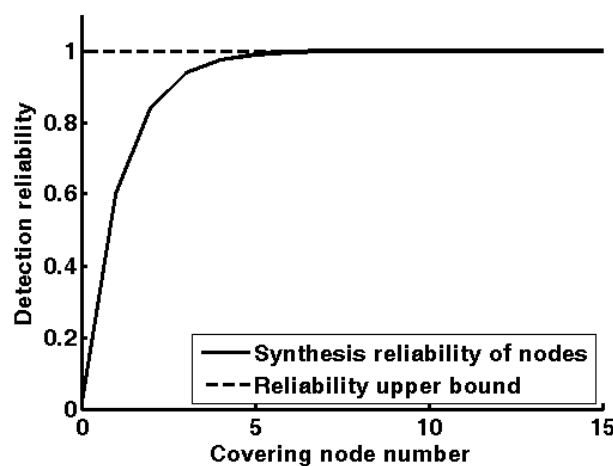


In the DPSOSA algorithm, the particle number pop is set as 30, the acceleration constants $c_1=c_2=1$, and the PSO iteration number PSO_ITER is specified as 40. During each SA optimization, we set initial temperature T as 0.0001 according to the fitness function. Parameter K is 4 in the cooling condition, while the cooling parameter λ is 0.6. Besides, Boltzmann constant $\gamma = 1$, and the PSO iteration number SA_ITER is specified as 5.

Table 2. The initial coverage metric of k-coverage area.

k	1	2	3	4	5	6
Coverage metric (%)	98.00	91.45	83.17	70.80	54.03	37.25

Figure 3. Detection reliability as a function of covering node number.



Target tracking application of the optimized WSN will be simulated on a modeling platform, Opnet Modeler, which is developed for communication network and distribution system. It is assumed that the sampling period of WSN is 0.5 s. Without loss of generality, a mobile target moves randomly in the sensing area for 120 s. Wireless channel model is bpsk, the free space propagation model is utilized and data rate is 1 Mbps.

5.2. Simulations of deployment optimization

With the stated simulation environment, the DPSOSA algorithm can be adopted to achieve energy-efficient coverage. First, we should define the coverage requirement, which is given by two parameters, detection reliability R^{req} and coverage ratio C_0 . Considering the initial coverage state, we discuss two kind of coverage requirement to analyze the performance of algorithm against different conditions: (1) $R^{req} = 0.8$, $C_0 = 95\%$; (2) $R^{req} = 0.9$, $C_0 = 95\%$. The required covering node numbers k^{req} are 2 and 3 respectively. According to Table 2, the latter coverage requirement is much stricter than the former one.

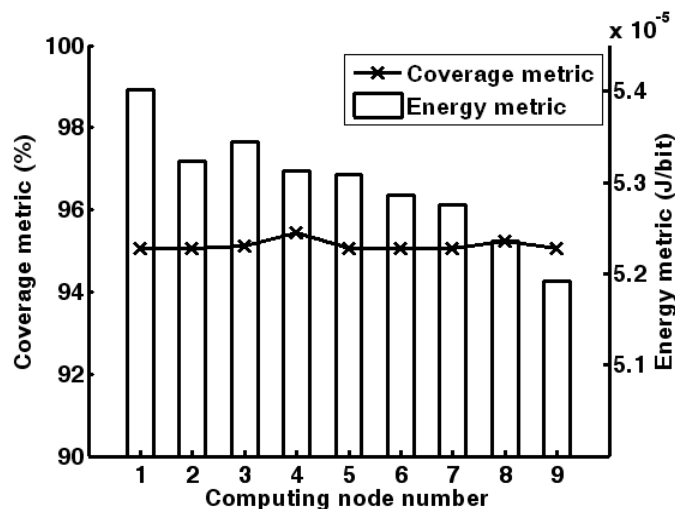
Second, the constant E_0 which denotes the upper bound of energy metric should be specified for the fitness calculation. Here, we search the lowest cost paths of the station nodes without any mobile node. Assume the related path cost is $\{D_i^s \mid i = 1, 2, \dots, n\}$, then E_0 is defined as:

$$E_0 = \max_{i=1}^n (D_i^s) \cdot m + \sum_{i=1}^n D_i^s \quad (39)$$

In this case, E_0 is 7.26×10^{-5} J/bit, and ρ is set as 10^5 .

Then, we implement DPSOSA to optimize the deployment of WSN with a different computing node number SA_NUM , which varies from 1 to 9. Specially, the algorithm is accomplished by the sink node when SA_NUM is set as 1. Since each wireless sensor node has little information to exchange with the sink node and the computing node number is limited during DPSOSA, its communication cost can be ignored. As shown in Figure 4, the optimization results of DPSOSA are obtained under the two kinds of coverage requirement. We can find that all the optimized coverage metrics exceed 95%, while the energy metric trends to be lower as the computing node number becomes larger. Hence, the performance of DPSOSA benefits from the computation capacity of multiple wireless sensor nodes.

Figure 4. Optimization results of DPSOSA utilizing different computing node numbers under two kinds of coverage requirement: (a) $R^{req} = 0.8$, $C_0 = 95\%$; (b) $R^{req} = 0.9$, $C_0 = 95\%$.



(a)

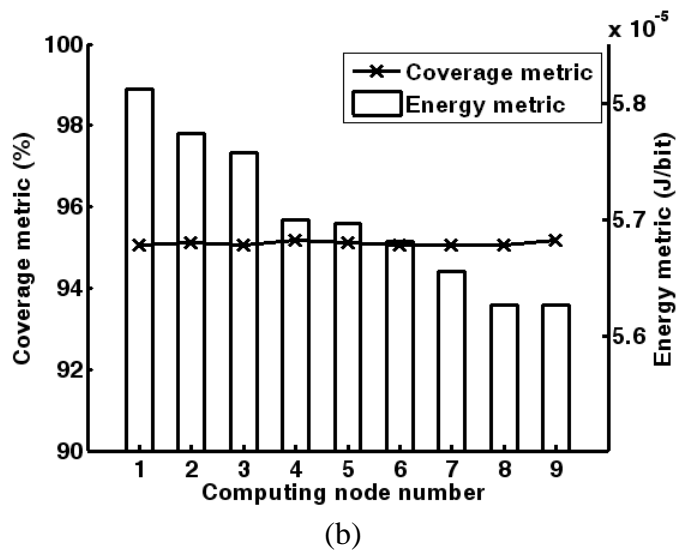
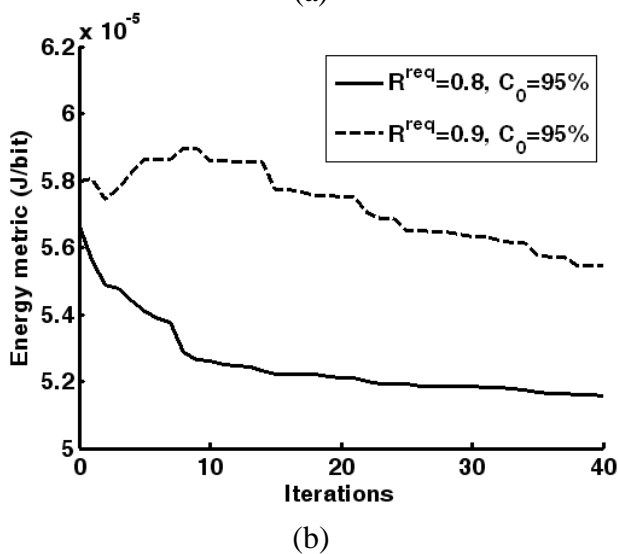
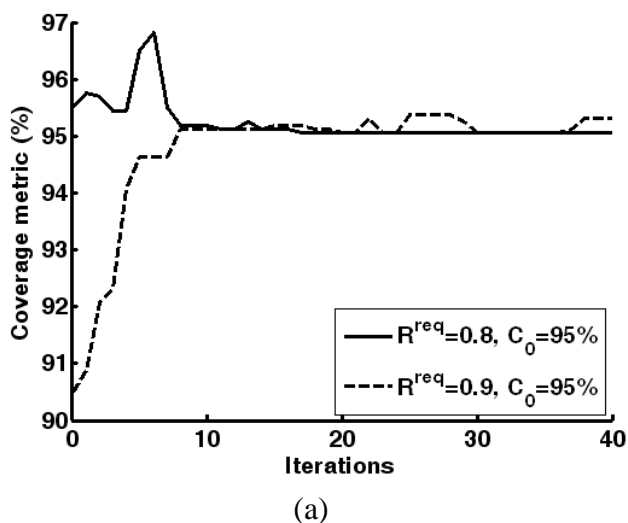
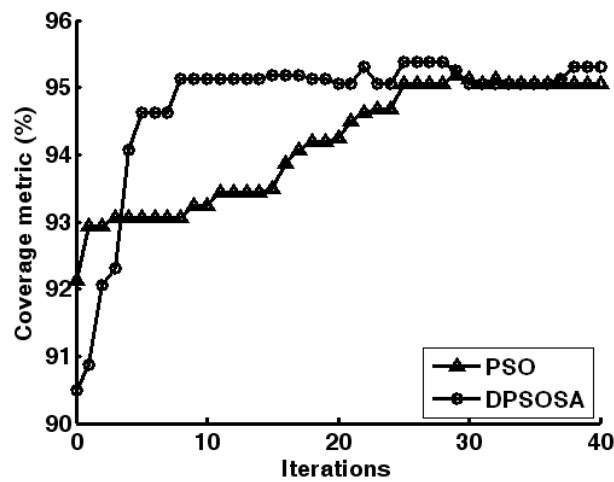


Figure 5. Convergence curves of the metrics under two kinds of coverage requirement during DPSOSA: (a) Coverage metric; (b) Energy metric.

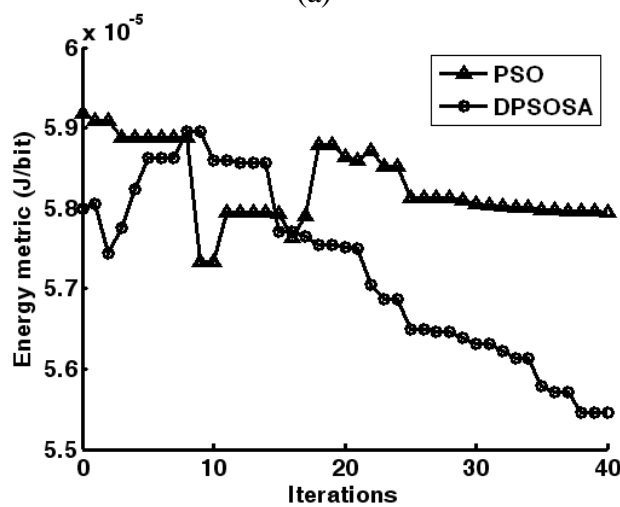


To obtain ideal optimization results, the computing node number is fixed as 9 in the following discussion. Then, Figure 5 shows the convergence curves of the metrics under the two kinds of coverage requirement. Rather than the coverage requirement that $R^{req} = 0.8$ and $C_0 = 95\%$, it is more difficult to achieve the coverage requirement that $R^{req} = 0.9$ and $C_0 = 95\%$. Therefore, the former coverage requirement is satisfied at the beginning, while the latter one is satisfied after 8 iterations in the optimization procedure, which is shown in Figure 5(a). In Figure 5(b), the algorithm can make more effort to achieve improved energy metric with the former coverage accordingly. Meanwhile, the former coverage requirement provides more adjustability for mobile node deployment to achieve lower energy metric.

Figure 6. Convergence curves of the metrics during DPSOSA and PSO under the coverage requirement that $R^{req} = 0.9$ and $C_0 = 95\%$: (a) Coverage metric; (b) Energy metric.



(a)



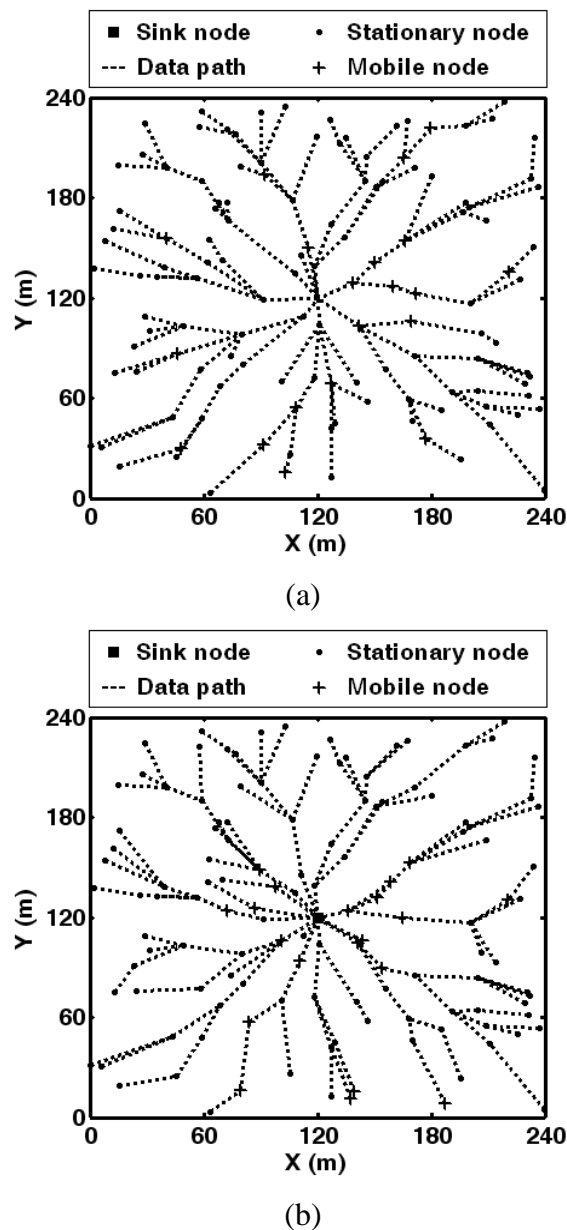
(b)

Furthermore, we will compare the performance of DPSOSA and general PSO algorithms. Here, only the coverage requirement that $R^{req} = 0.9$ and $C_0 = 95\%$ is considered. The same scenario and fitness function is employed in PSO. In Figure 6, the convergence curves of coverage and energy

metrics are presented during DPSOSA and PSO. From Figure 6(a), we can find that PSO spends much more iterations than DPSOSA to satisfy the coverage requirement though it has a better initial coverage of mobile nodes. And the energy metric is significantly improved by DPSOSA compared to the optimization result of PSO as shown in Figure 6(b).

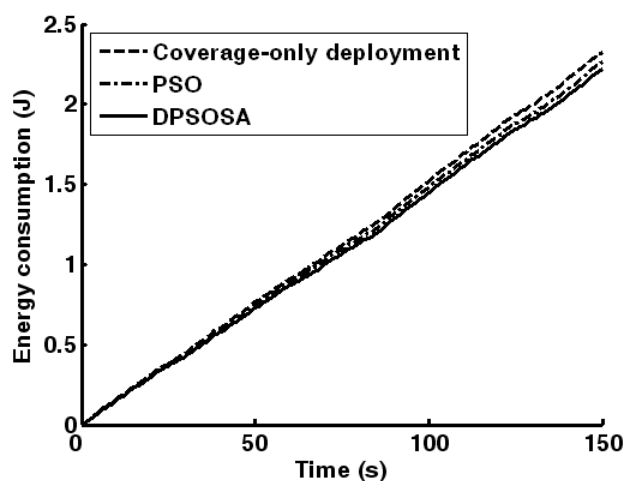
According to the optimization results of PSO and DPSOSA, we can obtain optimized deployment and communication paths of WSN as shown in Figure 7. The coverage ratio of the WSN in Figure 7(a) and (b) is 95.13% and 95.31%, respectively. It can be seen that the data paths obtained by DPSOSA tend to provide more potential for multi-hop communication instead of using longer distance data transmission, although both algorithms attempt to achieve energy efficiency. As a result, the energy metrics obtained by PSO and DPSOSA are $5.83 \times 10^{-5} \text{ J/bit}$ and $5.57 \times 10^{-5} \text{ J/bit}$, respectively.

Figure 7. Optimized WSN deployment and communication paths adopting two algorithms: (a) PSO; (b) DPSOSA.



Finally, scenarios of WSN are set up according to Figure 7(a) and (b) for target tracking simulations. Besides, we discuss a coverage-only deployment, which is optimized by DPSOSA taking only coverage metric into account. In each sensing period, the closest wireless sensor node to the mobile target is chosen via negotiation. It then acquires information and sends a 2KB data packet to the sink node along the optimized path. The total energy consumption over time is extracted from the simulations, as shown in Figure 8. We find that the WSN optimized by DPSOSA has a lower energy consumption than the one optimized by PSO. Moreover, target tracking is a long term task, so more energy could be saved during the lifetime of WSNs. Compared to the coverage-only deployment, DPSOSA achieves an energy conservation of 4.68%.

Figure 8. Energy consumption comparison of WSNs optimized by PSO and DPSOSA in target tracking application



From the experiments, the efficiency of multiple computing nodes is verified and it is shown that DPSOSA can be applied under different coverage requirements. Then, the improved energy efficiency of DPSOSA is demonstrated by algorithm simulations and target tracking application compared with general PSO.

6. Conclusions

Focusing on the energy-efficient coverage problem of WSNs, this paper has proposed distributed particle swarm optimization and simulated annealing to optimize the network deployment. In a network composed of stationary and mobile wireless sensor nodes, the proper placement of mobile nodes is discussed, considering sensing coverage and energy consumption. Then, the coverage metric is defined utilizing a grid exclusion algorithm, while the energy metric is calculated by Dijkstra's algorithm, which provides the optimal communication paths for data reporting. Particle swarm optimization and simulated annealing are combined to find the global optimal solution, where the fitness function is designed to minimize the energy metric guaranteeing specified coverage ratio. Besides, computation capability of multiple wireless sensor nodes is adopted to enhance the optimization capacity. Experimental results represent that significant energy conservation can be

achieved by the proposed optimization algorithm compared to general PSO, and energy efficiency of WSN is boosted up in target tracking application. This paper presents an evaluation method for energy-efficiency of coverage problem in WSNs. The application-oriented property is realized by target tracking. Still, further investigation should be made on adaptive routing schemes and scalable network topologic.

Acknowledgements

This paper is supported by the National Grand Fundamental Research 973 Program of China under Grant No.2006CB303000 and the National Natural Science Foundation of China (No.60673176; No.60373014; No.50175056).

References and Notes

1. Huang, C.F.; Tseng Y.C. A survey of solutions to the coverage problems in wireless sensor networks. *Journal of Internet Technology* **2005**, *6*, 1-8.
2. Dhillon, S.S.; Chakrabarty, K. Sensor placement for effective coverage and surveillance in distributed sensor networks. *IEEE Wireless Communications and Networking Record* **2003**, 1609-1614.
3. Meguerdichian, S.; Koushanfar, F. Coverage problems in wireless ad-hoc sensor networks. *Proceedings of the IEEE INFOCOM* **2001**, 1380-1387.
4. Wang, X.; Xing, G. Integrated coverage and connectivity configuration in wireless sensor networks. *Proceedings of ACM International Conference on Embedded Networked Sensor Systems* **2003**, 28-39.
5. Bai, X.; Kumar, S. Deploying wireless sensors to achieve both coverage and connectivity. *Proceedings of ACM MobiHoc'06* **2006**, 131-142.
6. Wang, X.; Ma, J.; Wang, S.; Bi, D. Prediction-based dynamic energy management in wireless sensor networks. *Sensors* **2007**, *7*, 251-266.
7. Wang, X.; Wang, S. Collaborative signal processing for target tracking in distributed wireless sensor networks. *Journal of Parallel and Distributed Computing* **2007**, *67*, 501-515.
8. Wang, X.; Wang, S.; Ma, J. An improved co-evolutionary particle swarm optimization for wireless sensor networks with dynamic deployment. *Sensors* **2007**, *7*, 354-370.
9. Hu, F.; Cao, X.; May, C. Optimized scheduling for data aggregation in wireless sensor networks. *Proceedings of IEEE International Conference on Networking, Sensing and Control* **2005**, *2*, 557-561.
10. Wang, X.; Wang, S.; Ma, J. An improved particle filter for target tracking in sensor system. *Sensors* **2007**, *7*, 144-156.
11. Sadler, B.M. Local and broadcast clock synchronization in a sensor node. *IEEE Signal Processing Letters* **2006**, *13*, 9-12.

12. Dutta, P.K.; Arora, A.K. Towards radar-enabled sensor networks information processing in sensor networks. *Proceedings of IEEE International Conference on Information Processing in Sensor Networks* **2006**, 467-474.
13. Yao, Q; Tan, S.K.; Ge, Y. An area localization scheme for large wireless sensor networks. *Proceedings of IEEE Vehicular Technology Conference* **2005**, 2835-2839.
14. Chhetri, A.S.; Morrell, D. Energy efficient target tracking in a sensor network using non-myopic sensor scheduling. *Proceedings of International Conference on Information Fusion* **2005**, 558-565.
15. Sobrinho. Algebra and algorithms for QoS path computation and hop-by-hop routing in the Internet. *IEEE/ACM Transactions on Networking* **2002**, *10*, 541-550.
16. Sudhakar, T.D. Supply restoration in distribution networks using Dijkstra's algorithm. *Proceedings of IEEE International Conference on Power System Technology* **2004**, *1*, 640-645.
17. Ciuprina, G.; Ioan, D.; Munteanu, I. Use of intelligent-particle swarm optimization in electromagnetics. *IEEE Transaction on Magnetics* **2002**, *38*, 1037-1040.
18. Miki, M.; Hiroyasu, T. Parallel simulated annealing with adaptive temperature determined by genetic algorithm. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics* **2002**, *3*, 1-6.
19. Calaor, A.E.; Hermosilla, A.Y. Parallel hybrid adventures with simulated annealing and genetic algorithms. *Proceedings of International Symposium on Parallel Architectures, Algorithms and Networks*, **2002**, 33-38.