A Probabilistic Model for the Deployment of Sensors

Brian Carter and Rammohan Ragade
Computer Engineering and Computer Science
University of Louisville
Louisville, KY

***Abstract*—Coverage is an important optimization objective in sensor deployment problems. This paper addresses the issue of covering a set of target points in an area with a finite set of sensors. A probabilistic model is proposed which takes in account the detection probabilities of the sensing devices which may decay with distance, environmental conditions, and hardware configuration. The objective is to deploy sensors so that the distribution of the sensors meets the probability of detection requirements while minimizing costs. The expected points to cover and the deployment points are assumed to be stationary and known a priori. A probabilistic coverage matrix is defined and the deployment is optimized using a genetic algorithm. Our experimental results verify that the proposed probabilistic sensor deployment model finds more efficient solutions requiring fewer sensors compared to other deployment schemes.**

I. Introduction

The problem of determining the minimum number of objects to deploy in a given area to meet given objectives, has been studied in many fields. In literature, the art gallery problem is one of the most researched and studied problems in this domain [1]. The foundation of the problem is to use the minimum number of guards to watch a polygon area. Each guard can watch any point which is in its line of sight.

In this paper, we are addressing the sensor deployment problem (SDP). The SDP has many of the same elements as found in the art gallery problem. First, the area to monitor is defined by a set of target points to watch. Sensors replace the concept of guards. Each sensor can cover any target point within its sensing range. Sensors can be deployed at defined deployment points in the area. The sensor deployment problem can be defined as minimizing the cost required to cover the points in the area.

Our previously work [2] defined a model to study the SDP where the objective was to minimize costs subject to covering all target points using a binary detection model. It has been found in field studies, that the detection probability of a sensor will decay with distance, environmental conditions, hardware configurations, and other problem specific attributes. In this paper, we propose to extend our binary model to a probabilistic coverage model. The coverage matrix is updated to take into consideration the coverage probabilities. There are extensive studies, which focus on how to determine the probabilities given a specific type of sensor, environment, layout, and field of views [3]. It is not the intent of this paper to define such probability detection functions, but to define a model which can utilize these studies in a sensor deployment model.

The objective of our research described in this paper is to develop a model that can provide insight into sensor deployment problems. Because this topic has not been the subject of extensive research [4], the paper starts with a general discussion of related work. In section III, we present the primitives of the model and our probabilistic techniques. Section IV formulates the SDP as an optimization problem. Our model and solution is defined in section V and VI. The paper concludes with a case study implementing our model.

II. Related Work

There has been relatively little research performed in the area of optimizing sensor deployment [4]. The standard deployment for sensor networks is grid-based placement or perimeter placement. Placing the sensors in a grid, ensures that all points in the area are covered. However, this dispersement requires a large number of sensors which results in a high cost for the solution. Grid based deployments are ineffective in problems with obstacles, resulting in an over deployment of equipment. Another common technique, is perimeter based placement. This technique ensures that any object coming within a certain distance must pass through at least one sensor, however the perimeter radius must be relatively small to avoid circumvention by intersection just within the perimeter.

Given an area to monitor, determining the set of sensors to deploy is a difficult problem and has been shown to be NP-complete [5]. This eliminates many proposed deterministic models because of the amount of time required to find a solution. Previous research reduced the dimensions of this problem by limiting the types of sensors used and by assuming a simple two dimensional plane [6]. While this approach is effective, it is often not practical. Each year, sensors and devices continue to advance. Due to the expense and need for replacement, heterogenous sensor networks are typical. With intrusion detection, border monitoring, and homeland security, the problem of height and elevation must be calculated into the model. Even within proximity of the sensors radius, it has been shown that such deployment methods do not work in most environments with obstacles, hills, and rivers [6].

Studies often propose the sensor deployment problem using a binary detection model. While this is appropriate for defining the foundation of research, it is not practical in actual deployments. Other studies assume a minimal number of isotropic sensors to cover a set of points [7], [8]. Research has shown that many sensing devices have a specific detection pattern and have proposed advanced techniques for field-of-view problems [9], [10]. Hardware studies have shown that the detection capabilities of sensors have a probabilistic pattern [3].

We propose a more realistic approach for the sensor deployment problem. Our model allows for the high coverage of grid based deployments while minimizing the cost as found in perimeter deployment techniques. A genetic algorithm is used to reduce the time to find a sufficient solution compared to deterministic schemes. A three dimensional area is used for defining the target and deployment points, allowing for longitude, latitude, and altitude calculations in the detection algorithm. We extend the model to take into consideration the detection probability of each sensor and show how this additional information finds more efficient solutions when compared to binary models.

III. Model Definitions

A. Primitives

We are given a three dimensional area A to be deployed with sensors. Within Area A, we define a target location t as a three dimensional grid point $< x, y, z >$ to be modeled. The Area A has i number of target points t_i defining the set T. Each target point t_i should be covered by a subset of sensors in the network.

Within Area A, we define a deployment location o as a three dimensional grid point $< x, y, z >$ where sensors may be deployed. We constrain the deployment of sensor to the points in the set O consisting of l number of deployment points o_l.

We are given a set of sensors types S, composed of k types s_k. Each sensor of a given type has the same sensing abilities such as sensing radius r_{s_k} and shape of sensing field for a given sensing function.

A sensor type s_k may be deployed to a deployment location o_l. A deployment tuple d_j is defined for the specific set of values. Deployment tuple d_j=$<\ s_k, o_l\ >$ defines the deployment of sensor type k at deployment location l. The deployment tuple can be extended to define other attributes such as orientations, power settings, or factors which would affect the sensing ability of the device at that location [2]. For simplification, we define the tuple as pairing of sensor type and deployment location.

We define a coverage matrix a_{ij}. Each row i is a target location t_i. Each column j is defined by a deployment tuple d_j. Given $a_{ij} = 1$, deployment tuple d_j will always detect an event at location t_i. A value of 0 indicates that the deployment tuple does not cover the target point. We extend our coverage matrix, to include detection probabilities, where the given value of the cell a_{ij} represents the probability that the deployment tuple will cover events at the target point. The value of a cell in the coverage matrix is between 0 and 1.

B. Detection Probability

Depending on the type of sensors used in the network, the detection probability of a sensor may decay with distance, environmental conditions, and hardware configuration. Simplistic models define a binary model where the detection probability p is typically calculated using the Euclidean distance where $dist(t_i, d_j)$ is the distance between the deployment tuple point o_l and target location t_i. If this distance is less than the sensing radius for the type of sensor r_{s_k}, then it is covered as shown in equation 1.

$$p(a_{ij}) = \begin{cases} 1 & \text{if } dist(t_i, d_j) \leq r_{s_k} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Research has shown that the quality of a sensor and its detection probability may decay with distance[11], [12], [13]. Although the binary detection model simplifies the analysis, it may not be realistic in many cases. In [6], the detection probability is defined as:

$$p(a_{ij}) = \begin{cases} e^{-\beta dist(t_i, d_j)} & \text{if } dist(t_i, d_j) \leq r_{s_k} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The parameter β is a parameter related to the physical characteristics of the sensor, which can be obtained from field experiments for the specific sensor [13]. In our heterogeneous environment, a β parameter is defined for each sensor type s_k. To revert back to a binary model, set β to 0.

The algorithms and equations to calculate $p(a_{ij})$ is determined by the problem and complexity required. Studies have proposed complex, multi-part algorithms for determining the detection for passive infrared sensors, cameras, and mobile sensors [14]. As shown, the detection probability can be simplistic or extremely complex. Equation 2 will be used for the remainder of the paper.

C. Probability Of Detection

Given a sensor type and a set of points, the probability that the sensor will detect an event at a given point can be calculated. Having many sensors in a solution, a target point may be covered by one or more sensors. In defining the problem, the objective is to minimize costs subject to all points covered above a given coverage threshold. With two sensors covering a point, a method needs to be defined for calculating the probability of detecting an event at a given target point. It is assumed that the events are independent, the occurrence or nonoccurrence of deployment tuple d_1 detecting the event does not affect the occurrence or nonoccurrence of d_2 to detect the event.

Our first attempt was to study the additive law of probability. The General Rule of Addition states when two or more events will happen at the same time, and the events are not mutually exclusive, then the probability addition rule can be applied:

$$p(a_{i1} \text{ or } a_{i2}) = p(a_{i1}) + p(a_{i2}) - p(a_{i1}a_{i2}) \tag{3}$$

The probability of either d_1 or d_2 detecting the event at target point t_i is the probability of a_{i1} plus the probability of a_{i2} minus the probability that both a_{i1} and a_{i2} will detect the event. Consider that a given target point in the area is covered by two sensors with different detection probabilities. Assume deployment tuple d_1 covers the target point t_i with a detection probability of 0.50 and deployment tuple d_2 covers with 0.75. Using equation 3, the probability of either d_1 or d_2 detecting the event is 0.875. The probability of both sensors missing the event is $(1 - p(a_{i1} \text{ or } a_{i2})) = (1 - 0.875) = 0.125$.

As the number of deployment tuples coving a target point increases, the additive law becomes complex to calculate as it is extended. We studied the special rule of multiplication for probability. When two or more events will happen at the same time and the events are independent then the special rule of multiplication law is used to find the joint probability. To apply this rule, we examined the problem from the miss probability perspective [11]. To calculate the probability that both d_1 and d_2 would miss the event (not cover) at target point t_i, the multiplication law of probability is used to define the miss probability (equation 4):

$$p_m = \prod_{a_{ij} \in D} (1 - p(a_{ij})), \text{for a given } a_i \tag{4}$$

In our case, we are looking for probability of detection. By subtracting 1 from equation 4, we derive the probability of detection [12] (equation 5):

$$p_d = 1 - \prod_{a_{ij} \in D} (1 - p(a_{ij})), \text{for a given } a_i \tag{5}$$

Through a simple exercise, one could show that equation 3 and equation 5 are equivalent. In review of our previous example: $1-[(1-p_1)\times(1-p_2)] = 1-[(1-0.5)\times(1-0.75)] = 1-0.125 = 0.875$. We use equation 5 in our model to simplify calculations.

IV. Problem Definition

The sensor deployment problem is the problem of covering the rows of a m-row, n-column, probabilistic coverage matrix (a_{ij}) by a subset of the columns at minimal cost. Defining $x_j = 1$ if column j is used in any rows of the solution and $x_j = 0$ otherwise. The cost of deploying column j is defined as c_j. The sensor deployment problem can be formulated as an optimization problem:

$$Minimise \qquad \sum_{j=1}^{n} c_j x_j \tag{6}$$

$$\text{Subject to} \qquad \sum_{j=1}^{n} a_{ij} x_j \geq 1, \quad \text{i=1,...,m} \tag{7}$$

$$p_d(a_i) \geq \alpha, \quad \text{i=1,...,m} \tag{8}$$

$$x_j \in \{0, 1\}, \quad \text{j=1,...,n} \tag{9}$$

Equation (6) defines the optimization problem where the objective is to minimize cost. The solution is constrained in equation (7), which requires that each row is covered by at least one column. Equation (8) constrains the solution further by requiring that probability of detection for each row is greater than or equal to a given probability coverage threshold α. Equation (9) is the integrality constraint.

V. Problem Modeling

To solve the optimization problem, a number of algorithms have been presented in literature [15], [16], [17]. It has been shown that optimal algorithms are not applicable due to the runtime required to find a solution [16]. Greedy heuristics have been proposed to lower the runtime for homogenous sensor deployments [8], [18]. It has been shown that evolutionary algorithms provide near-optimal results for heterogenous SDPs using GA and simulated annealing algorithms [5], [15], [16], [17].

We propose a probabilistic sensor deployment model (SDP-Model) which addresses the drawbacks of previous models. Any sensor network that has a longevity of more than a single deployment will involve the use of different sensing equipment. We have experienced this in our lab, where certain intrusion detection sensors have been discontinued. Given this new set, the collection of sensors have varying abilities.

The previously proposed homogeneous models are no longer applicable [19]. Second, the ability of a sensor to monitor a given target point may be dependent on the hardware settings, the deployment position, and environmental conditions. Also, objects in the target area may block or limit the sensing ability.

The previously perfect binary detection models are also not applicable [7], [8]. Third, previously proposed models are application-dependent to address a specific problem [9], [10]. The method used to determine if a target point is covered is tightly coupled with the deployment model. With so many techniques proposed for determining coverage, a decoupled method is required. Lastly, research has shown that the detection probability of a sensing device may vary dependent on the specific attributes of the deployment problem [20] which required us to revisit our binary detection model [2].

In our research, we found that the methods used to determine coverage are application dependent. Determining if a target point is covered by a given sensor is dependent on the hardware, the deployment point, orientation, environmental factors, and other attributes. We approach this by decoupling the elements from the SDP-Model by defining a probabilistic coverage matrix. Using the probabilistic coverage matrix a_{ij}, we define each row i as a three-dimensional grid point t_i which must be covered by at least one sensor and must be covered above a given probability of detection threshold α. Each column j is defined by a deployment-tuple d_j, which includes sensor type and deployment point. Additional attributes may be added to the deployment-tuple. Given $a_{i,j} = 0.90$, indicates target point t_i (row) is covered by deployment-tuple d_j (column) at a probability of detection of 90 percent. Where d_j defines the inclusion of a specific sensor type which is deployed at a given point.

To establish the coverage matrix, the first step is to define the points to be covered. A row is added to the matrix corresponding to each target point. The next step is to define the deployment-tuples. Given the available combinations of $<$ *sensor type*, *deployment point* $>$, the candidate deployment-tuples are defined. A column is added for each deployment-tuple. The final step is to determine the set of points covered by each deployment-tuple using a coverage determination method. Without loss of generality, we shall assume that the coverage determination method is defined using a pre-established approach [8], [9], [16], [10].

The coverage determination method used is specific for each problem. For example, a simplistic SDP may use the Euclidean distance for coverage determination where a complex SDP may use a FOV metric. Independent of the method used, the coverage matrix a_{ij} is formulated as shown in Table I. The target points t_i require coverage by selecting the deployment-tuples d_j. In the example, deployment-tuple d_1 covers points $\{t_1, t_2, t_3\}$ at detection probabilities $\{0.90, 0.95, 0.90\}$.

Our approach for defining the coverage matrix a_{ij} has many benefits. The model remains consistent for simple and complex problems by decoupling the model from the coverage determination method. The only task required to optimize a SDP is to provide the coverage matrix and algorithm parameters to the model.

The model implements a genetic algorithm to optimize the problem as shown in figure 1. The output of the model is a subset of deployment-tuples which cover all target points at a minimal cost. As shown in the next section, we devise a suitable representation scheme of the coverage matrix a_{ij} for the GA approach.

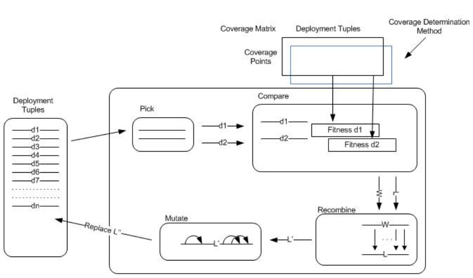

Fig. 1. Sensor Deployment Model (SDP-Model).

VI. Solution To The Sensor Deployment Problem

A genetic algorithm is often defined as an "intelligent" probabilistic search algorithm [15]. Holland [21] originally established the theoretical foundations using an evolutionary process. The principles of natural selection and "survival of the fittest" are used to reproduce children which are more "fit" for their environment. It has been shown that genetic algorithms can be applied to a variety of combinatorial optimization problems [22].

With the success of GAs, many adaptations have been presented. For this paper, the Microbial Genetic Algorithm [23] is applied by defining the following five components [22]:

- a genetic representation for potential solutions to the problem,
- a way to create an initial population of potential solutions,
- an evaluation function that rates solutions in terms of their "fitness",
- genetic operators (crossover, mutation) that alter the composition of children,
- values for various parameters used in the algorithm.

First, the chromosome is defined. Reviewing Table I, the task is to select a subset of the deployment-vectors which cover all points above a given probability of coverage threshold α subject to minimizing the cost. The chromosome can be represented as a sequence of 0's and 1's. The chromosome (101000) would represent the use of deployment-tuples $\{d_1, d_3\}$. A "0" in the second slot of the chromosome indicates the exclusion of d_2. As shown in Table II, the chromosome covers all target points with a minimum p_d of 0.70 and an average p_d of 0.842. The length of the vector is always the number of deployment-tuples n.

The next task is to determine the initial population P. A population is a set of potential solutions (chromosomes). The number of chromosomes included in the population is determined by the population size parameter POP. All bits in the chromosome are initialized randomly.

After creating an initial population, each chromosome is then evaluated and assigned a fitness value. When applying any GA, the difficult task is to define the fitness function. In this paper, the fitness function provides a measure of performance with respect to a particular set of parameters. The fitness function $f(v)$ for chromosome v is defined as the summation of cost for all sensors included in the solution plus a weighted penalty for not covering target points above a given probability of detection threshold α (Equation 10). The value $(m - coverage)$ represents the number of target points not covered at the required level. The parameter m represents the number of rows and $coverage$ represents the number of rows that satisfies the threshold. The variable w represents the weight of the penalty. Using a weighted penalty allows us to relax the constraints (Equations 7, 8) in our fitness function to allow better fitting chromosomes to evolve. In our fitness function, a lower fitness value indicates a better fit (Equation 10).

$$f(x) = \sum_{j=1}^{n} c_j x_j + w(m - coverage) \qquad (10)$$

For the genetic operators (crossover, mutation), the methods defined by Harvey [23] are used. For the SDP, the following parameters are defined:

- coverage matrix a_{ij},
- number of tournaments $TOURS$,
- population size POP,
- probability of crossover p_c,
- probability of mutation p_m.

The GA is adapted for the SDP as shown in Algorithm 1. The function rnd returns a random number between 0.0 and 1.0 and $rnd(1..POP)$ returns an integer between 1 and POP. Lines 4-5 of the algorithm perform the selection of two random chromosomes to evaluate. Their evaluation is performed in Lines 6-10, using the fitness function defined in equation (10). The crossover is performed in lines 11-13. The mutation of the losing chromosome v_l is performed in lines 14-16. The algorithm completes after a number of iterations ($TOURS$). The final output of the algorithm is the best solution chromosome v_{max} which has the best fit and satisfies all the constraints. Evolving our model to include probabilities only affected the fitness function retaining the original GA algorithm:

Algorithm 1 GA for Non-Isotropic SDPs

```
1: input: a_ij
2: initialize P = {v_i : i = 1..POP}
3: for i = 1 to TOURS do
4:    a=rnd(1..POP)
5:    do b=rnd(1..POP) while (a==b)
6:    if f(v_a) > f(v_b) then
7:       w = a, l = b
8:    else
9:       w = b, l = a;
10:   end if
11:   for j = 1 to n do
12:      if(rnd < p_c) v_l[j] = v_w[j]
13:   end for
14:   for j = 1 to n do
15:      if(rnd < p_m) v_l[j] = 1 - v_l[j]
16:   end for
17: end for
18: output: v_max
```

VII. Experimental Analysis

The specific problem considered in our analysis was attempting to optimize sensor placement to detect intruders along a border using passive infrared motion sensors. The sensor readings were simulated using a probabilistic model where the probability of detection was directly proportional to distance. The environment was simulated using a small section of the United States and Mexico border along a given section of river.

For our initial case study, a small section was chosen to allow manual evaluation of the model. In this problem, only one type of sensor is available, an isotropic passive infrared sensor with a ninety degree field of vision. A two dimensional area with six target points and six deployment points is defined as shown in Figure 2. The six triangles are defined as the points that require coverage. The five circles represent the deployment points. The figure may represent a river or road, where sensors can be deployed only on the outer edges of the target area. By defining the SDP-Model with coverage points and deployment points, such a realistic case study can be evaluated.

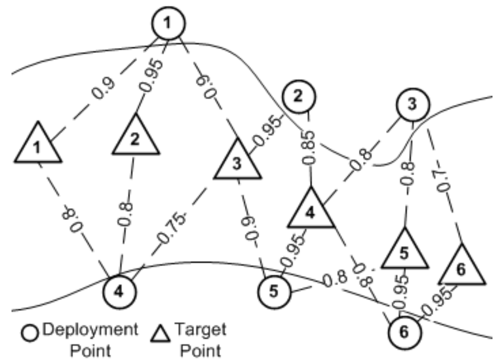

Fig. 2. Homeland Security Sensor Fence.

The problem is to select a subset of the deployment-tuples which minimize cost subject to covering all points above the given coverage threshold. Without loss of generality, we shall assume that the coverage determination method is defined using a pre-established field of view approach [9], [10]. To minimize the number of deployment-tuples to display, it is assumed that the sensors are fixed and have only one orientation.

Using the field of view for the coverage determination method, the matrix a_{ij} can be generated as shown in Table I.

TABLE I
Case Study Coverage Matrix a_{ij}

	d_1	d_2	d_3	d_4	d_5	d_6
t_1	0.90	0.00	0.00	0.80	0.00	0.00
t_2	0.95	0.00	0.00	0.80	0.00	0.00
t_3	0.90	0.95	0.00	0.75	0.90	0.00
t_4	0.00	0.85	0.80	0.00	0.95	0.80
t_5	0.00	0.00	0.80	0.00	0.80	0.95
t_6	0.00	0.00	0.70	0.00	0.00	0.95

For this problem, the optimal solution derived from the algorithm can be confirmed deterministically by generating all the possible permutations. We used the permutations to check the results from our simulation program which implements the SDP-Model. If we set the threshold α to 0.70, there are four chromosomes which satisfy the constraints using only two sensor to cover all target points (Table II).

TABLE II
CASE STUDY EVALUATION

chromosome	101000	100001	001100	000101
t_1	0.90	0.90	0.80	0.80
t_2	0.95	0.95	0.80	0.80
t_3	0.90	0.90	0.75	0.75
t_4	0.80	0.80	0.80	0.80
t_5	0.80	0.95	0.80	0.95
t_6	0.70	0.95	0.70	0.95
$minp_d$	0.70	0.80	0.70	0.75
$avep_d$	0.842	0.908	0.775	0.842

Reviewing the minimum and average probability of detection for the candidate chromosomes, chromosome 100001 is the optimal solution using two sensors. If we examine chromosome 101001, applying equation 5 would result in an a minimum p_d of 0.90 and an average p_d of 0.948 at the cost of an additional sensor. By adding detection probabilities to our model, it is clear to see the benefits gained over the binary model. The experiment shows that it is beneficial to include probabilities in our analysis to gain additional information to separate apparently equal chromosomes.

VIII. CONCLUSION

A modeling framework for the sensor deployment problem was presented. In our model, the problem is formulated as an optimization problem with the objective to minimize cost while covering all points above a given probability of detection coverage threshold. A probabilistic coverage matrix was defined to decouple the coverage determination method from the model. This allows the framework to be used for a wide range of sensor deployment problems. A GA approach was presented to solve the optimization problem. A homeland security sensor fence case study was given to illustrate the capabilities of the framework.

We defined the model in an attempt to address the diverse complexities found in the homeland security domain. In the case of an event, efficient and effective deployments of the available sensing and surveillance equipment must be determined. This evaluation may be recalculated at regular intervals based on failures or changes in the event.

Such NP-complete problems require a flexible deployment model which can be applied and adjusted based on parameters. Past problem specific models would require the development of new models, which may not be possible due to time constraints or availability of resources. Due to the number of dimensions in sensor deployment problems, deterministic schemes are not practical due to run time requirements. As shown in this paper, including detection probabilities provides valuable information which can not be derived from binary models.

The model presented in this paper address the diverse complexities through the use of parameters. Most notable is the definition of a probabilistic coverage matrix. The matrix provides the flexibility to include any combination of sensor types, deployment points, and other attributes in the model. The model determines a sensor deployment where costs are minimized and all points are covered above a given threshold.

The case studies demonstrate very promising results. Extending our SDP model to include probabilities provided very valuable information allowing for more detailed analysis of candidate solutions. We are currently finalizing the analysis on a large case study which shows promising results which we will include in a future paper.

REFERENCES

[1] J. O'Rourke, Art Gallery Theorems and Algorithms, Oxford University Press, NJ, 1987.

[2] B. Carter, R. Ragade, An extensible model for the deployment of non-isotropic sensors, IEEE Sensors Applications Symposium (2008) 22 – 25.

[3] H. Hefeeda, M. Ahmadi, A probabilistic coverage protocol for wireless sensor networks, IEEE International Conference on Network Protocols (2007) 41–50.

[4] S. Barrett, Optimizing sensor placement for intruder detection with genetic algorithms, IEEE Intelligence and Security Informatics (2007) 185–188.

[5] Q. Wu, N. Rao, X. Du, S. Iyengar, V. Vaishnavi, On efficient deployment of sensors on planar grid, Elsevier's Computer Communications (2007).

[6] Y.-C. W. C.-C. H. Y.-C. Tseng, Efficient placement and dispatch of sensors in a wireless sensor network, Mobile Computing, IEEE Transactions (2008) 262 – 274.

[7] S. Meguerdichian, F. Koushanfar, M. Potkonjak, M. B. Srivastava, Coverage problems in wireless ad-hoc sensor networks, IEEE Infocom (2001) 1380–1387.

[8] P. C. F.Y.S. Lin, A near-optimal sensor placement algorithm to achieve complete coverage-discrimination in sensor networks, Vol. 9, IEEE, 2005, pp. 43–45.

[9] J. Adriaens, S. Megerian, M. Potkonjak, Optimal worst-case coverage of directional field-of-view sensor networks, in: Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on, IEEE, 2006, pp. 336–345.

[10] J. Ai, A. A. Abouzeid, Coverage by directional sensors in randomly deployed wireless sensor networks, J. Comb. Optim 11 (1) (2006) 21–41.

[11] S. Dhillon, K. Chakrabarty, Sensor placement for effective coverage and surveillance in distributed sensor networks, IEEE Wireless Communications and Networking Conference (2003) 1609–1614.

[12] Y. Zou, K. Chakrabarty, Sensor deployment and target localization based on virtual forces, infocom (2003).

[13] J. Zhang, T. Yan, S. Son, Deployment strategies for differentiated detection in wireless sensor networks, Sensor and Ad Hoc Communications and Networks 1 (2006) 316–325.

[14] E. Hrster, R. Lienhart, On the optimal placement of multiple visual sensors, Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks (2006) 111 – 120.

[15] P. C. J. Beasley, A genetic algorithm for the set covering problem, European Journal of Operational Research 94 (1996) 392–404.

[16] R. Ramadan, H. El-Rewini, K. Abdelghany, Optimal and approximate approaches for deployment of heterogeneous sensing devices, EURASIP Journal on Wireless Communications and Networking (2007) 14 pages.

[17] S. J. Habib, Modeling and simulating coverage in sensor networks, Elsevier Computer Communications 30 (5) (2007) 1029–1035.

[18] A. Efrat, S. Har-Peled, J. Mitchell, Approximation algorithms for two optimal location problems in sensor networks, Broadband Networks, 2005 2nd International Conference on 1 (5) (2005) 714 – 723.

[19] X. Shen, J. Chen, Y. Sun, Grid scan: A simple and effective approach for coverage issue in wireless sensor networks, Communications 8 (3) (2006) 3480–3484.

[20] G. Holdrideg, S. Gould, P. Johnson, P. Lagasse, R. Morrison, K. Tresnak, Sensors for environmental observatories, NSF Technical Report.

[21] J. H. Holland, Adaption in Natural and Artificial Systems, University of Michigan Press, 1975.

[22] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, 3rd Edition, Springer-Verlag, 1996.

[23] I. Harvey, The microbial genetic algorithm, Evolutionary Computation (1996).