Available online at www.sciencedirect.com
ScienceDirect
Computer Communications 30 (2007) 2721–2734
computer communications
www.elsevier.com/locate/comcom

On efficient deployment of sensors on planar grid

Qishi Wu [a,*], Nageswara S.V. Rao [b,1], Xiaojiang Du [c], S. Sitharama Iyengar [d,2], Vijay K. Vaishnavi [e,3]

[a] Department of Computer Science, University Of Memphis, Memphis, TN 38152, USA
[b] Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6355, USA
[c] Department of Computer Science, North Dakota State University, Fargo, ND 58105, USA
[d] Department of Computer Science, Louisiana State University, Baton Rouge, LA 70803, USA
[e] Department of Computer Information Systems, Georgia State University, Atlanta, GA 30302, USA

Available online 15 June 2007

Abstract

One practical goal of sensor deployment in the design of distributed sensor systems is to achieve an optimal monitoring and surveillance of a target region. The optimality of a sensor deployment scheme is a tradeoff between implementation cost and coverage quality levels. In this paper, we consider a probabilistic sensing model that provides different sensing capabilities in terms of coverage range and detection quality with different costs. A sensor deployment problem for a planar grid region is formulated as a combinatorial optimization problem with the objective of maximizing the overall detection probability within a given deployment cost. This problem is shown to be NP-complete and an approximate solution is proposed based on a two-dimensional genetic algorithm. The solution is obtained by the specific choices of genetic encoding, fitness function, and genetic operators such as crossover, mutation, translocation for this problem. Simulation results of various problem sizes are presented to show the benefits of this method as well as its comparative performance with a greedy sensor placement method.

Keywords: Sensor deployment; Optimal surveillance; Genetic algorithm; Distributed sensor systems

1. Introduction

Distributed wireless sensor networks are becoming increasingly pervasive in many practical applications for either military or civil purposes, with the operational costs and characteristics of these networks considerably depending on the application. Sensors of different types are often deployed in these applications to meet strategic goals such as optimal surveillance and target detection. For optimal surveillance, the sensors must be deployed in a given region to achieve the maximum detection probability of potential targets with the total deployment expense constrained under a specified budget; for target detection, the sensors are positioned in such a way that every point in the surveillance region is covered by a unique subset of sensors [1,2]. The work presented in this paper is focused on the former.

Some theoretical problems that are closely related to the sensor deployment problem have been the subject of an enormous literature. For example, a typical facility-location problem, which is also regarded as a clustering problem, is to place a set of supply objects in an area of interest to serve a given set of demand points with the goal of minimizing certain cost functions. The Euclidean k-center cost function minimizes the maximum distance between a demand point and its nearest supply object. A corresponding decision version of this problem is to determine

* Corresponding author. Tel.: +1 901 678 1539; fax: +1 901 678 1506.
E-mail addresses: qishiwu@memphis.edu (Q. Wu), raons@ornl.gov (N.S.V. Rao), dxj@ieee.org (X. Du), iyengar@bit.csc.lsu.edu (S.S. Iyengar), vvaishna@gsu.edu (V.K. Vaishnavi).
[1] Tel.: +865 574 7517; fax: +865 574 0405.
[2] Tel.: +225 578 1495; fax: +225 578 1465.
[3] Tel.: +404 651 3891; fax: +404 651 3842.

0140-3664/$ - see front matter
doi:10.1016/j.comcom.2007.05.012

whether a given set of demand points can be covered by the union of k discs or balls of radius r. The cost function minimizing the sum of distances between each demand point and its nearest supply object leads to the Euclidean k-median problem, which is shown to be NP-hard for any dimension $d \geqslant 2$ [3]. The special case of $k = 1$ in a plane is the classical *Fermant–Weber problem* that traces back to the seventeenth century. A number of ε-approximation algorithms have been recently proposed to compute an approximate solution for the k-median problem [4–6].

Some sensor deployment problems with more practical considerations have also been studied in-depth for decades in a variety of scenarios. In the adaptive beacon placement, the strategy is to place a large number of sensors and then turn off some of them based on their localization information. In this context, Bulusu et al. [7,8] presented an adaptive algorithm based on measurements by considering the evaluations for spatial localization based on radio frequency-proximity. In a related area, Guibas et al. [9] presented a unique solution to the visibility-based pursuit evasion problem in robotics applications. In wireless sensor networks with the global knowledge of node positions, Meguerdichian et al. used a Voronoi diagram to compute the maximal breach path for the worst-case coverage and Delaunay triangulation to compute the maximal support paths for the best-case coverage [10]. Voronoi diagrams were also used in [26] by Wang et al. to discover coverage holes and several sensor deployment protocols were designed to provide high coverage by moving sensors from densely deployed areas to sparsely deployed areas. Both static and mobile sensor deployment schemes were considered in [27] to optimize sensing coverage and secure connectivity. In [22], sensor deployment strategies were investigated to provide sufficient coverage for distributed detection. Martinez and Bullo [24] studied optimal sensor placement and motion coordination strategies for mobile sensor networks in a target tracking scenario. To improve the integrity of sensed data and minimize the energy consumption for data communications, Ganesan et al. [25] tackled the combined optimization problem of sensor placement and transmission structure for data gathering.

Note that the problems described above in either theoretical or practical contexts are based on a simplistic service model of supply objects or sensor devices with binary deterministic detection capability. In other words, a point of interest is either covered or missed, solely depending on its distance from the center of a service point. In reality, however, the sensor coverage is not only determined by the geometrical distance [9], but also other factors such as environmental conditions and noise. In general, the probability of successful target detection by a sensor decreases in some way as the target moves further away from the sensor because of less received power, more noise and environmental interference. Therefore, the sensor detection is "probabilistic", which is the focus of this paper.

Moreover, the solutions to the above theoretical or practical problems do not consider the different expenses incurred by choosing different types of services and the number of service points is usually explicitly given as an input. Such a uniform sensor model with identical cost is not able to adequately capture the tradeoffs between the reliability and cost of a real sensor network. In practice, network implementers may choose from several available types of sensors with different detection qualities and costs, and the total expense of network implementation is usually limited by a total budget Moreover, the next-generation sensor networks must go beyond the deterministic coverage techniques to perform the assigned tasks such as online tracking/monitoring in unstructured environments.

Sensor deployment is a complex task in distributed sensor networks because of factors such as different sensor types and coverage ranges, sensor deployment and operational costs, and considerations for local and global coverage [11,12]. Essentially, the sensor deployment is an optimization problem, which often belongs to the category of multi-dimensional and nonlinear problems with complicated constraints. When the deployment locations are restricted to (discrete) grid points, this problem becomes a combinatorial optimization problem but still is computationally very difficult. In particular, this problem contains a considerable number of local maxima, and it is very difficult for the conventional optimization methods to obtain its global maximum [13].

In this paper, we formulate a generic sensor deployment problem over the planar grid to capture a sub-class of sensor network problems. We consider sensors of different types, wherein each type is characterized by a detection region and an associated detection probability distribution. Each sensor detects a target located in its detection region with certain probability and incurs certain cost for its deployment. We consider a Sensor Deployment Problem (SDP) that deals with choosing a set of sensors from an available pool of sensor types and placing them at various grid points to maximize the average detection probability while keeping the total expense under a specified limit.

We first show that this sensor deployment problem is NP-complete, and hence there is no polynomial-time algorithm that solves it exactly. We then present an approximate solution to this problem using a two-dimensional genetic algorithm (GA) for the case where the sensor detection distributions are statistically independent. Our solution is based on specifying the components of the genetic algorithm to suit the SDP. In particular, we specify the genetic encoding and fitness function to match the optimization criterion, and also specify the crossover, mutation and translocation operators to facilitate the search for the near-optimal solutions. In practice, near-optimality is often sufficient for this class of problems. We present simulation results for 50×50 or larger grids with five or more available sensor types when the *a priori* distribution of target is uniform. The proposed solution is quite effective in yielding solutions with high detection probability and low cost. We compare our method to a greedy approach of uniformly placing the sensors over the grid, and our method achieved significantly better detection probability.

The main focus of this paper is to conduct a theoretical study on a generic sensor deployment problem. We summarize the contributions of our work as follows:

- We adopt a probabilistic sensing model to characterize sensors' real performance in terms of coverage range, detection quality, and deployment cost.
- We formulate the sensor deployment problem using multiple sensors of different types over the planar grid with a budget constraint as a combinatorial optimization problem.
- We prove that the formulated sensor deployment problem is NP-complete.
- We successfully implemented and applied a two-dimensional genetic algorithm to the problem and obtained good suboptimal results. Comparison with a greedy approach of uniform deployment is conducted to demonstrate the performance superiority of the proposed solution.

The rest of this paper is organized as follows. In Section 2, we formulate the sensor deployment problem and construct a sensor model with probabilistic detection capability. We show this problem to be NP-complete by reducing the Knapsack problem to one of its special cases. In Section 3, we present an approximate solution based on a two-dimensional genetic algorithm. In Section 4, we discuss the experimental results and compare the performances of the genetic algorithm with those produced by a greedy algorithm. Then we conclude our work in Section 5.

2. Sensor deployment problem

We formulate the SDP in this section by specifying the surveillance region and sensor models, and then show it to be NP-complete.

2.1. Surveillance region

A planar surveillance region R is to be monitored by a set of sensors to detect a target T if located somewhere in the region (our overall method is applicable to the three-dimensional space). The planar surveillance region is divided into a number of uniform contiguous rectangular cells with identical dimensions as shown in Fig. 1. Each cell of R is indexed by a pair (i,j), and $C(i,j)$ denotes the corresponding cell. Let l_x and l_y denote the dimensions of a cell along x and y coordinates, respectively. As Fig. 1 shows, a circular coverage area is approximated by a set of cells within a certain maximum detection distance of sensor S_k.[4] The main reason we discretize the 2D space is to facilitate an efficient approximation of the sensor's sensing behavior and the region's coverage probability in the later computation of the GA-based solution. When the ratio of sensor detection range to cell dimension is very large, the sensor coverage area made up of many tiny rectangular cells will approach the circle.

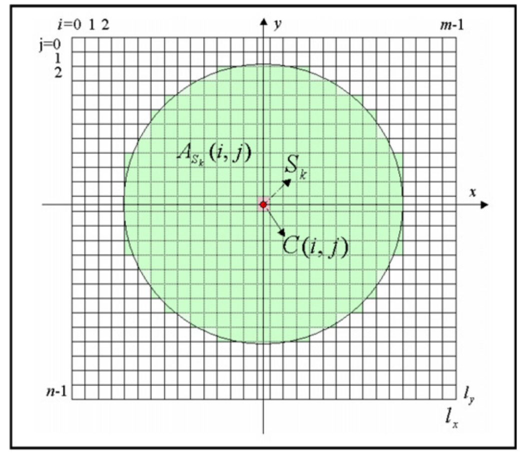


Fig. 1. Surveillance region R is divided into $m \times n$ rectangluar cells, with sensor S_k deployed in cell $C(i,j)$ convering area $A_{S_k}(i,j)$ in a probablistic sense.

Assume there are q types of sensors and a sensor of the kth type is denoted by S_k for $k \in \{1,2,\ldots,q\}$. There are N_k sensors of type k. A sensor S can be deployed in the middle of $C(i,j)$ to cover the discretized circular area $A_S(i,j)$ consisting of cells as shown in Fig. 1. A sensor S_k deployed at cell $C(i,j)$ detects the target $T \in A_{S_k}(i,j)$ according to the probability distribution $P\{S_k | T \in A_{S_k}(i,j)\}$ while incurring the cost $w(k)$.

A *sensor deployment* is a function $\Re$ from the cells of R to $\{\varepsilon,1,2,\ldots,q\}$ such that $\Re(i,j)$ is the type of sensor deployed at the cell $C(i,j)$; $\Re(i,j) = \varepsilon$ indicates no sensor is deployed, i.e. $w(\varepsilon) = 0$. The *cost of a sensor deployment* $\Re$ is the sum of cost of all sensors deployed in region R, which is given by

$$\text{Cost}(\Re) = \sum_{C(i,j) \in R} w(\Re(i,j)). \tag{1}$$

The *detection probability* $P\{\Re | T \in R\}$ of deployment $\Re$ is the probability that a target T located somewhere in region R will be detected by at least one deployed sensor. We now formally state the SDP considered in this paper:

Given a surveillance region R, cost budget Q, q types of sensors, and N_k sensors of type k, find a sensor deployment $\Re$ to maximize detection probability $P\{\Re | T \in R\}$ under the constraint $\text{Cost}(\Re) \leqslant Q$.

Informally, we are required to locate the sensors of various types on the grid points to achieve a maximum detection probability while keeping the deployment cost under a specified budget The *decision version* of the SDP asks for a deployment with detection probability at least A under the same cost condition, i.e., $P\{R | T \in R\} \geqslant A$ and $\text{Cost}(\Re) \leqslant Q$. The coverage problems have been extensively studied under various formulations related to the SDP [14,15],

[4] In Fig. 1 a cell is shaded if and only if its center is located within the maximum detection range of the sensor.

but we are not aware of works that directly address the SDP. The traditional polygon or rectangle coverage problems, studied in VLSI (Very Large Scale Integration) and related areas, focus on covering regions with minimum number of circles or rectangles and do not incorporate the probabilistic aspects of the sensors [15].

2.2. Probabilistic sensor detection model

We now briefly describe some detection probability distributions used in sensor deployment problems. The exact form of the distributions is not critical to the discussion in this paper, but they must be in computable form.

We consider that each sensor type is specified by its local detection probability of detecting a target at a point within its detection region. With regard to a sensor, detection is more likely as a target approaches the sensor. The cumulative detection probability of a sensor for a region is computed by integrating its local detection probability for detecting a target as the target gets close to the sensor, passes near the sensor, and then leaves it behind. In general, there are two ways of modeling a sensor detection performance based on how the integrated detection probability is approximated [16].

- *Definite range law approximation (cookie cutter):* In this model, only one parameter, i.e. maximum detection range is used. A target is always detected if it lies within a certain distance of the sensor, or it is never detected if it lies beyond the sensor's maximum detection range, as Fig. 2 shows.
- *Imperfect sensor approximation:* Besides the maximum detection range, a second parameter, mean detection probability (less than one) is specified for such a sensor model, as Fig. 3 shows.

Comparing the above two approximations, we suggest that the latter models the real situation more reasonably. Based on the imperfect sensor approximation, a more realistic sensor performance model may be specified by a Gaussian cumulative detection probability instead of mean

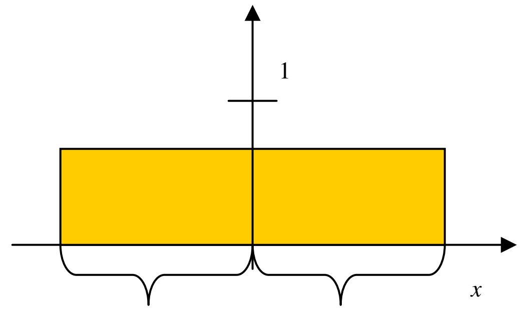


Fig. 3. Imperfect sensor approximation.

detection probability. Given the detection probability density function $p_{S_k}(x)$ for a sensor of type k, the detection probability $P\{S_k|T \in C(i,j)\}$ for cell $C(i,j)$ is given by

$$P\{S_k|T \in C(i,j)\} = \int_{x \in C(i,j)} p_{S_k}(x)\,\mathrm{d}x. \tag{2}$$

After obtaining the individual detection probabilities for all the cells covered by sensor S_k, we employ Gaussian function to compute the cumulative detection probability. The Gaussian cumulative detection probability approximating a real sensor detection performance is defined by

$$P(S_k,\tau,\alpha_{S_k}) = P\{S_k,\tau,\alpha_{S_k}|T \in A_{S_k,\tau}\} = \mathrm{e}^{-\frac{\tau^2}{2\cdot\alpha_{S_k}^2}}, \tau \in [0, d_{S_k}], \tag{3}$$

where τ is the distance from the target to the sensor. The sensor detection quality coefficient α_{S_k} determines the shape of the detection probability curve. Distance τ is in the range between 0 and the maximum detection distance d_{S_k}. The attribute parameters of a sensor and its typical integrated detection probability of Gaussian sensor approximation is shown in Fig. 4, where the measure of detection probability is assumed to reach 1 when the target is very close to the sensor. We employ this distribution in our computations in Section 4, but our genetic algorithm method is applicable to other computable sensor distributions.

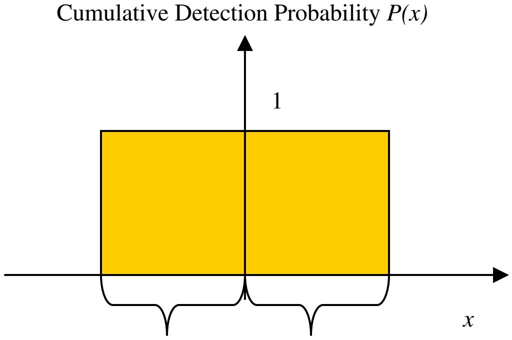


Fig. 2. Definite range law approximation.

2.3. NP-completeness of the sensor deployment problem

We now show that the sensor deployment problem is NP-complete by reducing the Knapsack Problem (KP) [23] to a special case of the SDP, wherein each sensor monitors a single cell with a specified probability. We consider q-KP: Given a set U of n items such that for each $u \in U$, we have size $s(u) \in Z^+$ and the value $v(u) \in Z^+$, Does there exist a subset $V \in U$ of exactly q items such that $\sum_{u \in V} s(u) \leqslant B$ and $\sum_{u \in V} v(u) \geqslant K$ for given B and K? Note that we require exactly q items as opposed to an unrestricted value in the usual KP; note that KP and q-KP are polynomial equivalent [14] since $q \leqslant n$ and the input for either problem instance has at least n items.

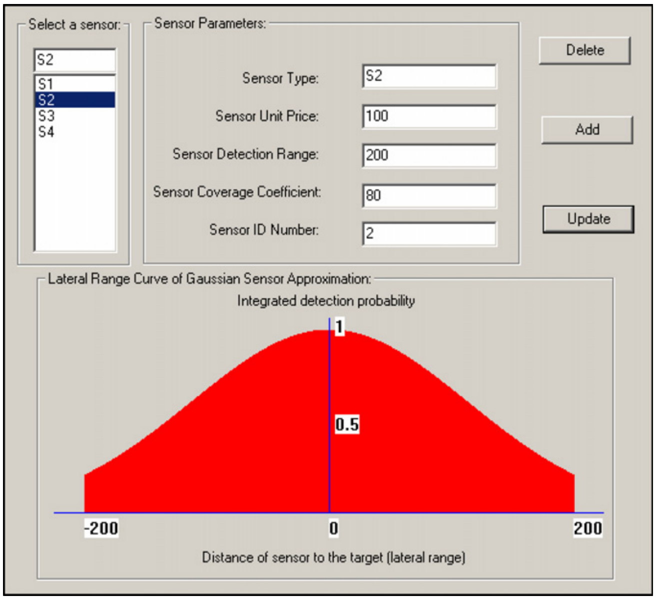


Fig. 4. Sensor attribute parameters and integrated detection probability of Gaussian sensor approximation.

Consider the decision version of the SDP that asks for exactly q sensors to be deployed. We reduce the q-KP to a particular restriction of the SDP, denoted by q-SDP, such that we are given only one sensor of each type, and each sensor S monitors a single cell and when two sensors are located in the same cell only one of them detects the target (i.e. suitable conditional probabilities are zero). For this special case, to maximize the detection probability, without the loss of generality each cell is assumed to occupy by no more than one sensor. Furthermore, under the uniform prior distribution of target T in cells combined with the non-overlapping sensor regions, the probability of detection is simply the average of the probability of detection of the deployed sensors. Considering the sensor deployment $\Re$ deploys q sensors, we have $P\{\Re|T\in R\} = \frac{1}{q}\sum_{\Re(i,j)=k;\Re(i,j)\neq\varepsilon}P\{S_k|T\in A_{S_k}\}$. Given an instance of q-KP, we map each $u\in U$ to a sensor S_u such that its cost and value are given by $w(u)=s(u)$ and $P\{S_u|T\in A_{S_u}\}=\frac{v(u)}{\sum_{a\in U}v(a)}$. Then we specify the sensor cost bound as $Q=B$ and the detection probability as $A=\frac{K}{q\sum_{a\in U}v(a)}$. Given a solution to q-KP, a solution to q-SDP exists by just placing the sensors corresponding to the members of V on non-overlapping grid points. Let (i_u,j_u) be the cell receiving a sensor due to $u\in V$. Then we have

$$\sum_{\Re(i,j)\neq\varepsilon;\Re(i,j)=k} w(k) = \sum_{\Re(i_u,j_u):u\in V} w(k) = \sum_{u\in V} s(u) \leqslant Q,$$

which satisfies the first condition for q-SDP. For the second condition, we have

$$\begin{aligned}P\{\Re|T\in R\} &= \frac{1}{q}\sum_{\Re(i,j)\neq\varepsilon;\Re(i,j)=k} P\{S_k|T\in A_{S_k}\}\\ &= \frac{1}{q}\sum_{\Re(i,j)\neq\varepsilon;\Re(i,j)=k}\frac{v(u)}{\sum_{a\in U}v(a)} \geqslant A.\end{aligned}\tag{4}$$

Given a solution to the sensor deployment problem, we obtain the solutions to q-KP by choosing the items corresponding to the deployed sensors. Let $u_{(i,j)}$ denote the chosen item in corresponding to sensor located at $\Re(i,j)$. The first condition for q-KP follows from:

$$\sum_{u\in V} s(u) = \sum_{u(i,j)} s(u_{(i,j)}) = \sum_{\Re(i,j)=k} w(k) \leqslant B. \tag{5}$$

The second condition for q-KP follows from:

$$\begin{aligned}\sum_{u\in V} v(u) &= \sum_{u(i,j)} v(u_{(i,j)}) = \sum_{a\in U} v(a)\sum_{\Re(i,j)\neq\varepsilon;\Re(i,j)=k} P\{S_k|T\in A_{S_k}\}\\ &= q\sum_{a\in U} v(a)P\{\Re|T\in R\} \geqslant Aq\sum_{a\in U} v(a) = K.\end{aligned}\tag{6}$$

We have shown that the SDP is NP-Complete even when severe restrictions are imposed on the joint distributions, which is an indication of the computational complexity of this problem. Thus it is unlikely that polynomial time solutions that optimally solve the SDP exist, which motivates us to consider approximate solutions.

2.4. Sensor detection probability under independence condition

In this section, we consider a restricted version of the SDP such that sensors satisfy certain statistical independence condition, which enables the joint detection probabilities be efficiently computed. To guarantee high probability of detection, sensor detection range should overlap to ensure that critical areas of the surveillance region are covered by at least one sensor [17]. The local detection probability $P\{\Re|T\in C(i,j)\}$ must be suitably accumulated for each cell $C(i,j)$ covered by two or more sensors. To determine the sensor detection probabilities for such cells, we first consider a simple case with two detection probabilities, $P\{S_m|T\in C(i,j)\}$ and $P\{S_n|T\in C(i,j)\}$ corresponding to sensors S_m and S_n which overlap in $C(i,j)$. The detection probability $P\{S_m\vee S_n|T\in C(i,j)\}$ is the probability of detecting the target successfully by at least one of the two sensors. Let $P(S_l)$ denote $P\{S_l|T\in C(i,j)\}$, for $l=m$, n and $P(S_m\vee S_n)$ denote $P\{S_m\vee S_n|T\in C(i,j)\}$. There are two mutually exclusive and collectively exhaustive cases for the successful detection, as Fig. 5 shows.

We assume that sensors S_m and S_n are statistically independent such that $P(S_m\wedge S_n)=P(S_m)P(S_n)$. When there is an overlap between two sensor detection areas, the probability that the target is detected by either sensor is calculated as the sum of two individual detection probabilities minus their joint detection probability. This joint detection

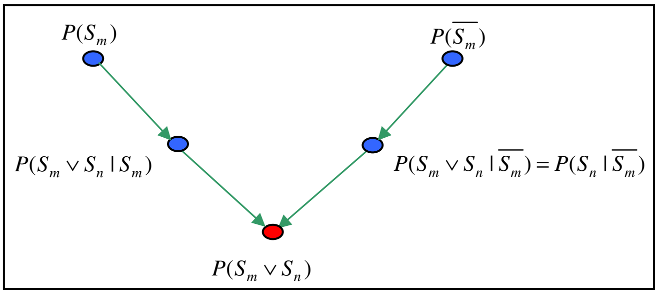


Fig. 5. Multiple cases to successful detection in the simplest case of two sensors.

probability denotes the probability that the target is detected simultaneously by both sensors within the overlapping area, i.e. $P(S_m \vee S_n) = P(S_m) + P(S_n) - P(S_m)P(S_n)$. Therefore reducing this overlapping area increases the overall detection probability.

For a general case of n sensors covering a cell, by the inclusion–exclusion principle [13] we have

$$P(S_1 \vee S_2 \vee \cdots \vee S_n) = P(S_1 \vee S_2 \vee \cdots \vee S_{n-1}) + P(S_n) - P(S_1 \vee S_2 \vee \cdots \vee S_{n-1}) \cdot P(S_n) \cdots\cdots \quad (7)$$
$$= \sum_{i=1}^{n} P(S_i) - \sum_{1 \leqslant i<j \leqslant n} P(S_i)P(S_j) + \sum_{1 \leqslant i<j<k \leqslant n} P(S_i)P(S_j)P(S_k) + \cdots + (-1)^{n-1} P(S_1)P(S_2)\cdots P(S_n)$$

The overlap of local detection probabilities for n sensors is computed by applying the simple formula in Eq. (7) repeatedly for each additional sensor as follows to compute $P\{\Re|T \in C(i,j)\}$ for each cell $C(i,j)$:

Step 1. Initialize local coverage probabilities and total cost.
Step 2. Locate a cell in which a sensor is deployed.
Step 3. Determine the sensor type.
Step 4. Update total cost.
Step 5. Compute the detection area of this sensor using Eq. (7).
Step 6. For each cell within the discretized circular detection area, compute the overlapping detection probability.
Step 7. Update local detection probability for each cell covered by this sensor.
Step 8. Go back to Step 2 until all cells in the whole surveillance region are examined.

The details of the algorithm to compute local coverage probabilities and total cost outlined above are given as follows:

Input: sensor deployment scheme $\Re$

Output: local coverage probability $P\{\Re|T \in C(i,j)\}$ for each cell $C(i,j)$ and total cost, where $i = 0, 1, 2, \ldots, m-1$, $j = 0, 1, 2, \ldots, n-1$

```
Begin
   Initialize P{ℜ | T ∈ C(i.j)} to 0;
   Initialize Cost (ℜ) to 0;
   for i = 0 to m − 1
   {   for j = 0 to n − 1
       {   let S_k = ℜ(i, j);
           if (S_k = = ε)
               continue
           else
               Update(S_k)
       }
   }
End
```

Auxiliary function *Update*(*Sensor k*)

```
Update(Sensor S_k)
Begin
  let Cost (ℜ) = Cost (ℜ) + w(S_k);
  let a = ⌊ d_{S_k} / l=x ⌋;
  let b = ⌊ d_{S_k} / l=y ⌋;
  for r = − a to a
     for s = − b to b
     {   let τ = sqrt((r * l_x)^2 + (s * l_y)^2);
         if (τ ≤ d_{S_k})
         {
             let overlap = P{ℜ|T ∈ C(i + r, j + s)} * P(S_k, τ, α_{S_k});
             let P{ℜ|T ∈ C(i + r, j + s)} = P{ℜ|T ∈ C(i + r, j + s)} + P(S_k, τ, α_{S_k}) − overlap;
         }
     }
End
```

Then we compute $P\{\Re|T\in R\}$ by adding all local detection probabilities in the surveillance region given by

$$P\{\Re|T\in R\}=\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}P\{\Re|T\in C(i,j)\}*P\{T\in C(i,j)\}, \tag{8}$$

which is the objective function to be maximized under the cost constraint $\text{Cost}(\Re)\leqslant Q$. Given *a priori* distribution $P\{T\in C(i,j)\}$ of target T in a computable form and the sensor distributions, the objective function is computable.

This version of the SDP, namely under statistical independence condition, can be shown to be NP-complete by a simple extension of the results of the last section. Under the statistical independence, within each cell the probability of joint detection is the product of individual probabilities and hence is smaller than either. Thus, any overlapping sensors within a cell can be separated to increase the probability of detection, and it suffices to consider no more than one sensor per cell. Rest of the proof follows the last section: under the restriction that each sensor detects target in the cell it is currently located, this problem reduces to q-SDP in last section, which shows the current problem to be NP-complete by restriction.

3. Approximate solutions using genetic algorithm

Genetic algorithm is a computational model that simulates the process of genetic selection and natural elimination in biological evolution [18]. It has been widely used to solve the combinatorial and non-linear optimization problems with complex constraints or non-differentiable objective functions [19,20]. The computation of genetic algorithm is an iterative process towards achieving the global optimality. During the iterations, candidate solutions are retained and ranked according to their quality. A fitness value is used to screen out unqualified solutions. Genetic operations of crossover, mutation, translocation, inversion, addition and deletion are then performed on those qualified solutions to create new candidate solutions of the next generation. The above process is carried out repeatedly until certain stopping or convergence condition is met For simplicity, a maximum number of iterations can be chosen to be the stopping condition. The variation difference of the fitness values between two adjacent generations may also serve as a good indication for convergence. To utilize the genetic algorithm method, various parts of the SDP must be mapped to the components of the genetic algorithm as will be shown in this section.

3.1. Genetic encoding for sensor deployment

Since a candidate solution to the SDP requires a two-dimensional sensor ID matrix, we adopt a two-dimensional numeric encoding scheme to make up the chromosomes instead of the conventional linear sequence. As Fig. 6 shows, we construct a sensor ID matrix for a possible sensor deployment scheme. Each element in the matrix on the right-hand side corresponds to a cell within a surveillance region on the left-hand side. As mentioned above, an empty value ε in the matrix indicates that its corresponding cell has no sensor deployed in and should be covered by the sensors deployed in its neighborhood area. Furthermore, we arrange q types of available sensors in the following order:

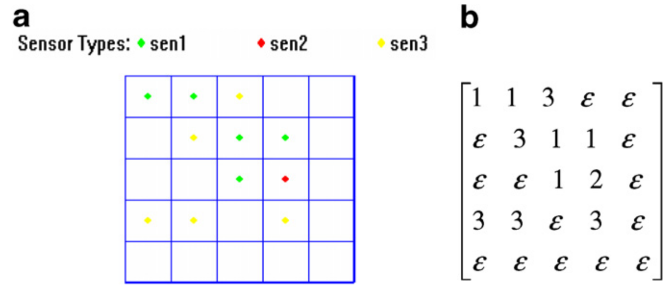


Fig. 6. (a) Visual illustration of a deployment solution using three sensor types; (b) corresponding sensor ID matrix.

$$d_{S_1}/w(s_1)\geqslant d_{S_2}/w(s_2)\geqslant\cdots\geqslant d_{S_k}/w(s_k)\geqslant\cdots\geqslant d_{S_q}/w(s_q).$$

Recall that d_{S_k} and $w(s_k)$ are the maximum detection distance and cost of sensor of type k, respectively. Besides, each sensor type has a sensor coverage coefficient that determines the variation of its detection capability along the target-sensor distance. The rank of ratio is used to decide the probability of sensor type selected during the population initialization as well as the addition operation.

3.2. Fitness function

We construct the fitness function from the objective function as

$$f(\Re)=P\{\Re|T\in R\}+g, \tag{9}$$

where g is the penalty function for overrunning the constraint, which is defined by

$$g=\begin{cases}0, & \text{Cost}(\Re)\leqslant Q\\ \delta*E_m*(Q-\text{Cost}(\Re))/Q & \text{Cost}(\Re)>Q\end{cases} \tag{10}$$

where δ is a proper penalty coefficient and is set to 100, and $E_m=\max_k\{d_{S_k}/w(S_k)\}$.

3.3. Selection of candidates

The selection operation, also called reproduction operation, retains good candidate and eliminates others from the population based on the individual fitness values. It aims to inherit good individuals either directly from the last generation or indirectly from the new individuals produced by mating the old individuals. The frequently used selection mechanisms include fitness proportional model, rank-based model, expected value model, and elitist model [19].

In our implementation, the survival probability B_i for each individual (solution) $\Re_i$ is computed based on the following fitness proportional model:

$$B_i = f(\Re_i) \Big/ \sum_{j=1}^{M} f(\Re_j), \tag{11}$$

where M is the population size. The hybridization individuals are produced according to the selection rule so that the individual with bigger B_i has a higher probability to survive.

3.4. Implementation of genetic operators

The solution set of each new generation after initial population is generated as follows. Randomly select two hybridization individuals $\Re_u$ and $\Re_v$, and combine them to get two other individuals $\Re'_u$ and $\Re'_v$ of new generation by using combinatorial rules of crossover, mutation, inversion, translocation, addition and deletion [21]. Some of these genetic operators are carried out on a two dimensional basis. Except for crossover, all the other operators operate on only one parent solution. This process continues until all M individual solutions of new generation are created.

1. *Crossover:* Crossover is an operation of segment exchange for two solutions. Given two parent (hybridization individuals) solutions on the left side in Fig. 7, a two-dimensional two-point crossover operation produces two child solutions on the right side as Fig. 7 illustrates. Both window size and location for crossover are selected randomly.
2. *Translocation:* The objective of translocation is to exchange information between different segments within a single solution, as Fig. 8 illustrates. Same as in crossover operation, the translocation window size is selected at random as well as its source and destination position.
3. *Mutation:* The mutation operator chooses one or more cells randomly in the surveillance region and changes their values by the preset mutation probability. Actually, it is a combination of addition and deletion operators. As Fig. 9 shows, a sensor of type A is moved to a new place, a sensor of type B is replaced by a sensor of type C, a sensor of type C is deleted, and a sensor of type D is replaced by a sensor of type B. The selection probability of a certain type of sensor for addition operator depends on the ratio of its detection range to unit price, so does for the population initialization.

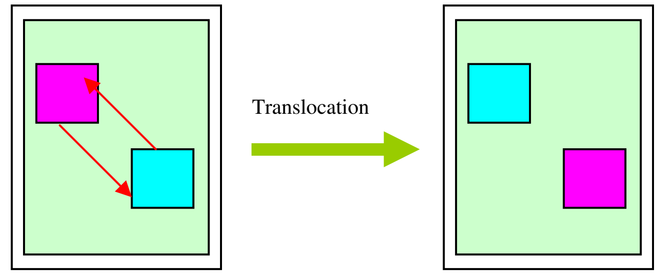


Fig. 8. Two-dimensional translocation.

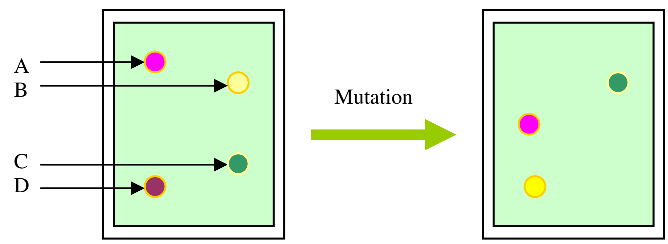


Fig. 9. Mutation operator.

4. Computational results

In our work, we consider a probabilistic sensing model with multiple sensor characteristics and a global budget constraint, while in many other existing solutions, only a single deterministic sensing model is used. These differences in the problem formulation make it difficult to conduct direct comparison of our approach to others. In this section, we present simulation results of the approximation solution based on genetic algorithm (GA) and compare its performances with those of a greedy solution based on uniform placement (UP) of sensors. The UP method employs two greedy algorithms in terms of selecting the sensor that provides the maximum ratio of detection range to unit price and deploying them in the optimal positions.

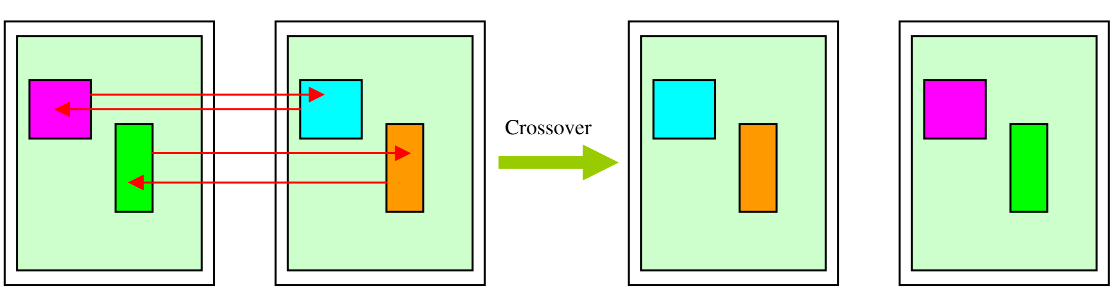


Fig. 7. Two-dimensional two-point crossover.

Specifically, we distribute a maximum set of "best" sensors under a given budget limit in the surveillance region in an optimal way such that the overlapping areas between adjacent sensors are minimized.

Both algorithms are implemented in C++. We consider that the target has a uniform *a priori* distribution in the surveillance region such that the probability of target T appearing in a cell $C(i,j)$ is $P\{T \in C(i,j)\} = 1/(m \cdot n)$. Therefore, from Eq. (8) calculating the average detection probability, we have our objective function as follows:

$$P\{\Re | T \in R\} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} P\{\Re | T \in C(i,j)\}/(m \cdot n), \qquad (12)$$

with the constraint $\text{Cost}(\Re) \leqslant Q$. We utilize this formula for our computations in this section, and the case in which the target has other *a-prior* distributions can be handled using Eq. (8) in place of above expression.

4.1. First case of small region size

In the first case, we consider a surveillance region of 50×50 cells with five types of sensors as listed in Table 1. All parameters used by the genetic algorithm are specified in Table 2. The investment limit is set to be 1800 U expense and the maximum generation number is set to be 200. Upon the completion of optimization process, the GA achieves a sub-optimal deployment scheme with detection probability of 94.52% for the surveillance region within the investment budget

The graphical representation of the deployment scheme computed by the genetic algorithm is illustrated on the left side of Fig. 10(a). A local detection probability is given in each cell for evaluation.[5] The GA optimization process curve is plotted on the right side of Fig. 10(a) with the generation number represented by the x-axis and the corresponding fitness value represented by the y-axis. Its corresponding 3-D display of the local coverage probabilities of 50×50 cells is plotted in Fig. 10(b).

We conduct a series of GA computations for this surveillance region to investigate the impact of the investment limit on the average detection probability. Fig. 11 shows such a plot where the average detection probability increases in response to the increment of investment limit. It is observed that increasing the investment beyond 1800 U does not pay off as much since the incremental gain of the detection probability is marginal. This graph could help determine a proper sensor deployment scheme within any given cost for a given surveillance region. More importantly, it gives a good hint for choosing a proper initial investment limit for the given surveillance region.

[5] The values of local detection probabilities are overwritten by those values of neighbor cells except for the right-most column due to the relatively small display screen, so is the case in Fig. 11.

Table 1
Attribute parameters of five types of sensors used in Case 1

Sensor Type	Sensor ID	Unit price	Detection range	Detection coefficient
Sen1	1	86	124	80
Sen2	2	111	159	78
Sen3	3	113	163	68
Sen4	4	135	195	68
Sen5	5	139	200	84

Table 2
Parameters used by the genetic algorithm in Case 1

Parameters in genetic algorithm	Values
Maximum generation number	200
Maximum investment limit	1800
Population size	30
Probability of crossover	0.99
Probability of mutation	0.24
Probability of deletion	0.10
Probability of translocation	0.99
Probability of inversion	0.82
Probability of addition	0.10

Fig. 12(a) shows the computational result of the same surveillance region based on UP using the sensors of type with the maximum ratio of detection range to unit price. The UP achieves an average detection probability of 88.83% within the specified investment budget Its corresponding 3-D display of the local coverage probabilities of 50×50 cells is plotted in Fig. 12(b).

4.2. More cases of larger region sizes

We now consider larger surveillance regions and more sensor types with different parameters. Same as in Case 1, the UP only uses the sensors of type with the maximum ratio of detection range to unit price. The comparisons of computational results between the genetic algorithm and the uniform placement are summarized in Table 3.

For further comparison, the average detection probabilities computed by both GA and UP for each region size are plotted as clustered columns in Fig. 13. In all the simulation cases we studied, the GA achieved significantly higher probability of detection than the UP under the given cost bound. Also, these experimental results show that our GA-based approach is able to scale well with the region size, number of sensor types, and various constraints on the investment budget

5. Conclusions

Optimal surveillance and target detection are critical but difficult task of the sensor deployment, particularly if the sensors are of different types and have different costs. We formulated a general sensor deployment problem for a

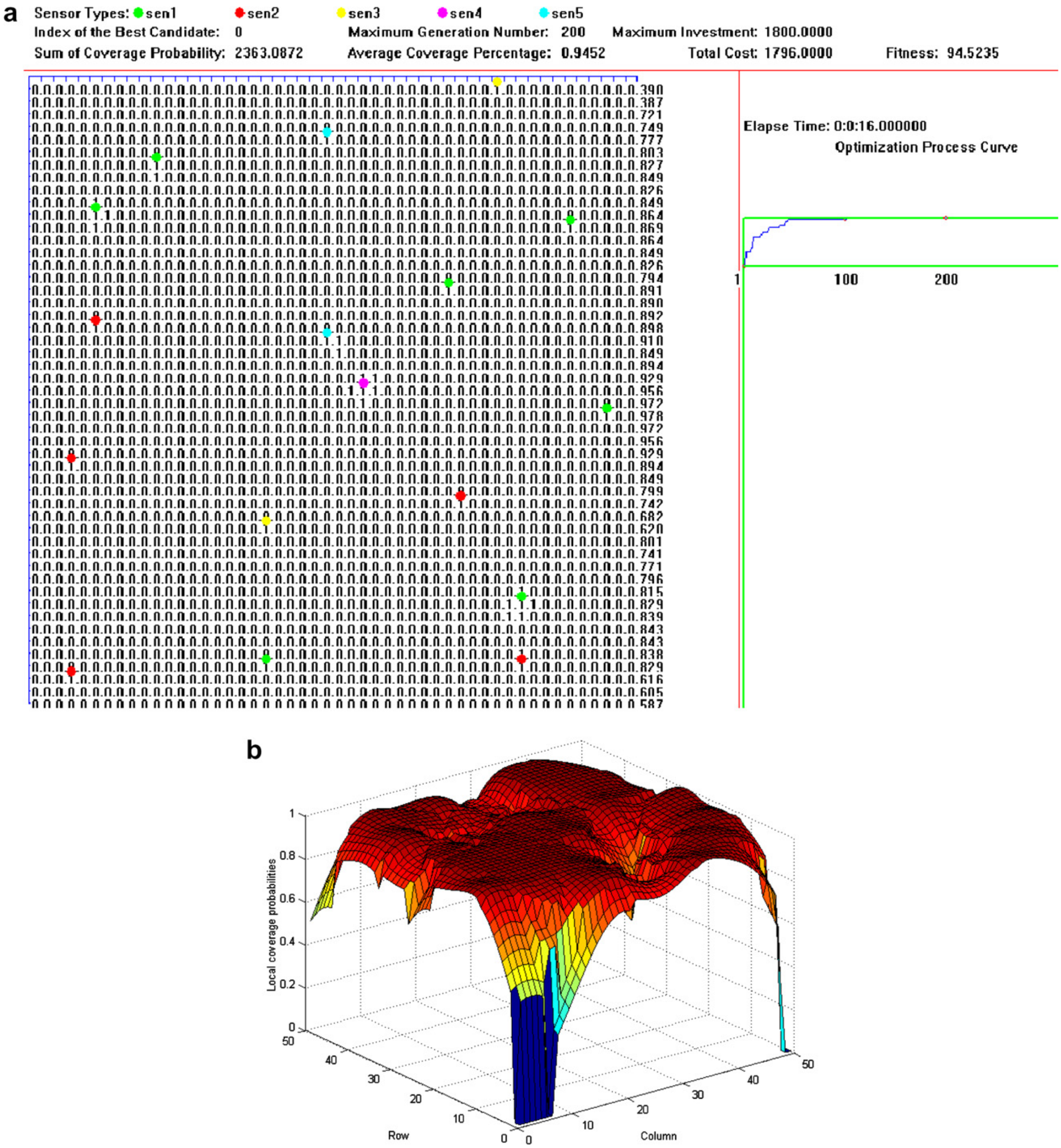


Fig. 10. (a) Graphical representation of deployment scheme. (b) 3-D display of the local coverage probabilities for a surveillance region with 50×50 cells based on genetic algorithm.

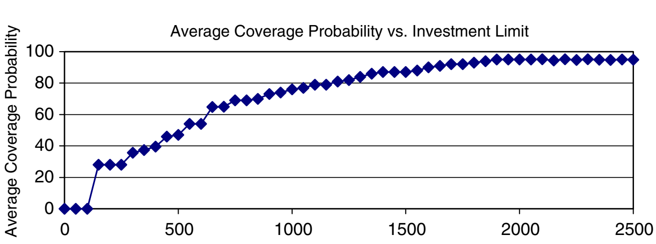


Fig. 11. Average detection probability versus investment limit for a region with 50×50 cells.

planar grid region with the objective of maximizing the detection probability within a given deployment cost. We showed this problem to be NP-complete, and then presented an approximate solution using a genetic algorithm for the case the sensor distributions are statistically independent. Computational results were presented when the target has uniform prior distribution and Gaussian approximation for sensor distributions, which showed that this solution performs favorably in solving the sensor deployment problem. This solution is applicable to more general

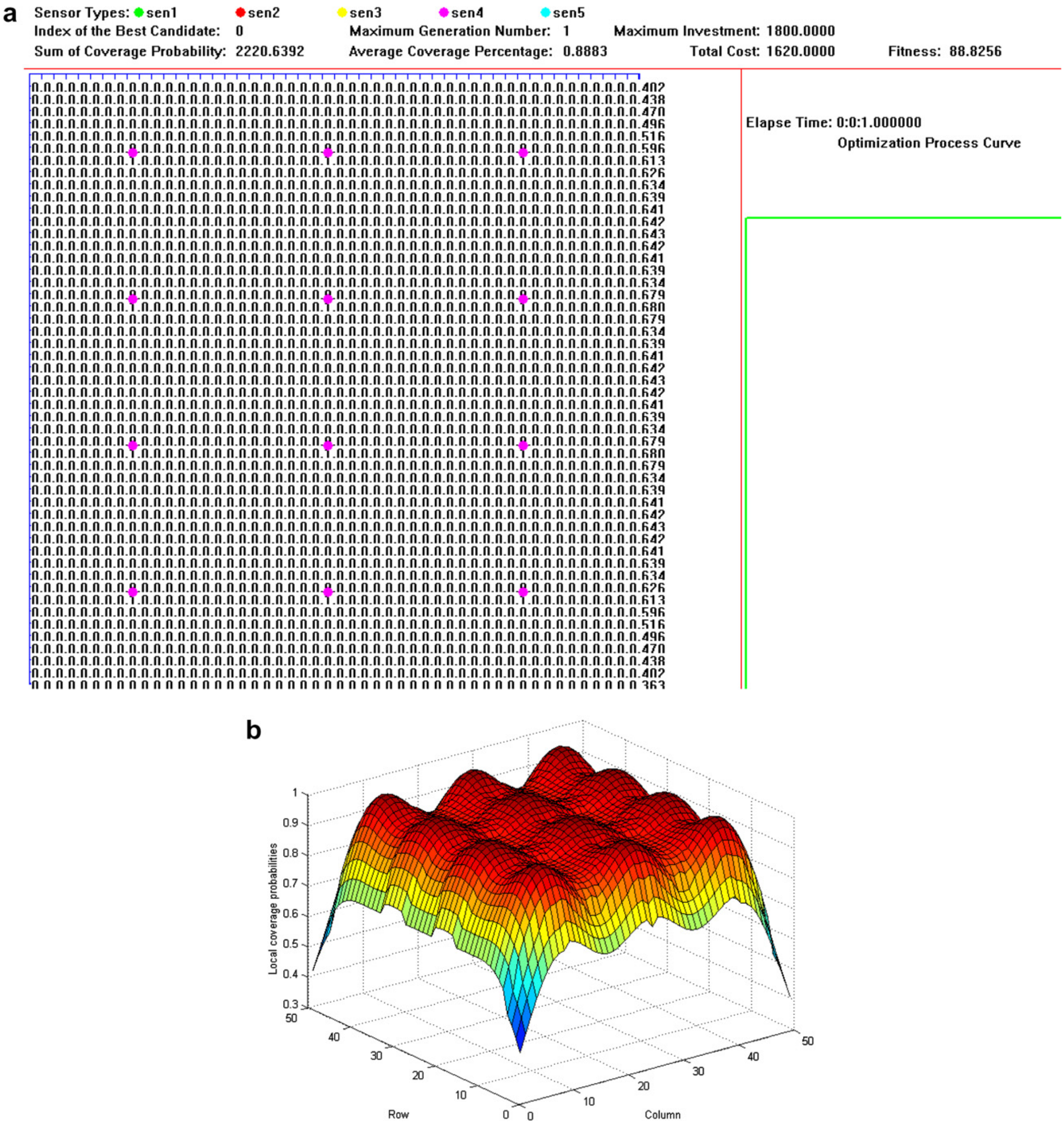


Fig. 12. (a) Graphical representation of deployment scheme. (b): 3-D display of the local coverage probabilities for a surveillance region with 50×50 cells based on uniform placement.

cases in which the target's *a-priori* distribution is not uniform and the sensor distributions are more complicated but computable. In general, the computational cost of such extensions would be correspondingly higher.

There are a number of avenues for further research. First, it would be interesting to see if analytical performance bounds can be placed on the solution computed by our method. Also, extensions of the proposed method when the statistical independence is not satisfied would be applicable to larger classes of sensor deployment problems. The challenge in this case is to ensure low computational complexity by utilizing the domain-specific knowledge of the sensors. In particular, the simple incremental formula computing the overlapping detection probability is no longer valid and in the worst-case this computation may have an exponential complexity for arbitrary distributions. From an algorithmic perspective, polynomial-time approximations to the sensor deployment problem that are guaranteed to be provably close-to-optimal will be of future interest.

Table 3
Comparison of computational performances of GA and UP

Case no.	Surveillance region size	No. of sensor types	Max investment limit	Genetic algorithm		Uniform placement	
				Total cost	Ave. detection probability (%)	Total cost	Ave. detection probability (%)
1	50×50	5	1800	1796	94.52	1620	88.83
2	100×100	5	2100	2081	93.03	1920	84.41
3	120×120	7	2250	2226	94.18	2160	87.44
4	150×150	7	2350	2340	93.12	2187	85.96
5	200×200	8	2600	2587	93.97	2430	88.61
6	300×300	5	3900	3861	93.64	3630	88.75
7	600×600	6	4600	4598	96.84	4400	87.69
8	750×750	5	6000	5995	93.81	5670	87.40
9	900×900	8	9000	8949	95.16	8993	86.61
10	1000×1000	9	9700	9698	93.70	9680	88.58

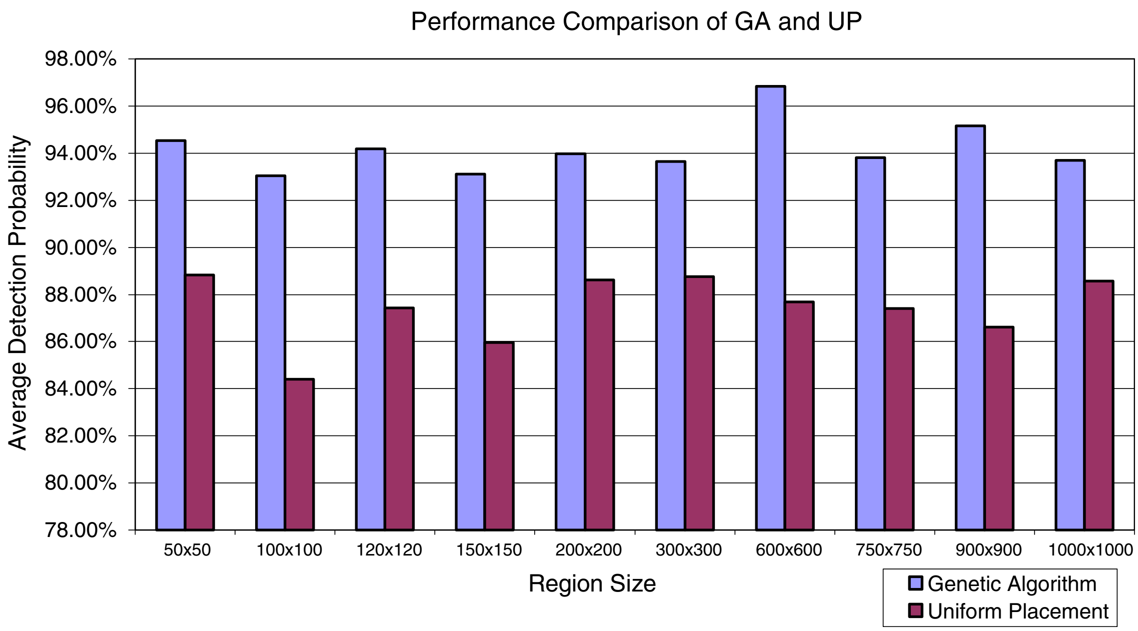


Fig. 13. Performance comparison of GA and UP for a region with 50×50 cells.

Acknowledgements

This research is supported in part by Defense Advanced Research Projects Agency under Grant No. N66001-00-C-8046 and by Office of Naval Research under Grant No. N000140110712. This research is also sponsored by Ballistic Missile Defense Organization Under MIPR No. 0100568954 and by the Engineering Research Program of the Office of Science, U.S. Department of Energy, managed by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725.

References

[1] K. Chakrabarty, S.S. Iyengar, H. Qi, E. Cho, Grid coverage for surveillance and target location in DSNs, IEEE Transactions on Computers 51 (12) (2002) 1448–1453.

[2] K. Chakrabarty, S.S. Iyengar, H. Qi, E.C. Cho, Coding theory framework for target location in distributed sensor networks, in: Proceedings of the IEEE International Conference on Information Technology: Coding and Computing, Las Vegas, Nevada, pp. 157, 2001.

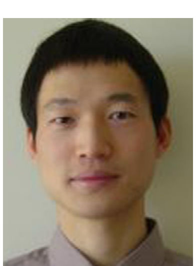
[3] N. Megiddo, K.J. Supowit, On the complexity of some common geometric location problems, SIAM J. Comput. 13 (1984) 182–196.

[4] S. Arora, P. Raghavan, S. Rao, Approximation schemes for Euclidean *k*-median and related problems, in: Proceedings of the 30th Annual ACM Symposium on Theory of Computing, pp. 106–113, 1998.

[5] S. Kolliopoulos, S. Rao, A nearly linear-time approximation scheme for the Euclidean *k*-median problem, in: Proceedings of the 7th European Symposium on Algorithms, pp. 378–389, 1999.

[6] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, A.Y. Wu, A local search approximation algorithm for *k*-means clustering, in: Proceedings of the 18th Annual ACM Symposium on Computational Geometry, pp. 10–18, 2002.

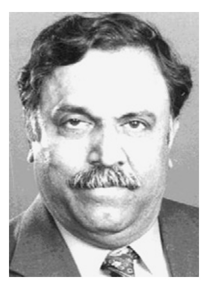
[7] N. Bulusu, J. Heidemann, D. Estrin, Adaptive beacon placement, in: Proceedings of ICDCS-21, Phoenix, Arizona, USA, 2001.
[8] N. Bulusu, D. Estrin, Scalable ad hoc deployable RF-based localization, Proceedings of the Grace Hopper Celebration of Women in Computing Conference, Vancouver, British Columbia, Canada, 2002.
[9] L. Guibas, D. Lin, J.C. Latombe, S. LaValle, R. Motwani, Visibility-based pursuit evasion in a polygonal environment, International Journal of Computational Geometry Applications (2000).
[10] S. Meguerdichian, F. Koushanfar, M. Potkonjak, M. Srivastava, Coverage problems in wireless ad hoc sensor networks, in: Proceedings of IEEE Infocom 2001, Anchorage, Alaska, pp. 22–26, 2001.
[11] S.S. Iyengar, L. Prasad, Hla Min, Advances in Distributed Sensor Integration: Application and Theory, Prentice Hall Publication Co., New Jersey, 1995, 07458, pp. 130–134.
[12] R.R. Brooks, S.S. Iyengar, Multi Sensor Fusion: Fundamentals and Applications with Software, Prentice Hall Publication Co., New Jersey, 1997, 07458.
[13] C.L. Liu, Introduction to Combinatorial Mathematics, McGraw-Hill, 1968.
[14] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Co, 1979.
[15] A compendium of NP optimization problems, <http://www.nada.kth.se/~viggo/problemlist/compendium.html/>.
[16] <http://www.nosc.mil/robots/research/manyrobo/detsensors.html/>.
[17] <http://www.nlectc.org/perimetr/full2.htm/>.
[18] J.H. Holland, Adaptation in Nature and Artificial Systems, The University of Michigan Press, 1975, reprinted by MIT Press, 1992.
[19] D.A. Coley, An Introduction to Genetic Algorithms for Scientists and Engineers, World Scientific, 1999.
[20] P.H. Winston, Artificial Intelligence, third ed., Addison-Wesley Publishing Company, 1993.
[21] D.E. Goldberg, Genetic Algorithms in Search, Optimization & Machine Learning, Addison-Wesley Publishing, 1989.
[22] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, K.K. Saluja, Sensor deployment strategy for target detection, in: Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications, 2002.
[23] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman Publishing, 1979.
[24] S. Martinez, F. Bullo, Optimal sensor placement and motion coordination for target tracking, Automatica 42 (4) (2006) 661–668.
[25] D. Ganesan, R. Cristescu, B. Beferull-Lozano, ACM Transactions on Sensor Networks 2 (2) (2006).
[26] G. Wang, G. Cao, T. La Porta, Movement-assisted sensor deployment, INFOCOM (2004).
[27] Y. Mao, M. Wu, Coordinated Sensor Deployment for Improving Secure Communications and Sensing Coverage, SASN'05, Alexandria, Virginia, USA, November 7, 2005.



Qishi Wu received the B.S. degree in remote sensing and GIS from Zhejiang University, People's Republic of China in 1995, the M.S. degree in geomatics from Purdue University in 2000, and the Ph.D. degree in computer science from Louisiana State University in 2003. He worked as a research fellow at Oak Ridge National Laboratory during 2003–2006. He is currently an assistant professor in the Department of Computer Science at University Of Memphis. His research interests include computer networks, remote visualization, distributed sensor networks, algorithms, and artificial intelligence. He is a member of the IEEE and IEEE Computer Society.

Nageswara S.V. Rao received B.Tech in Electronics Engineering from Regional Engineering College, Warangal, India in 1982, M.E. from School of Automation, Indian Institute of Science, Bangalore in 1984 and Ph.D. in Computer Science from Louisiana State University in 1988. He is currently a UT-Battelle Corporate Fellow at Oak Ridge National Laboratory where he joined in 1993. His research interests include network transport dynamics, statistical approaches to transport control, and information and sensor fusion.

Xiaojiang Du is an assistant professor in Department of Computer Science, North Dakota State University. Dr. Du received his B.E. degree from Tsinghua University, Beijing, China in 1996, and his M.S. and Ph.D. degrees from University of Maryland, College Park in 2002 and 2003, respectively, all in Electrical Engineering. His research interests are wireless sensor networks, mobile ad hoc networks, network security and network management. Dr. Du is an associate editor of Wiley Journal on Wireless Communication & Mobile Computing, and International Journal of Sensor Networks. He is (was) the chair of Computer and Network Security Symposium of ACM International Wireless Communication and Mobile Computing Conference 2007 (2006). He is (was) a TPC member for many major IEEE conferences such as INFOCOM, ICC, GLOBECOM, WCNC, IM, and NOMS.



Dr. S.S. Iyengar is the Chairman and Roy Paul Daniels Chaired Professor of Computer Science and is also Satish Dhawan Chaired Professor at the Indian Institute of Science. He has been awarded the Distinguished Alumnus Award at the Indian Institute of Science in March 2003. He is the Distinguished Research Master Award winning Professor of the Computer Science Department at Louisiana State University. He has been involved with research in high-performance algorithms, data structures, sensor fusion, data mining, and intelligent systems since receiving his Ph.D. degree (in 1974 at Mississippi State University) and his M.S. from the Indian Institute of Science (1970). He has directed over 30 Ph.D. candidates, many of whom are faculty at major universities worldwide or scientists or engineers at national labs/industry around the world. He has served as a principal investigator on research projects supported by the Office of Naval Research (ONR), Defense Advanced Research Project Agency (DARPA), the National Aeronautics and Space Administration (NASA), the National Science Foundation (NSF), California Institute of Technology's Jet Propulsion Laboratory (JPL), the Department of Navy-NORDA, the Department of Energy (DOE), LEQSF-Board of Regents, and the U.S. Army Research Office. His publications include 13 books (authored or coauthored textbooks; Prentice-Hall, CRC Press, IEEE Computer Society Press, John Wiley & Sons, etc.) and over 280 research papers in refereed journals and conference in areas of high-performance parallel and distributed algorithms and data structures for image processing and pattern recognition, and distributed data mining algorithms for biological databases. His books have been used in Berkeley, Purdue, University of Southern California, University of New Mexico, etc. He was a visiting professor at the Jet Propulsion Laboratory-Cal. Tech, Oak Ridge National Laboratory, the Indian Institute of Science, and at the University of Paris. Dr. Iyengar has served as an associate editor for the Institute of Electrical and Electronics Engineers and as guest editor for the *IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions and SMC, IEEE Transactions on Software Engineering,* *Journal of Theoretical Computer Science, Journal of Computer and Electrical Engineering, Journal of the Franklin Institute, Journal of American Society of Information Science, International Journal of High Performance Computing Applications,* etc. He has served on review panel committees for NSF, NASA, DOE-ORNL, the U.S. Army Research Office, etc. He has been on the prestigious National Institute of Health-NLM Review Committee, in the area of Medical Informatics for 4 years. Dr. Iyengar is a series editor for *Neuro-Computing of Complex Systems* for CRC Press.

In 1998 Dr. Iyengar was the winner of the IEEE Computer Society Technical Achievement Award for Outstanding Contributions to Data Structures and Algorithms in Image Processing and Sensor Fusion Problems. This is the most prestigious research award from IEEE Computer Society. Dr. Iyengar was awarded the LSU Distinguished Faculty Award for Excellence in Research, the Hub Cotton Award for Faculty Excellence, and the LSU Tiger Athletic Foundation Teaching Award in 1996. He has been a consultant to several industrial and government organizations (JPL, NASA etc.). In 1999, Professor Iyengar won the most prestigious research award titled Distinguished Research Award and a university medal for his research contributions in optimal algorithms for sensor fusion/image processing.

He is also a Fellow of Association of Computing Machinery (ACM), a Fellow of the IEEE, a Fellow of American Association of Advancement of Science (AAAS), a Williams Evans Fellow, IEEE Distinguished Visitor, etc. He has been a Distinguished Lecturer for the IEEE Computer Society for the years 1995–1998, an ACM National Lecturer (1986–1992), and Member of the Distinguished SIAM Lecturer Program (2000–2002). He was a member of the ACM accreditation committee for 2000–2002. He is a member of the European Academy of Sciences and the New York Academy of Sciences. He has been the Program Chairman for many national/international conferences. He has given over 50 plenary talks and keynote lectures at numerous national and international conferences.

Vijay K. Vaishnavi (SM '89-F'01) received the BE degree (with distinction) in electrical engineering from Jammu and Kashmir University and the M.Tech and Ph.D. degrees in electrical engineering (with major in computer science) from the Indian Institute of Technology, Kanpur. He has also done postdoctoral work in computer science for two years at McMaster University, Canada. Dr. Vaishnavi is currently Professor of Computer Information Systems and Professor of Computer Science at Georgia State University. His current areas of research interest include inter-organizational systems (semantic interoperability, directory services, web-based virtual communities, coordination, security), software development (object-oriented metrics, software specifications and their maturity, object-oriented modeling and design), and data structures and algorithms (multi-sensor networks and fusion). The National Science Foundation and private organizations, including IBM, Nortel, and AT&T, have supported his research. He has authored numerous papers in these and related areas. His papers have appeared in *IEEE Transactions on Software Engineering, IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Computers, SIAM Journal on Computing, Journal of Algorithms,* and several other major international journals and conference proceedings. Dr. Vaishnavi is an IEEE Fellow. He is also member of the IEEE Computer Society, the Association for Computing Machinery (ACM), and the Association for Information Systems (AIS).