Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm[☆]

Jie Jia[a,*], Jian Chen[a], Guiran Chang[a,b], Zhenhua Tan[a,c]

[a] School of Information Science and Engineering, Northeastern University, Shenyang, 110004, China
[b] Computing Center, Northeastern University, Shenyang, 110004, China
[c] School of Software, Northeastern University, Shenyang, 110004, China

ARTICLE INFO

Keywords:
Coverage control
Energy
Multi-objective genetic algorithm
Wireless sensor network

ABSTRACT

Due to the constrained energy and computational resources available to sensor nodes, the number of nodes deployed to cover the whole monitored area completely is often higher than if a deterministic procedure were used. Activating only the necessary number of sensor nodes at any particular moment is an efficient way to save the overall energy of the system. A novel coverage control scheme based on multi-objective genetic algorithm is proposed in this paper. The minimum number of sensors is selected in a densely deployed environment while preserving full coverage. As opposed to the binary detection sensor model in the previous work, a more precise detection model is applied in combination with the coverage control scheme. Simulation results show that our algorithm can achieve balanced performance on different types of detection sensor models while maintaining high coverage rate. With the same number of deployed sensors, our scheme compares favorably with the existing schemes.

1. Introduction

The wireless sensor network (WSN) has emerged as a promising tool for monitoring the physical world, utilizing self-organizing networks of battery-powered wireless sensors that can sense, process and communicate. It can be deployed rapidly and cheaply, thereby enabling large-scale, on-demand monitoring and tracking over a wide range of applications such as danger alarm, vehicle tracking, battle field surveillance, habitat monitoring, etc [1,2].

Sensor nodes in such a network usually have limited onboard processing and wireless communication capabilities, and are equipped with batteries with limited power. Moreover, it is impractical or infeasible to replenish energy via replacing batteries on these devices in most applications. For example, habitat monitoring may require continuous operation for months and monitoring civil structures requires an operational lifetime of several years. As a result, it is well accepted that a sensor network should be deployed with high density (up to 20 nodes/m3 [3]) in order to prolong the network lifetime. However, if all the sensor nodes simultaneously operated, redundant sensing data, corresponding wireless communication collision and interference will cause much energy to be wasted. How does one cover all the sensing area with the least active nodes so that no blind-point exists and connectivity kept is significant? Coverage becomes a serious problem in large scale sensor networks where hundreds and thousands of nodes are randomly deployed.

[☆] The project was supported by the National High Technology Research and Development Program of China (Grant nos. 2006AA01Z413), and the National Natural Science Foundation of China (Grant nos. 60602061).

[*] Corresponding address: # 402 College of Information Science and Engineering, Northeastern University, 11 No. 3, Lane Wenhua Blvd. Heping District Shenyang, 110004 Liaoning Province, China. Tel.: +86 13604056128; fax: +86 04122680059.

E-mail addresses: jiajieneu@163.com (J. Jia), chen.jian@neusoft.com (J. Chen).

0898-1221/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2008.10.036

The coverage problem is one of the most fundamental issues in wireless sensor networks, which directly affects the capability and efficiency of the sensor network. Generally, it can be considered as the measure of QoS in a sensor network [4]. Current solutions are based for the most part on node scheduling, the main idea of which is to find the optimal number of active nodes while maintaining coverage and connectivity. The problem in finding the maximal number of covers in a sensor network is addressed in [5], where a cover is defined as a set of nodes that can completely cover the monitored area, and a centralized solution to this problem is proposed. What is more, they prove the np-completeness of this problem. Network coverage is closely related to network energy consumption. In the S-MAC scheme [6], energy consumption is reduced by allowing randomly selected sensors to go into the sleep mode. The sleeping sensors wake up periodically to retrieve the stored packets from their neighboring nodes. A distributed probing-based density control mechanism for robust sensing coverage named PEAS is proposed in [7]. A set of nodes are made active to maintain coverage while others are put into sleep mode to conserve energy. By adjusting the probing range of nodes, different coverage redundancy can be achieved. Although the algorithm guarantees that the distance between any pair of working nodes is at least the probing range, it does not completely preserve the original sensing coverage after turning off some nodes. Both [8] and [9] apply linear programming techniques to select the minimal set of active nodes to maintain coverage. Although achieving energy conservation by scheduling nodes to sleep is not a new approach, none of the existing algorithms satisfy the complete set of requirements in wireless sensor networks.

Several algorithms aim to find a close-to-optimal solution based on local information. In [10], an algorithm is proposed to turn off nodes based on the necessity for neighbor connectivity. They intend to reduce the system energy consumption without significantly diminishing the connectivity of the network. An algorithm, called GAF (Geographical Adaptive Fidelity) was proposed in [11], which uses geographic location information to divide the area into fixed square grids. Within each grid, it keeps only one node staying awake to forward packets. Both the two node-scheduling schemes above turn off nodes from a communication perspective without considering sensing coverage. While unusual events could happen at any time at any place, the main role of each node is sensing in wireless sensor networks. Therefore, if we only turn off nodes which are not participating in data forwarding, certain areas in the deploying area may become blind points [12]. In fact, the sensor network should remain connected so that the information collected by sensor nodes can be relayed back to data sinks or controllers. An important result was proved in [13], which states that if the communication range R_c is at least twice the sensing range R_s, a complete coverage of a convex area implies connectivity of the working nodes. Based on these results, the authors proposed a distributed, localized algorithm, called optimal geographical density control (OGDC). OGDC assumes that the sensor density is so high that a sensor can be found at any desirable point. However, it seems unrealistic in practice. Wang et al. proposed the coverage configuration protocol (CCP) [14] that can dynamically configure the network to provide different coverage degrees requested by applications. To facilitate the computation of intersection points, every node maintains a table with neighbor information and periodically broadcasts a hello beacon with its current location and status. For the case when ($R_c < 2R_s$), CCP does not guarantee network connectivity.

In order to extend the network lifetime by saving energy while meeting the coverage requirement, a novel searching algorithm, ECCA (Energy-efficient Coverage Control Algorithm), inspired by the multi-objective genetic algorithms (MOGAs [15]), is proposed in this paper to minimize the number of working nodes while maintaining full coverage. The sensor field is represented by a two-dimensional grid, and the dimensions of the grid provide a measure of the sensor field. The granularity of the grid, i.e. distance between grid points can be adjusted to trade off computation time of ECCA with the effectiveness of the coverage measure. Detection by each sensor is considered as a circle on the two-dimensional grid. The center of the circle denotes the sensor while the radius denotes the sensor's detection range. We first consider a binary detection model in which a target is detected (or not detected) with complete certainty by the sensor if a target is inside (or outside) its circle. Then we introduce the realistic probabilistic model in which the probability of the sensor detecting a target depends on the relative position of the target.

The remainder of the paper is organized as follows. In the next section, preliminary work of the coverage model in WSN is discussed. Then in Section 3, an energy-efficient coverage control algorithm named ECCA is proposed to compute the optimal sensor set. Section 4 shows the simulation results to validate our analysis. Finally, Section 5 describes the conclusions and future work.

2. Preliminary work and problem formulation

2.1. Preliminary assumptions

As the number of sensors deployed in the target area is greater than the optimum needed to perform full coverage, an important energy-efficient method coverage solution must be devised. The random deployment of sensors may lead to redundant nodes that share the same area. We consider that a sensor node radio can go to the sleep mode when it is not scheduled to perform the sensing task. A central data collector node, which is referred to as the base station (BS), can be the cluster head in a more general cluster-based framework. Sensed data might be processed locally by the sensors or the BS, from where it is aggregated and forwarded to the user. For the cluster-based sensor network architecture, the following assumptions are made:

- First, all the sensors have location determination capabilities (e.g. GPS) and they are able to communicate with the cluster head and send their location information to the BS or cluster head.

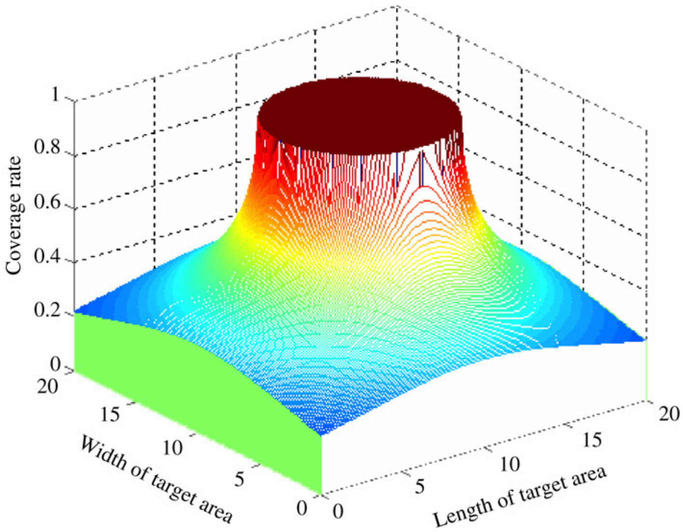


Fig. 1. Illustration of probabilistic sensor detection model where $r = 10$, $r_e = 5$, $\beta = 0.5$ and $\lambda = 0.5$.

- Second, every sensor within a cluster operates independently in their sensing activities and schedules itself for active/sleep intervals.
- Third, the cluster head is responsible for executing ECCA algorithm and broadcasting the schedule.

In this paper, we are concerned with designing the node scheduling mechanism, and do not address the problem of which protocol is used for data gathering or node synchronization. To efficiently transmit data from the sensors to the BS, a mechanism like LEACH [16] or PEGASIS [17] can be used. For node synchronization, one method is to make the BS send short beacons periodically.

2.2. Sensor coverage model in wireless sensor network

Assume the network is an $m \times n$ sensor field and k sensors are deployed in the random deployment stage. Each sensor has a detection range r, and sensor s_i is deployed at point (x_i, y_i). For any point P at (x, y), we denote the euclidean distance between s_i and P as $d(s_i, P)$.

$$d(s_i, P) = (x_i - x)^2 + (y_i - y)^2. \tag{2.1}$$

The binary sensor model shown in [16,17] expresses the coverage $c_{xy}(s_i)$ of a grid point P by sensor s_i.

$$C_{xy}(S_i) = \begin{cases} 1 & \text{if } d(s_i, p) < r_s \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

It assumes that sensor readings have no associated uncertainty in reality, and sensor detections are imprecise. The coverage $c_{xy}(s_i)$ needs to be expressed in probabilistic terms, hence a precise detection model is introduced:

$$C_{xy}(S_i) = \begin{cases} 0, & \text{if } r + r_e \leqslant d(s_i, P) \\ \mathrm{e}^{-\lambda a^{\beta}}, & \text{if } r - r_e < d(s_i, P) < r + r_e \\ 1, & \text{if } r - r_e \geqslant d(s_i, P) \end{cases}$$
$$\alpha = d(s_i, P) - (r - r_e) \tag{2.3}$$

where r_e ($r_e < r$) is a measure of uncertainty in the sensor detection, α and β are parameters that measure detection probability when a target is at distance greater than r_e but within maximum from the sensor. Note that distances are measured in units of grid points. This model reflects the behavior of range sensing devices such as infrared and ultrasound sensors. Different values of the parameters α and β yield different translations reflected by different detection probabilities, which can be viewed as the characteristics of various types of physical sensors. When $r = 10$, $r_e = 5$, $\beta = 0.5$ and $\lambda = 0.5$, the probabilistic detection model is shown in Fig. 1.

If $r_e \approx 0$, we use the binary sensor detection model, attempting to prevent the detection regions of two sensors from overlapping, thereby minimizing wasted overlap and covering a large grid with a small number of sensors. This optimal distribution of sensor nodes is illustrated in Fig. 2.

If $r_e > 0$, r_e is not negligible and the probabilistic sensor detection model is used. Due to the uncertainty in sensor detection responses, grid points are not uniformly covered with the same probability. Some grid points will have low

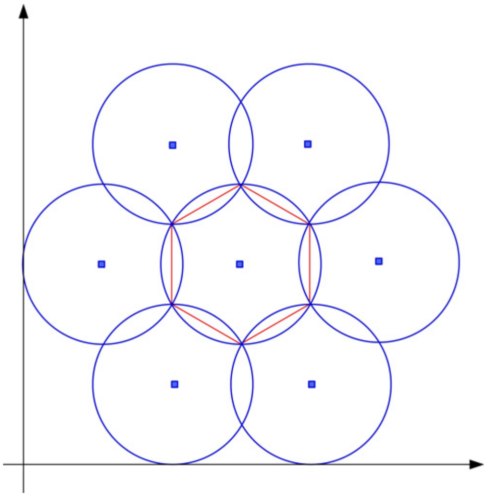


Fig. 2. Optimal hexagon-based sensor distribution.

coverage rate if they are covered only by one sensor and far from other sensors. In this case, it is necessary to overlap the sensor detection area to compensate for the low detection probability. Consider a grid point with coordinate (x, y) lying in the overlap region of sensors s_i and s_j. Since sensors within a cluster operate independently in their sensing activities, if neither s_i nor s_j covers grid point at (x, y), obviously, the probability of the grid point (x, y) being covered is denoted as,

$$c_{xy}(s_i, s_j) = 1 - (1 - c_{xy}(s_i))(1 - c_{xy}(s_j)). \tag{2.4}$$

Let c_{th} be the desired coverage threshold for all grid points. This implies that

$$\min_{x,y}\{c_{xy}(s_i, s_j)\} \geqslant c_{th}. \tag{2.5}$$

It can also be extended to a region which is overlapped by a set of k_{sov} sensors, $k_{sov} = |k|$, $k_{sov} = \{s_1, s_2, \dots, s_k\}$. The coverage in this case is given by:

$$c_{xy}(k_{sov}) = 1 - \prod_{s_i \in k_{sov}} (1 - c_{xy}(s_i)). \tag{2.6}$$

The coverage rate for the entire grid is calculated as the fraction of grid points exceeding the threshold c_{th}, and the coverage rate of the sensor set k_{sov} is defined as follows:

$$P_{cov}(k_{sov}) = \sum_{x=1}^{m}\sum_{y=1}^{n} c_{x,y}(k_{sov})/(m \times n). \tag{2.7}$$

2.3. Multi-objective optimization approach

Having measurements of system performance is only the first step towards solving the coverage problem. The important characteristic is that there are two interrelated functions, the network coverage rate and the number of working sensors. Increasing the coverage rate probably causes more working sensors, and reducing the working sensors will lead to a lower coverage rate. Obviously, these two conditions need to be considered simultaneously. It can be classified as a multi-objective optimization problem (MOP) as it involves simultaneous optimization of two objectives, which is distinguished from the single objective optimization problem (SOP). Formally, the mathematical description of the problem can be defined as follows:

Problem formulation: Given a set of N potential sensors, $k_{sov} = \{s_1, s_2, \dots, s_N\}$, the optimization goal is:

$$\boldsymbol{z} = \arg\max / \min(\boldsymbol{y}) = \arg\max / \min\{f_1(\boldsymbol{z}), f_2(\boldsymbol{z})\} \tag{2.8}$$

where $\boldsymbol{z} = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\} \in Z, y = \{y_1, y_2\} \in Y, Z$ is called the decision space and Y is called the objective space. Note that $f_1(\boldsymbol{z})$ is defined as the coverage rate of $\boldsymbol{z}$, and $f_2(\boldsymbol{z})$ is defined as the sensors used rate of k_{sov}.

$$\begin{cases} f_1(\boldsymbol{z}) = \sum_{x=1}^{m}\sum_{y=1}^{n} c_{xy}(\boldsymbol{z})/(m \times n) \\ f_2(\boldsymbol{z}) = |\boldsymbol{z}|/|k|. \end{cases} \tag{2.9}$$

To translate the max/min problem to a maximization problem, the optimization problem is defined as:

$$\boldsymbol{z} = \arg\max\{f_1(\boldsymbol{z}), 1 - f_2(\boldsymbol{z})\} \tag{2.10}$$

and the subset $\boldsymbol{z}$ is considered as the optimal sensor set of the target area.

3. Energy-efficient coverage control algorithm

Recently, genetic algorithms (GAs) are recognized to be well qualified to tackle multi-objective optimization problems. And now there is rapidly growing interest in the area of multi-objective genetic algorithms (MOGAs) [15]. The most important characteristic of MOGAs is the population-based search mechanism.

3.1. Definitions

Definition 1. For any two objective vectors y_1, y_2,

$$\begin{aligned} &y_1 = y_2, \quad \text{iff} \quad y_{1,i} = y_{2,i}, \text{for all } i \in [1, m] \\ &y_1 \geq y_2, \quad \text{iff} \quad y_{1,i} \geq y_{2,i}, \text{for all } i \in [1, m] \\ &y_1 > y_2, \quad \text{iff} \quad y_{1,i} > y_{2,i}, \text{for all } i \in [1, m]. \end{aligned} \tag{3.1}$$

The relation $\leq$, $<$ can be similarly defined.

Definition 2 (*Pareto Dominance*). For any two decision vectors $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$,

$$\begin{aligned} &z_1 \succ z_2, z_1 \text{ dominates } z_2 \quad \text{iff} \quad f(z_1) > f(z_2) \\ &z_1 \succeq z_2, z_1 \text{ partially dominates } z_2 \quad \text{iff} \quad f(z_1) \geqslant f(z_2) \\ &z_1 \sim z_2, z_1 \text{is indifferent to } z_2 \quad \text{iff} \quad f(z_1) \ngeqslant f(z_2) \wedge f(\boldsymbol{z}_2) \ngeqslant f(\boldsymbol{z}_1). \end{aligned} \tag{3.2}$$

Definition 3 (*Pareto Optimality*). A decision vector $\boldsymbol{z}$ is said to be non-dominated regarding a set $A \subseteq Z$, iff $\exists \boldsymbol{z} \notin A : \boldsymbol{z} \succ \boldsymbol{a}$. Furthermore, $\boldsymbol{a}$ is said to be *pareto* optimal iff $\boldsymbol{a}$ is non-dominated regarding Z.

Definition 4 (*Non-Dominated Sets and Fronts*). Let $A \subseteq Z$, the non-dominated set regarding A represented by A_p, is denoted as $A_p = \{\boldsymbol{a} \in A | \boldsymbol{a} \text{ is non-dominated regarding } A\}$. The corresponding set of objective vectors is defined as the non-dominated front regarding A, given by

$$F(A_p) = \{f(\boldsymbol{a}) \in A | \boldsymbol{a} \in A_p\}. \tag{3.3}$$

Furthermore, the non-dominated set regarding Z, represented by Z_p, is called the *pareto* optimal set, and $Y_p = F(Z_p)$ is the *pareto* optimal front.

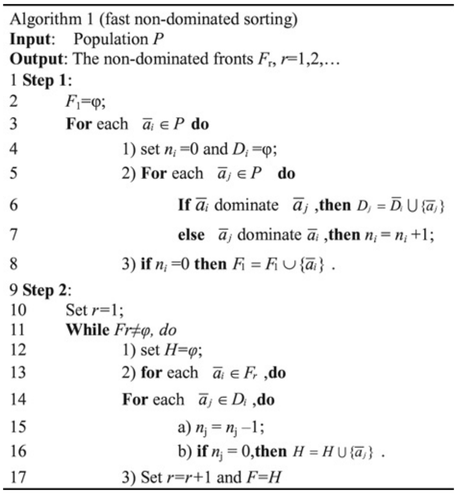
From the above definition, the primarily three goals that the multi-objective optimization algorithm used must have:

- The resulting non-dominated front should approximate the global *pareto* optimal front, although it may not be exactly a global optimum.
- The obtained solutions should be distributed uniformly so as to represent the entire *pareto* optimal set.
- The spread of the resulting non-dominated front should be maximized, such that for each objective as many values as possible could be covered by the obtained solutions.

3.2. Multi-objective genetic algorithm

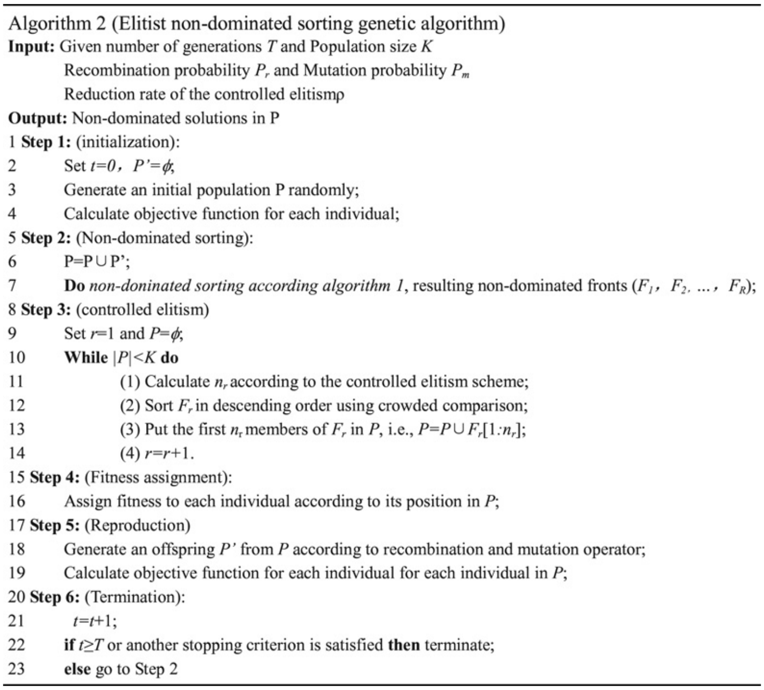
There are several well-known MOGAs [15,18,19]. NSGAII [19] (elitism non-dominated sorting genetic algorithm) is one of the most popular algorithms proposed as an improvement of NSGA [15]. In this paper, we present an approach based on NSGAII to find perfect solutions to the coverage problem in WSN.

By introducing the fast non-dominated sorting approach, the crowded comparison operator and elitism, it can search the *pareto* optimal front effectively. The goal of the fast non-dominated sorting approach is to find the non-dominated fronts, and the overall complexity of the algorithm is $O(mN^2) + O(N^2)$, where N is the population size. The whole procedure of the fast non-dominated sorting approach is described in Fig. 3. The crowded comparison operator uses two criteria to compare different solutions: the non-domination rank (r_i) and crowding distance (d_i). It makes sure that the search is guided towards a uniformly outspread *pareto* optimal front. The controlled elitism approach allows solutions from many different fronts to coexist in the population, making the recombination operator able to create diverse solutions, which is an important feature for preventing premature convergence. The elitist non-dominated sorting genetic algorithm (NSGAII) is described in Fig. 4. This algorithm is based on the structure of basic genetic algorithm. Some steps, e.g. initialization and reproduction



```
Algorithm 1 (fast non-dominated sorting)
Input:   Population $P$
Output: The non-dominated fronts $F_r$, $r$=1,2,…
Step 1:
    $F_1$=φ;
    For each  $\bar{a}_i \in P$ do
        1) set $n_i$ =0 and $D_i$ =φ;
        2) For each  $\bar{a}_j \in P$  do
            If $\bar{a}_i$ dominate  $\bar{a}_j$ ,then $D_j = \bar{D}_i \cup \{\bar{a}_j\}$
            else  $\bar{a}_j$ dominate $\bar{a}_i$ ,then $n_i = n_i + 1$;
        3) if $n_i$ =0 then $F_1 = F_1 \cup \{\bar{a}_i\}$ .
Step 2:
    Set $r$=1;
    While $Fr \neq \varphi$, do
        1) set $H=\varphi$;
        2) for each  $\bar{a}_i \in F_r$ ,do
        For each  $\bar{a}_j \in D_i$ ,do
            a) $n_j = n_j - 1$;
            b) if $n_j = 0$,then $H = H \cup \{\bar{a}_j\}$ .
        3) Set $r$=$r$+1 and $F$=$H$
```

Fig. 3. Procedure of the fast non-dominated sorting approach.



```
Algorithm 2 (Elitist non-dominated sorting genetic algorithm)
Input: Given number of generations $T$ and Population size $K$
       Recombination probability $P_r$ and Mutation probability $P_m$
       Reduction rate of the controlled elitismρ
Output: Non-dominated solutions in P
Step 1: (initialization):
    Set $t$=0， $P'$=$\phi$;
    Generate an initial population P randomly;
    Calculate objective function for each individual;
Step 2: (Non-dominated sorting):
    P=P∪P';
    Do non-doninated sorting according algorithm 1, resulting non-dominated fronts ($F_1$， $F_2$, …， $F_R$);
Step 3: (controlled elitism)
    Set $r$=1 and $P$=$\phi$;
    While |$P$|<$K$ do
        (1) Calculate $n_r$ according to the controlled elitism scheme;
        (2) Sort $F_r$ in descending order using crowded comparison;
        (3) Put the first $n_r$ members of $F_r$ in $P$, i.e., $P$=$P$∪$F_r$[1:$n_r$];
        (4) $r$=$r$+1.
Step 4: (Fitness assignment):
    Assign fitness to each individual according to its position in $P$;
Step 5: (Reproduction)
    Generate an offspring $P'$ from $P$ according to recombination and mutation operator;
    Calculate objective function for each individual for each individual in $P$;
Step 6: (Termination):
    $t$=$t$+1;
    if $t \geq T$ or another stopping criterion is satisfied then terminate;
    else go to Step 2
```

Fig. 4. Elitist non-dominated sorting genetic algorithm.

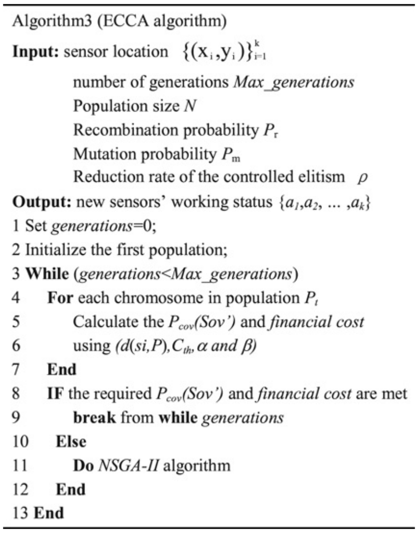
are identical. Moreover, the above discussed multi-objective search techniques, such as non-dominated sorting, crowded comparison, and controlled elitism, are implemented.

The chromosome of the GA contains all the building blocks to a solution for the genetic operators and the fitness function. Each individual sensor node is represented by a 1-bit binary number called gene. Gene defines the status of sensor nodes as follows:

$$a_i = \begin{cases} 1, & \text{if } s_i \text{ is selected} \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

With this coding representation, we set up the initial population in GA. Initial chromosomes are seeded randomly using a number of generators and the population of previous samples.

The implementation of recombination and mutation necessarily depends on the underlying genotypic representation. The recombination operator used in this paper is two-point crossover, which is a typical recombination for binary or other string-like chromosomes, and the crossing points are selected at random. Concerning the crossover rate p_c, we find that it



```
Algorithm3 (ECCA algorithm)
Input: sensor location $\{(x_i, y_i)\}_{i=1}^{k}$
        number of generations $Max\_generations$
        Population size $N$
        Recombination probability $P_r$
        Mutation probability $P_m$
        Reduction rate of the controlled elitism $\rho$
Output: new sensors' working status $\{a_1, a_2, \ldots, a_k\}$
Set generations=0;
Initialize the first population;
While (generations<Max_generations)
    For each chromosome in population $P_t$
        Calculate the $P_{cov}(Sov')$ and financial cost
        using $(d(si,P), C_{th}, \alpha$ and $\beta)$
    End
    IF the required $P_{cov}(Sov')$ and financial cost are met
        break from while generations
    Else
        Do NSGA-II algorithm
    End
End
```

Fig. 5. Pseudocode of the ECCA algorithm.

does not have a significant influence, but $p_c = 0.9$ or $p_c = 1$ performs slightly better. The mutation operator is applied for each new generated child after crossover. It works by complementing some genes in the child's chromosome randomly. The mutation type used in this paper is the classical one for binary representation, that is, the swapping of the bits of each string (0 becomes 1 and vice versa) with some specific low probability. The mutation rate is recognized to have a great influence on the convergence characteristics of GAs.

After a new population has been produced through the genetic operators, selection is done in an extended space composed of all parent and offspring individuals. This extended sampling space allows large probability of mutation and crossover while keeping the population relatively stable. Assign each individual having two fitness functions (coverage rate and sensors used rate), by introducing the non-dominated sorting, crowded distance operator and elitism. The probability of selecting some individual to become a parent for the production of the next generation is proportional to its fitness value. Each time there are two solutions of different non-domination ranks, we prefer the higher one. Otherwise, if both the points belong to the same front, we prefer the one which has larger crowded distance. In addition, in order to assure that the best individual of each generation was not destroyed by the crossover and mutation operators during the evolution process, elitism was included in the algorithm, meaning that the current best individual at each generation of the algorithm always survived to the next generation.

Similar to the VFA algorithm [20], ECCA can be performed by the cluster head of the sensor nodes. As the cluster head is equipped with a better computing resource, it would save the computing power for each individual sensor. The complete algorithm is described in Fig. 5. Note that the generations for computing the sensors' status can be performed by the cluster head and only the final results obtained are sent back to the sensor nodes. Therefore, the actual status is performed only once by each sensor. To run ECCA algorithm, the cluster head needs to collect the sensors' key index sets and their current locations. The advantage of using a cluster head to implement the status updating algorithm is that the power consumption of individual sensors can be reduced. When the cluster head is not available, the algorithm can be performed by the individual sensors only based on its neighborhood information, and thus the parallel genetic algorithm is performed. When ECCA is performed, the sensors need to do status transformation after some generations. At the same time, computing and comparing the full coverage would not be feasible, and the number of generations must be limited to reduce the power consumption in the communication of sensors.

4. Main results

In this section, we present some experimental results as the performance evaluation of ECCA. Throughout these experiments, we set the communication radius as twice the sensing radius to ensure coverage connectivity.

In the first experiment, we place 100 potential sensor nodes in a 100 × 100 square field. Among them, 39 sensors are distributed in a hexagonal structure to cover the whole area, which is the optimal sensor distribution, as shown in Fig. 6(a). The other 61 sensors are randomly deployed in the area, so these nodes' x and y-coordinates are random. Using ECCA, the locations of the sensor nodes are selected as the working set. The recombination rate and mutation rate are set as 0.9 and 0.01 respectively, which corresponds to T. Back's statement that the optimal value of mutation rate is about the inverse of the chromosome length [21].

Better solutions are obtained in subsequent generations shown in Fig. 6. For instance, the 120th generation improves the coverage rate with a small compromise in the average number of working nodes. What is more, the solution shown

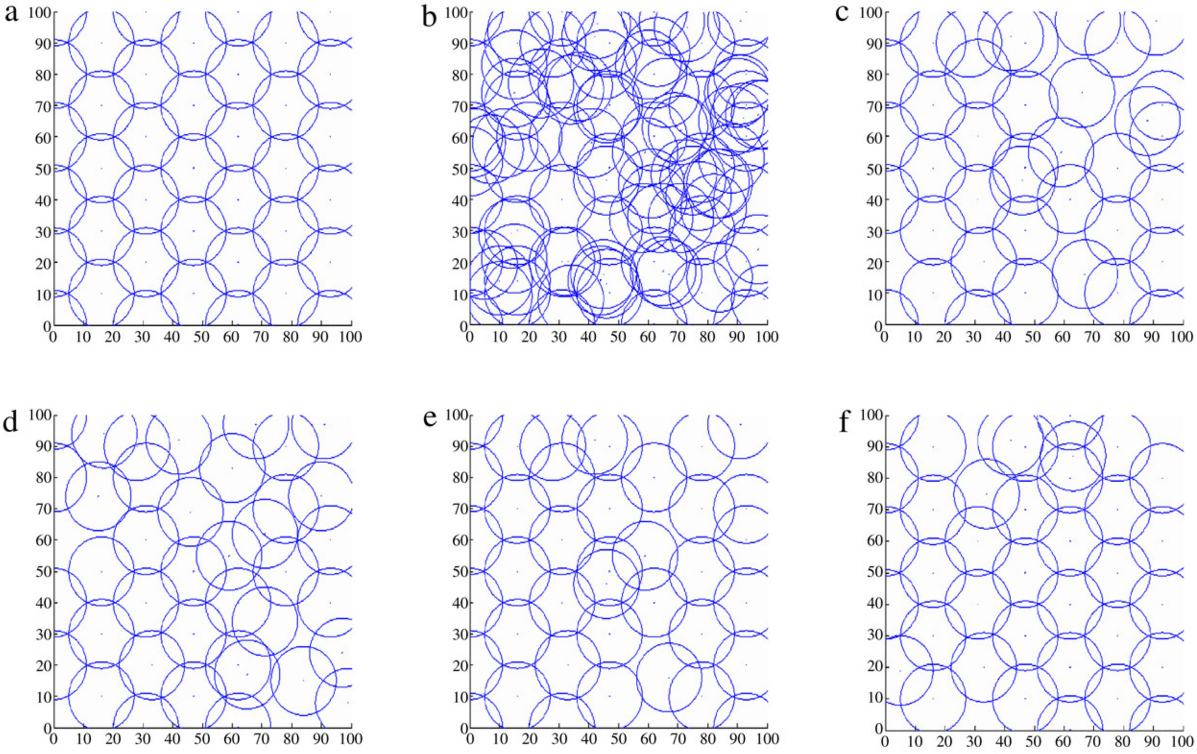


Fig. 6. Illustration of non-dominated solutions obtained in the first experiment. (a) optimal sensor distribution, 39 sensors. (b) initial distribution, 100 nodes. (c) the 60th generation, coverage rate 0.9392, 39 sensors. (d) the 80th generation, coverage rate 0.94, 40 sensors. (e) the 100th generation, coverage rate 0.9544, 39 sensors. (f) the 120th generation, coverage rate 0.956, 37 sensors.

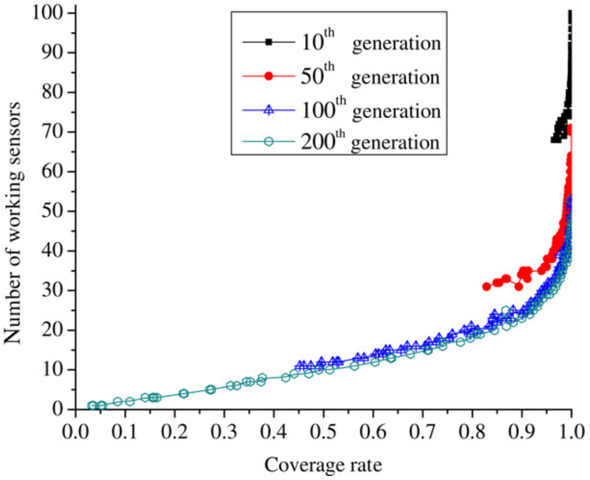


Fig. 7. Simulation results using ECCA in the second experiment.

in Fig. 6(f) is closest to the hexagonal geometry which is generally acknowledged to be the optimal sensor distribution. It requires only 37 nodes to achieve a coverage rate of 0.968.

In the second experiment, 200 sensor nodes are deployed randomly in a 100 × 100 square field. GA parameters used in this experiment are similar to the first experiment, where the mutation rate is set as 0.005. Specially, the sensing radius is set to 13 units. *Pareto* optimal solutions after running a number of generations are shown in Fig. 7. The initial population is randomly generated, and at the 10th generation, there are only a few non-dominated solutions, which constitute a non-dominated front (the short curve in Fig. 7). After 200 generations, many more non-dominated solutions are found. As a result, the non-dominated solutions get uniformly distributed making the *pareto* optimal front become fully outspread.

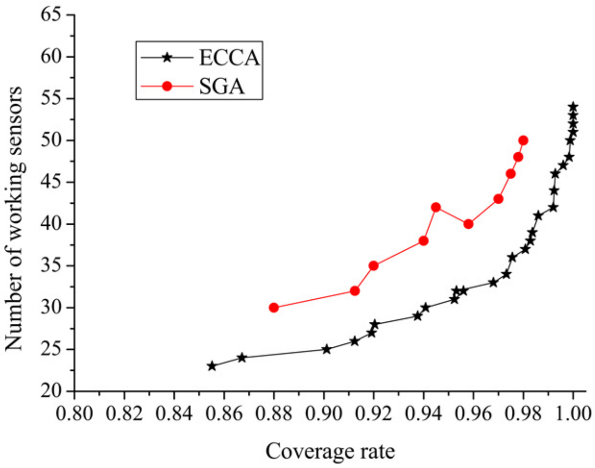


Fig. 8. Comparison between ECCA and SGA.

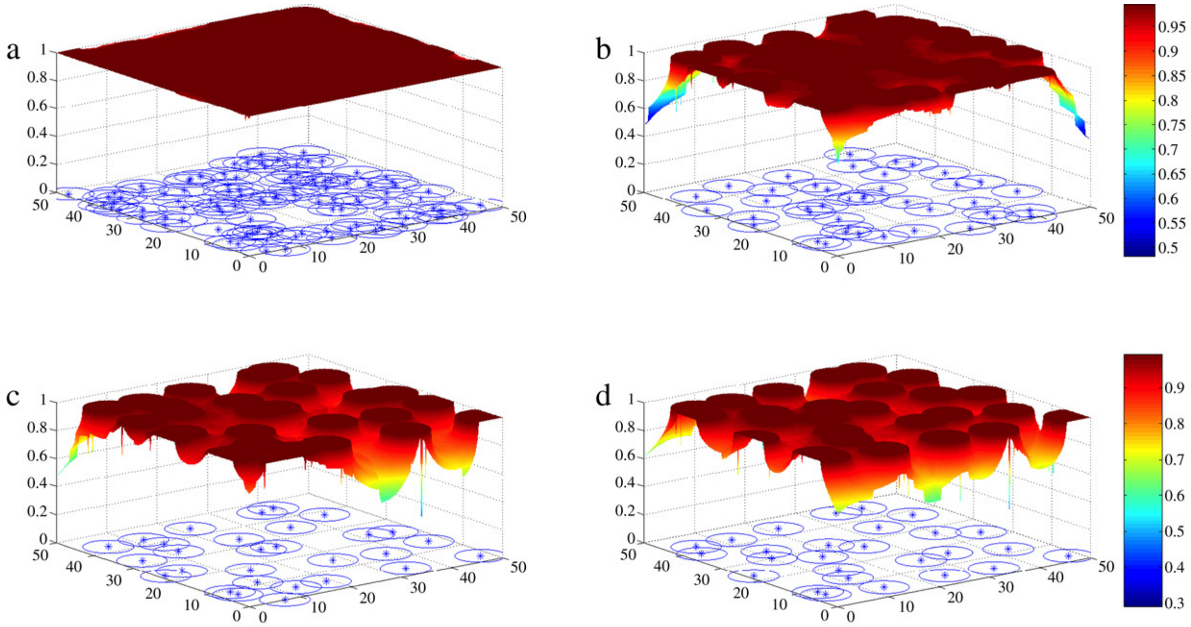


Fig. 9. Illustration of results obtained in the probabilistic detection model. (a) initial distribution, 100 sensors, full coverage; (b) the 40th generation, 55 sensors, coverage rate 0.9858; (c) the 80th generation, 29 sensors coverage rate 0.9804; (d) the 120th generation, 27 sensors, coverage rate, 0.9831.

Based on the tradeoff information obtained in the experiment, the decision maker can find a compromise solution between the conflicting objectives.

In the third experiment, we compare the ECCA and single objective genetic algorithm (SGA) in terms of sensing coverage and working nodes and the parameters are the same as those used in the second experiment. Here, SGA is obtained using the constraint method. Each time the coverage rate constraint is specified (from 0.9 to 1). The optimization objective is then to minimize the number of selected sensors. For clarity, only solutions with coverage rate higher than 0.85 are recorded. For each different constraint, SGA runs 200 generations and the best solution with the minimum number of sensors is shown in Fig. 8. The constraint method is very time-consuming because the optimization algorithm has to run many times in order to get different solutions. In contrast, ECCA can find various tradeoff solutions in a single run. Moreover, the solutions obtained by ECCA are even better than those of the constraint method.

In the fourth experiment, the probabilistic detection model is applied to study the performance of ECCA. Each sensor has a detection radius as 9 units ($r_e = 5$). Fig. 9 is a 3-D surface plot of working node number in different generations for the probabilistic sensor model with parameters set as $\lambda = 0.5$, $\beta = 0.5$, and $c_{th} = 0.7$. As discussed above, if $c_{th} \geq 0.7$, the grid is considered as being covered. With the color bar shown in Fig. 9, we can see each grid of the field is covered or not

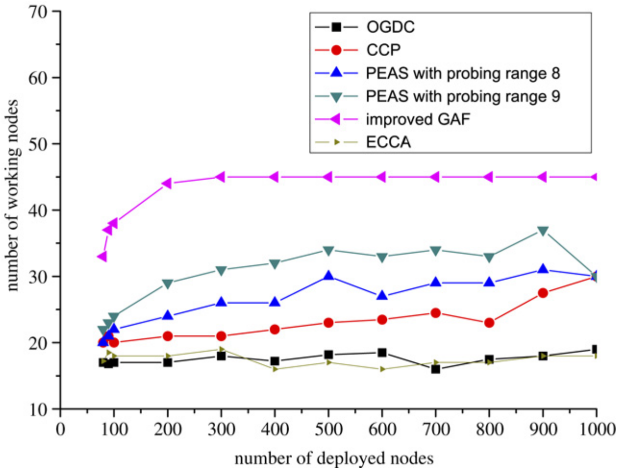


Fig. 10. Working nodes vs. node density.

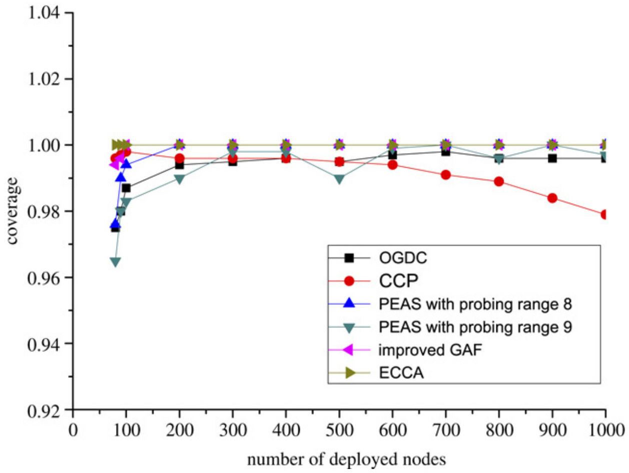


Fig. 11. Coverage rate vs. node density.

intuitively. Compared to the binary detection model, it needs more sensors to increase the number of grid points in which coverage exceeds the required threshold c_{th}. The results confirm that our estimate is accurate in general.

To compare the algorithm performance with OGDC, PEAS, CCP and improved GAF algorithm, another simulation is conducted in a 50×50 region where up to 1000 sensors are randomly distributed. The parameters used in this simulation are the same as [13], where the sensing range of each sensor is 10.

Every data point reported in Figs. 10 and 11 is an average of twenty simulations running unless specified. Both metrics are measured after the density control process is completed. ECCA needs only half as many nodes as the hexagon-based GAF algorithm and achieves a full coverage. This is better than OGDC which achieves less coverage rate in each round. As PEAS can control the number of working nodes by using different probing ranges, we tried with two different ranges, 8 and 9. In either case, the coverage rate achieved is worse than by ECCA. As CCP exchanges a lot of hello messages to maintain neighborhood information, the coverage rate of CCP actually decreases when the number of working nodes becomes very large. To our knowledge, the sponsor area based node scheduling algorithm is the best scheme that can preserve the original sensing coverage after turning off some nodes. ECCA can not only provide complete sensing coverage of the original network, but can also obtain more off-duty eligible nodes than other algorithms, thus longer system lifetime can be expected.

5. Conclusions and future work

In this paper, we have proposed a novel energy efficient coverage control algorithm for application-specific data gathering in wireless sensor networks. The basic goal of ECCA is to activate a minimum number of sensor nodes in a densely deployed environment. It is subject to two restrictions: one is the certain coverage rate of the network and the other is the number of the chosen nodes from the whole network. By searching the whole state-space, it can avoid the partial optimized solutions.

For a practical approach, a precise probabilistic sensor detection model is imported. We have also shown how a probabilistic detection model can be used in the coverage control scheme. The ECCA algorithm offers a number of important advantages, including negligible computation time and one-time resetting of the working state of the sensor nodes. Moreover, the desired sensor field coverage and model parameters can be provided as inputs to the ECCA algorithm, thereby ensuring flexibility.

In the future, our work will focus on heterogeneous sensor networks composed of different types of sensors and discuss the approaches to guarantee the expected coverage rate.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive suggestions, which improved the technical quality of the paper.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, IEEE Communications Magazine 40 (2002) 102–114.

[2] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, Next century challenges: Scalable coordination in sensor networks, in: J. Heidemann (Ed.), Proc. of the 5th Annual ACM/IEEE Int'l Conf. on Mobile Computing and Networking, Seattle, USA, 1999, pp. 263–270.

[3] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, A. Chandrakasan, Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks, in: S. Basagni, K. Sivalingam (Eds.), Proc. of the 7th Annual International Conference on Mobile Computing and Networking, Rome, Italy, 2001, pp. 272–287.

[4] S. Meguerdichian, F. Koushanfar, M. Potkonjak, M.B. Srivastava, Coverage problems in wireless ad-hoc sensor networks, in: Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, Anchorage, 2003, pp. 1380–1387.

[5] S. Slijepcevic, M. Potkonjak, Power efficient organization of wireless sensor networks, in: IEEE International Conference on Communications, ICC, Helsinki, Finland, 2001, pp. 472–476.

[6] W. Ye, J. Heidemann, D. Estrin, An energy-efficient MAC protocol for wireless sensor networks, in: Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM, New York, USA, 2002, pp. 1567–1576.

[7] F. Ye, G. Zhong, J. Cheng, S. Lu, L. Zhang, PEAS: A robust energy conserving protocol for long-lived sensor networks, in: Proc. of the 23rd International Conference on Distributed Computing Systems, ICDCS, Providence, USA, 2003, pp. 28–37.

[8] K. Chakrabarty, S.S. Iyengar, H. Qi, E. Cho, Grid coverage for surveillance and target location in distributed sensor networks, IEEE Transactions on Computers 51 (2002) 1448–1453.

[9] S. Meguerdichian, M. Potkonjak, Low power 0/1 coverage and scheduling techniques in sensor networks, UCLA Technical Reports, 030001, Jan. 2003.

[10] B. Chen, K. Jamieson, H. Balakrishnana, R. Morris, Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks, in: S. Basagni, K. Sivalingam (Eds.), Proc. of the 7th Annual International Conference on Mobile Computing and Networking, Rome, Italy, 2001, pp. 85–96.

[11] Y. Xu, J. Heidemann, D. Estrin, Geography-informed energy conservation for ad hoc routing, in: S. Basagni, K. Sivalingam, (Eds.), Proc. of the 7th Annual International Conference on Mobile Computing and Networking, Rome, Italy, 2001, pp. 70–84.

[12] F. Ye, G. Zhong, S. Lu, L. Zhang, Energy efficient robust sensing coverage in large sensor networks, Technical Report.

[13] H. Zhang, J.C. Hou, Maintaining sensing coverage and connectivity in large sensor networks, in: Ad-hoc and Sensor Wireless Networks 1, 2005, pp. 89–124.

[14] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, C. Gill, Integrated coverage and connectivity configuration in wireless sensor networks, in: Proc. of the First international conference on Embedded networked sensor systems, Los Angeles, CA, USA, 2003, pp. 28–39.

[15] N. Srinvas, K. Deb, Multi-objective function optimization using non-dominated sorting genetic algorithms, Evolutionary Computation 2 (1995) 221–248.

[16] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in: Proc. of the 33rd Annual Hawaii International Conference on System Sciences, Hawaii, 2000, pp. 3005–3014.

[17] S. Lindsey, C.S. Raghavendra, PEGASIS: Power-efficient gathering in sensor information systems, in: IEEE Aerospace Conference Proceedings, Montana, USA, 2002, pp. 1125–1130.

[18] C.M. Fonseca, P.J. Fleming, Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, in: Proc. of the 5th International Conference on Genetic Algorithms, ICGA, San Mateo, 1993, pp. 416–423.

[19] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2002) 182–197.

[20] Y. Zou, K. Chakrabarty, Sensor deployment and target localization based on virtual forces, in: Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM, San Francisco, USA, 2003, pp. 1293–1303.

[21] T. Back, Optimal mutation rates in genetic search, in: Proc. of the 5th International Conference on Genetic Algorithms, ICGA, San Mateo, 1993, pp. 2–9.