Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks

Krishnendu Chakrabarty, *Senior Member, IEEE,*
S. Sitharama Iyengar, *Fellow, IEEE,*
Hairong Qi, *Member, IEEE,* and
Eungchun Cho

Abstract—We present novel grid coverage strategies for effective surveillance and target location in distributed sensor networks. We represent the sensor field as a grid (two or three-dimensional) of points (coordinates) and use the term target location to refer to the problem of locating a target at a grid point at any instant in time. We first present an integer linear programming (ILP) solution for minimizing the cost of sensors for complete coverage of the sensor field. We solve the ILP model using a representative public-domain solver and present a divide-and-conquer approach for solving large problem instances. We then use the framework of identifying codes to determine sensor placement for unique target location. We provide coding-theoretic bounds on the number of sensors and present methods for determining their placement in the sensor field. We also show that grid-based sensor placement for single targets provides asymptotically complete (unambiguous) location of multiple targets in the grid.

Index Terms—Covering codes, identifying codes, integer linear programming, optimization, sensor density, sensor field.

———————————— ✦ ————————————

1 INTRODUCTION

DISTRIBUTED networks are essential for effective surveillance in the digitized battlefield and for environmental monitoring. An important issue in the design of these networks is the placement of sensors in the surveillance zone, also described as the sensor field. In a typical scenario, several different types of sensors are available which can be appropriately placed in the sensor field. These sensors differ from each other in their monitoring range, detection capabilities, and cost. Clearly, sensors that can accurately detect targets at longer distances have higher cost. However, the use of these expensive, long-range sensors may be prohibitive in terms of total placement cost. On the other hand, if only small-range sensors are used, effective surveillance can only be achieved with a large number of these sensors. Therefore, efficient sensor placement strategies are necessary to minimize cost and yet achieve mandated levels of surveillance accuracy. Fig. 1 shows a sensor field in which grid points (circles) are at distances of 100m and two sensors are shown with different costs and range of coverage.

Another important problem in sensor networks is that of target location. If the sensor field is represented as a grid (two or three-dimensional), target location refers to the problem of pinpointing a target at a grid point at any point in time. For enhanced coverage, a large number of sensors are typically deployed in the sensor field and, if the coverage areas of multiple sensors overlap, they may all
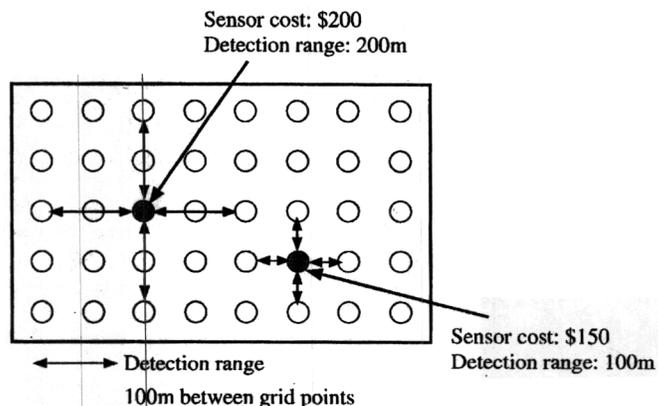

Fig. 1. An example of a two-dimensional sensor field.

report a target in their respective zones. The precise location of the target must then be determined by examining the location of these sensors. In many cases, it is even impossible to precisely locate the target (within the granularity of a single grid point). Alternatively, target location can be simplified considerably if the sensors are placed in such a way that every grid point in the sensor field is covered by a unique subset of sensors. In this way, the set of sensors reporting a target at time t uniquely identifies the grid location for the target at time t. The trajectory of a moving target can also be easily determined in this fashion from time series data.

Previous research in distributed sensor networking has largely ignored the above sensor placement issues. Most prior work has concentrated exclusively on efficient sensor communication [1], [2] and sensor fusion [3], [4] for a given sensor field architecture. However, as sensors are used in greater numbers for field operation, efficient deployment strategies become increasingly important. Related work on terrain model acquisition for motion planning has focused on the movement of a robot in an unexplored "sensor field" [8]. While knowledge of the terrain is vital for surveillance, it does not directly solve the sensor placement problem.

There exists a close resemblance between the sensor placement problem and the guard placement problem (AGP) addressed by the art gallery theorem [9]. The AGP problem can be informally stated as that of determining the minimum number of guards required to cover the interior of an art gallery. (The interior of the art gallery is represented by a polygon.) Several variants of AGP have been studied in the literature, including mobile guards, exterior visibility, and polygons with holes. Our sensor placement problem differs from AGP in two fundamental ways: 1) The sensors can have different ranges, unlike in AGP where guards are assumed to have similar capabilities, and 2) the "identifying" problem for target location requires more sensors than the "covering" problem.

The sensor placement problem for target location is also closely related to the alarm placement problem described in [5]. The latter refers to the problem of placing "alarms" on the nodes of a graph G such that a single fault in the system (corresponding to a single faulty node in G) can be diagnosed. The alarms are therefore analogous to sensors in a sensor field. It was shown in [5] that the alarm placement problem is NP-complete for arbitrary graphs. However, we show that, for restricted topologies, e.g., a set of grid points in a sensor field, a coding theory framework can be used to efficiently determine sensor placement. The sensor locations correspond to codewords of an identifying code constructed over the grid points in the sensor field. Such coding frameworks are often used in computing systems, e.g., for error control [6], and, more recently, for resource placement in multicomputers [7].

- *K. Chakrabarty is with the Department of Electrical and Computer Engineering, Duke University, 130 Hudson Hall, Box 90291, Durham, NC 27708. E-mail: krish@ee.duke.edu.*
- *S.S. Iyengar is with the Department of Computer Science, Louisiana State University, 298 Coates Hall, Baton Rouge, LA 70803.*
- *H. Qi is with the Department of Electrical and Computer Engineering, University of Tennessee, 319 Ferris Hall, 1508 Middle Drive, Knoxville, TN 37996-2100.*
- *E. Cho is with the Division of Mathematics and Sciences, Kentucky State University, 210 Carver Hall, Frankfort, KY 40601.*

We are interested in the following sensor placement problems: 1) Given a surveillance region (grid points) and sensors of different types (with different ranges and costs), determine the placement and type of sensors in the sensor field such that the desired coverage is achieved and cost is minimized; 2) how should the sensors be placed at grid points such that every grid point is covered by a unique subset of these sensors.

We first formulate the sensor placement problem in terms of cost minimization under coverage constraints. We then develop an integer programming (ILP) model to solve the sensor deployment problem. This allows us to leverage efficient ILP solvers for combinatorial optimization problems. We carry out a case study for sensor deployment using a representative ILP solver available in the public domain [11]. We also present a divide-and-conquer approach for solving large problem instances. Finally, we use the theoretical framework of identifying codes [12] to determine the best placement of sensors such that the grid point for a target can be uniquely identified.

2 MINIMUM-COST SENSOR PLACEMENT

Let the sensor field consist of n_x, n_y, and n_z grid points in the x, y, and z dimensions, respectively. We assume that two types of sensors (Type A and Type B) are available for deployment, with costs C_A and C_B and ranges R_A and R_B, respectively. (The model can be easily extended to more than two sensor types.) The separation between the grid points in any dimension is at least $\min\{R_A, R_B\}$. We make the simplifying assumption here that a sensor always detects a target that lies within its range.

The problem that is studied in this section is to minimize the cost of sensors deployed in the sensor field by optimally assigning sensors to grid points. A sensor with range R_A (R_B) placed on grid point (x_1, y_1, z_1) can detect a target (covers) at grid point (x_2, y_2, z_2) if the distance between these two grid points ($\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$) is less than R_A (R_B). However, every grid point must be covered by at least $m \geq 1$ sensors. The parameter m measures the amount of fault tolerance inherent in the sensor deployment scheme. For example, if $m = 1$, then a single sensor failure is likely to make several grid points in the sensor field unobservable. In our optimization framework, we consider m to be a parameter for the ILP model.

The optimization problem is stated as follows:

- $\mathcal{P}1$: Given a parameter $m \geq 1$, a set of grid points, two types of sensors (Type A and Type B) with costs C_A and C_B, and ranges R_A and R_B, respectively, find an assignment of sensors to grid points such that every grid point is covered by at least m sensors and the total cost of the sensors is minimum.

$\mathcal{P}1$ can easily be shown to be NP-hard using the method of restriction. If all sensors have the same cost, the restricted problem $\mathcal{P}1^\star$ is equivalent to the minimum-cost satisfiability problem.

Let a_{ijk} be a 0-1 (binary) variable defined as follows:

$$a_{ijk} = \begin{cases} 1, & \text{if a type A sensor is placed at grid point } (i, j, k) \\ 0, & \text{otherwise.} \end{cases}$$

Likewise, let b_{ijk} be a 0-1 variable defined as follows:

$$b_{ijk} = \begin{cases} 1, & \text{if a type B sensor is placed at grid point } (i, j, k) \\ 0, & \text{otherwise.} \end{cases}$$

The total cost C of sensor deployment is therefore given by the following equation:

$$C = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \sum_{k=1}^{n_z} (C_A a_{ijk} + C_B b_{ijk}).$$

Let $cov^A((i1, j1, k1), (i2, j2, k2))$ be a (derived) binary variable defined as follows:

$$cov^A((i1, j1, k1), (i2, j2, k2)) = \begin{cases} 1, & \text{if a type A sensor placed at grid point } (i1, j1, k1) \\ & \text{covers grid point } (i2, j2, k2) \\ 0, & \text{otherwise.} \end{cases}$$

Similarly, let $cov^B((i1, j1, k1), (i2, j2, k2))$ be the corresponding binary variable for a Type-B sensor.

We now formulate an integer programming model for minimizing the cost of sensor deployment while ensuring that all grid points are covered adequately.

Objective: Minimize the cost function

$$C = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \sum_{k=1}^{n_z} (C_A a_{ijk} + C_B b_{ijk})$$

subject to

$$\sum_{i1=1}^{n_x} \sum_{j1=1}^{n_y} \sum_{k1=1}^{n_z} (a_{i1,j1,k1} cov^A((i1, j1, k1), (i2, j2, k2)) + b_{i1,j1,k1} cov^B((i1, j1, k1), (i2, j2, k2)) \geq m$$
$$1 \leq i2 \leq n_x, 1 \leq j2 \leq n_y, 1 \leq k2 \leq n_z.$$

In order to solve the above problem by standard ILP techniques, we first need to express the derived binary variables in terms of the independent binary variables. This is done as shown below for $cov^A((i1, j1, k1), (i2, j2, k2))$. A similar method can be used for $cov^B((i1, j1, k1), (i2, j2, k2))$.

Let

$$d((i1, j1, k1), (i2, j2, k2)) = \sqrt{(i1 - i2)^2 + (j1 - j2)^2 + (k1 - k2)^2}$$

denote the distance between grid points $(i1, j1, k1)$ and $(i2, j2, k2)$. Since $cov^A((i1, j1, k1), (i2, j2, k2)) = 1$ if and only if a Type-A sensor placed at $(i1, j1, k1)$ covers grid point $(i2, j2, k2)$, we introduce the following two inequalities. (For notational convenience, we use cov^A and d without any loss in generality.)

$$cov^A \cdot (R_A - d) \geq 0 \tag{1}$$

$$(1 - cov^A) \cdot (d - R_A) \geq 0. \tag{2}$$

We can verify that if $d < R_A$, then cov^A must be 1 in order to satisfy (2). Similarly, if $d > R_A$, then cov^A must be 0 in order to satisfy (1). Note that the case $d = R_A$ is not considered here—instead, we assume that this case is avoided since the range is usually an integer while the distance (computed by the square root operator) will be a noninteger. Even if the distance is an integer, a fractional offset can be added to the range in order to use the ILP model.

The constraint in the optimization model is nonlinear since it involves products of binary variables. In order to linearize the constraint inequalities, we introduce a new binary variable for each appearance of a nonlinear term. Suppose u and v are binary variables. Then, their product uv can be replaced by a new binary variable w with the following additional constraints [10]: 1) $u + v \geq 2w$, 2) $u + v - 1 \leq w$. The resulting integer linear programming model is shown in Fig. 2.

We carried out a case study for two-dimensional sensor fields with a given number (p) of grid points in each dimension using the two types of sensors discussed in Section 1. These include a type-A sensor with cost \$150 and range 100m and a type-B sensor with

$$\text{Minimize } C = \sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\sum_{k=1}^{n_z}(C_A a_{ijk} + C_B b_{ijk}) \text{ subject to:}$$

1. $cov^A \cdot (R_A - d) \geq 0$ for all pairs of grid points

2. $(1 - cov^A) \cdot (d - R_A) \geq 0$ for all pairs of grid points

3. $cov^B \cdot (R_B - d) \geq 0$ for all pairs of grid points

4. $(1 - cov^B) \cdot (d - R_B) \geq 0$ for all pairs of grid points

5. $\sum_{i1=1}^{n_x}\sum_{j1=1}^{n_y}\sum_{k1=1}^{n_z}(g((i1,j1,k1),(i2,j2,k2)) + h((i1,j1,k1),(i2,j2,k2))) \geq m,$
   $1 \leq i2 \leq n_x, 1 \leq j2 \leq n_y, 1 \leq k2 \leq n_z$

6. $cov^A((i1,j1,k1),(i2,j2,k2)) + d((i1,j1,k1),(i2,j2,k2)) \geq 2g((i1,j1,k1),(i2,j2,k2)),$
   $1 \leq i2 \leq n_x, 1 \leq j2 \leq n_y, 1 \leq k2 \leq n_z$

7. $cov^A((i1,j1,k1),(i2,j2,k2)) + d((i1,j1,k1),(i2,j2,k2)) - 1 \leq g((i1,j1,k1),(i2,j2,k2)),$
   $1 \leq i2 \leq n_x, 1 \leq j2 \leq n_y, 1 \leq k2 \leq n_z$

8. $cov^B((i1,j1,k1),(i2,j2,k2)) + d((i1,j1,k1),(i2,j2,k2)) \geq 2g((i1,j1,k1),(i2,j2,k2)),$
   $1 \leq i2 \leq n_x, 1 \leq j2 \leq n_y, 1 \leq k2 \leq n_z$

9. $cov^B((i1,j1,k1),(i2,j2,k2)) + d((i1,j1,k1),(i2,j2,k2)) - 1 \leq h((i1,j1,k1),(i2,j2,k2)),$
   $1 \leq i2 \leq n_x, 1 \leq j2 \leq n_y, 1 \leq k2 \leq n_z$

10. $a_{ijk}, b_{ijk} = 0$ or $1, 1 \leq i \leq n_x, 1 \leq j \leq n_y, 1 \leq k \leq n_z$

11. $a_{ijk} + b_{ijk} \leq 1, 1 \leq i \leq n_x, 1 \leq j \leq n_y, 1 \leq k \leq n_z$

Fig. 2. Integer linear programming model for sensor placement.

cost \$200 and range 200m. We used the *lpsolve* package from Eindhoven University of Technology in The Netherlands [11]. The *lpsolve* input files were automatically generated using a Perl script. The results on sensor placement for two values of p and m are shown in Fig. 3 and Fig. 4 and in Table 1. Note that an exhaustive search would require us to examine a total of 3^{p^2} possible sensor deployments, hence the ILP model provides a convenient method for performing this search in a systematic manner.

We draw several important conclusions from the case study. First, as the value of m increases, it is more economical to use the Type-B sensor, even though it costs more than the Type-A sensor. This can be attributed to the fact that, even though a Type-B sensor costs 1.5 times more, it has a range that is twice that of a Type-A sensor. We expect the converse to be true if the cost increases at a faster rate than the increase in the sensor range. The second observation we make is that an exact solution to the ILP model takes an excessive amount of computation time for larger problem instances. Therefore, we propose a "divide-and-conquer" near-optimal approach for

sensor placement when the number of grid points is very large (> 50). It is based on the following observation:

Given a set of available sensors, let $N_{p,n}$ be an optimal number of sensors required for covering a sensor field in n dimensions, with p grid points in each dimension. Let the corresponding cost of sensor deployment be $C_{p,n}$. Then, the number of sensors are given by: 1) $N_{2p,n} \leq 2^n N_{p,n}$, 2) $C_{2p,n} \leq 2^n C_{p,n}$.

For example, when $p = 8$ and we use the two sensor types discussed above, we can obtain complete coverage for $m = 2$ using $2^2 \cdot 5 = 20$ sensors (sensor density = 0.31). We can also determine the number of sensors and their cost as follows: 1) for $p = 4 \cdot 2^k$, $N_{p,n}^1 \leq N_{4,n}^1 \cdot 2^{nk}$, $N_{p,n}^2 \leq N_{4,n}^2 \cdot 2^{nk}$. 2) $N_{p,n}^3 \leq N_{4,n}^3 \cdot 2^{nk}$. Moreover, $C_{p,n}^1 \leq C_{4,n}^1 \cdot 2^{nk}$, $C_{p,n}^2 \leq C_{4,n}^2 \cdot 2^{nk}$, and $C_{p,n}^3 \leq C_{4,n}^3 \cdot 2^{nk}$. 3) For $p = 5 \cdot 2^k$, $N_{p,n}^1 \leq N_{5,n}^1 \cdot 2^{nk}$, $N_{p,n}^2 \leq N_{5,n}^2 \cdot 2^{nk}$. Finally,

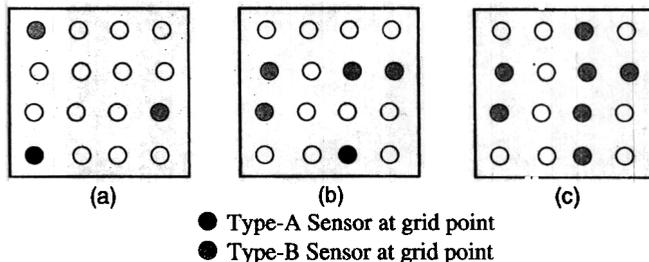$$C_{p,n}^1 \leq C_{5,n}^1 \cdot 2^{nk}, \quad C_{p,n}^2 \leq C_{5,n}^2 \cdot 2^{nk}.$$

(a)   (b)   (c)

● Type-A Sensor at grid point
◕ Type-B Sensor at grid point

Fig. 3. Optimal (minimum-cost) sensor placements for various sensor fields for $p = 4$: (a) $m = 1$, (b) $m = 2$, (c) $m = 3$.

(a)   (b)

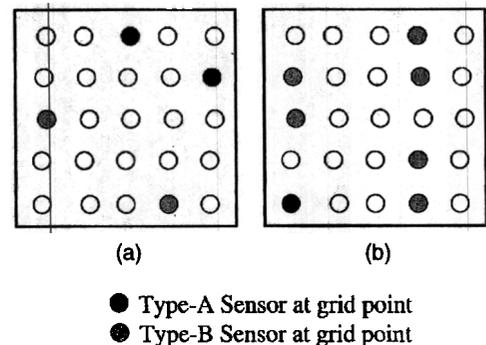● Type-A Sensor at grid point
◕ Type-B Sensor at grid point

Fig. 4. Optimal (minimum-cost) sensor placements for $p = 5$: (a) $m = 1$, (b) $m = 2$.

TABLE 1
Results on Optimal Sensor Placement Using Integer Linear Programming

p	m	No. of grid points p	Number of sensors			Cost ($)	Sensor density
			Type-A	Type-B	Total		
4	1	16	1	2	3	550	0.19
4	2	16	1	4	5	950	0.31
4	3	16	0	7	7	1400	0.44
5	1	25	2	2	4	700	0.16
5	2	25	1	6	7	1350	0.28

Note that the sensor density remains constant as larger sensor fields are considered using a divide-and-conquer approach. For example, the sensor density is the same for $p = 4$ and $p = 8$. Note that alternative divide-and-conquer approaches are also possible. For example, instead of dividing p by 2, we can also divide by l ($l > 2$), which yields 1) $N_{lp,n} \leq l^n N_{p,n}$ and 2) $C_{lp,n} \leq l^n C_{p,n}$.

3 SENSOR PLACEMENT FOR TARGET LOCATION

In this section, we address the problem of placing sensors on grid points such that the grid positions of targets can be uniquely identified from the subset of sensors that detect the targets. This approach is based on the concept of identifying codes for uniquely identifying vertices in graphs [12].

The identifying code problem can be stated as an optimal covering of vertices in an undirected graph G such that any vertex in G can be uniquely identified by examining the vertices that cover it. A ball of radius r centered on a vertex v is defined as the set of vertices that are at distance at most r from v, where the distance between any two vertices u and v is defined to be the number of edges on a shortest path from u to v. The vertex v is then said to cover itself and every other vertex in the ball with center v. The formal problem statement is as follows: Given an undirected graph G and an integer $r \geq 1$, find a (minimal) set C of vertices such that every vertex in G belongs to a unique set of balls of radius r centered at the vertices in C. The set of vertices thus obtained constitutes a code for vertex identification.

We now show that the problem of placing sensors for unique target identification can be solved using the theory of identifying codes. The grid points in the sensor field correspond to the vertices in the graph G, while the centers of the balls correspond to the grid points where sensors are placed. The unique identification of a vertex in G corresponds to the unique location of a target by the sensors in the sensor field. Each sensor at a grid point can detect a target at grid points that are adjacent to it.

Let S_n^p denote the number of sensors required for uniquely identifying targets in an n-dimensional ($n \leq 3$) sensor field with p grid points in each dimension. We refer to such a grid as an (n,p) grid. The following theorem provides upper and lower bounds on S_n^p for $r = 1$. Its proof follows from the properties of identifying codes on regular graphs [12].

Theorem 1. The number of sensors S_n^p for uniquely identifying a target in an (n,p) grid is given by $p^n/(n+1) \leq S_n^p \leq p^n/n$.

For example, for a two-dimensional sensor field with 100 grid points in each dimension, at least 3,334 sensors are required for the 10^4 grid points. However, 5,000 sensors are adequate for unique target identification. For a two-dimensional sensor field, the upper bound corresponds to a checkerboard placement of sensors on grid points, as shown in Fig. 5. The grid points are marked by their (x,y) coordinates and each sensor can detect a target at distances up to the next grid point in each dimension. Note that each grid point for this placement is covered by a unique subset of sensors.

We now describe more efficient sensor placement strategies based on coding theory principles from [12]. We first review some terminology. For every grid point (x,y,z) in a sensor field, we associate a parity vector (p_x,p_y,p_y) as follows: $p_x = x \bmod 2$, $p_y = y \bmod 2$, $p_z = z \bmod 2$. For example, the parity vector for grid point $(2,4,5)$ in a three-dimensional sensor field is $(0,0,1)$. The set of parity vectors corresponding to the set of grid points C is called the binary parity code and denoted by $\mathcal{P}(C)$.

Theorem 2. For a $(3,p)$ grid with p even and $p > 2$, target location is achieved with a smallest possible number of sensors ($S_3^p = p^3/4$) if the binary parity code $\mathcal{P}(C)$ is the perfect binary $(3,1,3)$ Hamming code, where a perfect (n,k,d) Hamming code consists of 2^k codewords in n dimensions and the minimum distance between codewords is d.

Proof. We first prove that every grid point is uniquely covered. Every sensor is covered only by itself because the Hamming distance between any two parity vectors is at least three. Next, consider a noncodeword vertex with coordinates (x_1,x_2,x_3) and corresponding parity vector (p_1,p_2,p_3). There are two vertices with coordinates $x' = (x_1',x_2',x_3')$ and $x'' = (x_1'',x_2'',x_3'')$ such that they have the same parity vector (q_1,q_2,q_3), x' and x'' are neighbors of x in the n-dimensional sensor field, (q_1,q_2,q_3) belongs to the Hamming code, and the Hamming distance between (p_1,p_2,p_3) and (q_1,q_2,q_3) is one. We note that x' and x'' are uniquely determined by x.

To prove necessity, we note that if two sensors in the (n,p) grid are neighbors, their parity vectors are at distance 1. Thus, for an identifying code, the covering radius of the set of parity vectors must be equal to 1 and the smallest set with this property is a perfect $(3,1,3)$ code. ☐

The following theorem shows that if the number of grid points in each dimension is even, the lower bound on the number of sensors (Theorem 1) can be achieved for a three-dimensional sensor field. The proof follows from Theorem 2.

Theorem 3. For a $(3,p)$ grid $(p > 4$, p even), sensor placement with a minimum number of sensors ($S_n^p = p^3/4$) is be achieved if and only if
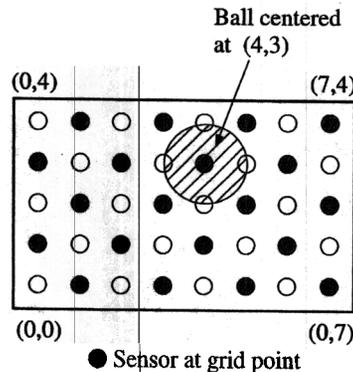

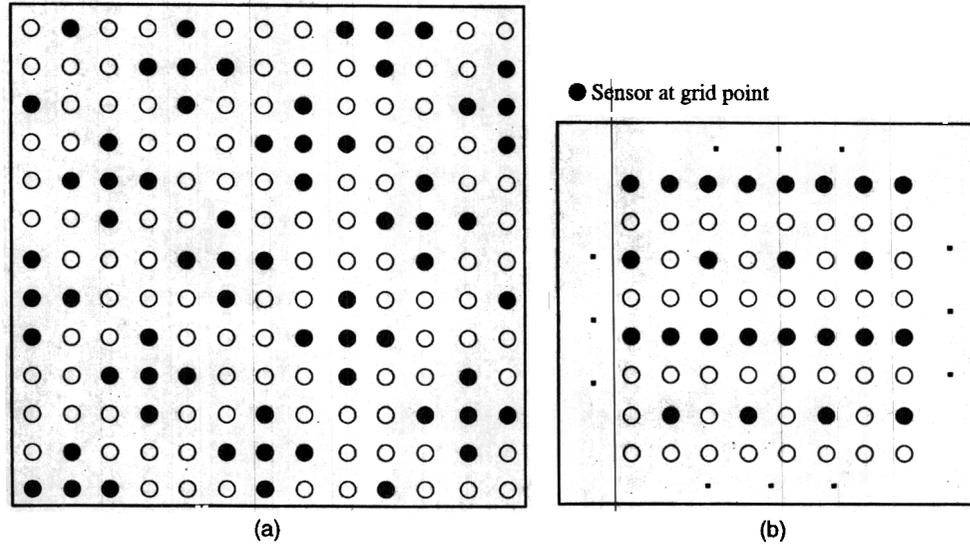Fig. 5. A checkerboard placement of sensors.

Fig. 6. (a) An efficient placement of sensors given by Theorem 3. (b) An efficient ad hoc placement of sensors.

sensors are placed on grid points whose parity vectors are (0, 0, 0) and (1, 1, 1).

Theorem 3 shows that if p is even, the sensor density (average number of sensors per grid point) for three-dimensional sensor fields is only 0.25. For example, let $p = 6$. From Theorem 2, we see that sensors should be placed at the set of grid points $\{S_0, S_1\}$, where S_0 and S_1 are the set of grid points with parity vectors (0, 0, 0) and (1, 1, 1), respectively, as shown below:

$$S_0 = \{(0,0,0),(0,0,2),(0,2,0),(0,2,2),(0,0,4),(0,4,0),(0,4,2),$$
$$(0,2,4),(0,4,4),(2,0,0),(2,0,2),(2,2,0),(2,2,2),(2,0,4),(2,4,0),$$
$$(2,2,4),(2,4,4),(4,0,0),(4,0,2),(4,2,0),(4,2,2),(4,0,4),(4,4,0),$$
$$(4,4,2),(4,2,4),(4,4,4)\};$$

$$S_1 = \{(1,1,1),(1,3,1),(1,3,1),(1,3,3),(1,1,5),(1,5,1),(1,5,3),$$
$$(1,3,5),(1,5,5),(3,1,1),(3,1,3),(3,3,1),(3,3,3),(3,1,5),(3,5,1),$$
$$(3,3,5),(3,5,5),(5,1,1),(5,1,3),(5,3,1),(5,3,3),(5,1,5),(5,5,1),$$
$$(5,5,3),(5,3,5),(5,5,5)\}.$$

Hence, a total of 54 sensors are required for the 216 grid points.

The next theorem addresses cases where p is not necessarily even. For a sensor field with p grid points in each dimension, we can define an n-dimensional p-ary code C with covering radius 2 as follows: C is the smallest set of grid points (vertices) such that each noncodeword is at a distance at most two from a codeword. Note that the distance between two points (x_1, y_1, z_1) and (x_2, y_2, z_2) in this context is given by $d = |x_1 - x_2| + |y_1 - y_2| + |z_1 - z_2|$.

Theorem 4. Let $K^p(n, 2)$ be the minimum number of codewords in a p-ary n-dimensional code with covering radius 2. Then, for any $p > 4$, an upper bound on the minimum number of sensors S_n^p for target location in an (n, p) grid is given by $S_n^p \leq (2n + 1)K^p(n, 2)$.

Proof. It is sufficient to show that all grid points in a ball B_2 of radius 2 with center v can be uniquely identified by balls of radius 1 centered at all gridpoints that belong to the ball B_1 of radius 1 centered at v. Without loss of generality, assume that $v = (0, 0, \ldots, 0)$. Then,

$$B_1 = \{(0, 0, \quad , 0)\} \bigcup \{(0, \quad , 0, \pm 1, 0, \quad , 0) \bmod (p)\}$$

and

$$B_2 = B_1 \bigcup \{(0, \ldots, 0, \pm 2, 0, \ldots, 0) \bmod (p)\} \bigcup$$
$$\{(0, \ldots, \pm 1, 0, \ldots, 0, \pm 1, 0, \ldots, 0) \bmod (p)\}.$$

Let $x \in B_2$. Consider the following four cases:

1. $x = (0, \ldots, 0)$. Then, x belongs to all balls of radius 1 with centers in B_1.
2. $x = (0, \ldots, 0, \pm 1, 0, \ldots, 0)$. Then, x belongs to two balls of radius 1 with centers at x and $(0, \ldots, 0)$, respectively.
3. $x = (0, \ldots, 0, \underset{i}{\pm 1}, 0, \ldots, \underset{j}{\pm 1}, 0, \ldots, 0)$. Then, x belongs to two balls with centers $(0, \ldots, 0, \underset{i}{\pm 1}, 0, \ldots, 0)$ and $(0, \ldots, 0, \underset{j}{\pm 1}, 0, \ldots, 0)$
4. $x = (0, \ldots, 0, \underset{i}{\pm 2}, 0, \ldots, 0)$. Then, x belongs to one ball with center $(0, \ldots, 0, \underset{i}{\pm 1}, 0, \ldots, 0)$

This completes the proof. □

Theorem 4 implies that sensor placement can be carried out by first determining a code $K^p(n, 2)$ with covering radius 2. (Tables of covering codes are easily available [13].). Sensors are then placed on the grid points corresponding to the codewords, as well as on all grid points that are adjacent to codewords of $K^p(n, 2)$. This is shown in Fig. 6a for a two-dimensional sensor field with $p = 13$. We need a total of 65 sensors for 169 grid points (sensor density = 0.38), which is slightly greater than the lower bound of 57 predicted by Theorem 1. Note, however, that the lower bound need not always be achievable. As another example, let $p = 5$ and $n = 3$. For this case, $K^5(3, 2) = 5$ [13], hence a total of 35 sensors placed at the 125 grid points provides unique target location.

While the above sensor placement strategy can be used in general for any $p > 4$, the sensor density can often be decreased for specific values of p. For example, consider the special case $p = 8s$ and $n = 2$. An ad hoc sensor placement given by Fig. 6b yields a sensor density of only 0.375, which improves upon the construction of Theorem 4.

We have assumed thus far that the location of only a single target in the sensor field has to be uniquely identified. We now show that sensor placement for unique location of single target provides a near-complete location of sets of targets. This demonstrates that the sensor placement strategy outlined in this section is effective even for tracking multiple targets in the sensor field. Let

$C(l)$ be the fraction of sets of targets of cardinality exactly l that are uniquely identifiable. The following lemma provides a lower bound on the fraction of multiple targets that can be located:

Lemma 1. *The fraction $C(l)$ of sets of targets of cardinality exactly l that are uniquely identifiable with $r = 1$ by sensor placement for single targets is lower bounded by $C(l) \geq \Pi_{i=0}^{l-1} \frac{N - iV(4)}{N - i}$, where $V(4)$ is an upper bound on the number of grid points at a distance up to four from any given grid point and N is the number of grid points in the sensor field.*

Proof. A set of targets is uniquely identifiable if the distance between any two targets (grid points) in this set is at least five. Note that this condition is sufficient but not necessary. The fraction of identifiable sets of vertices is therefore lower-bounded by

$$C(l) \geq \frac{N(N - V(4))(N - 2V(4)) \cdots (N - (l-1)V(4))}{\binom{N}{l} l!}$$

$$= \Pi_{i=0}^{l-1} \frac{N - iV(4)}{N - i}$$

$\square$

The above lemma can be used to show that if the number of grid points is sufficiently large relative to the cardinality of the set of targets, the multiple targets can be uniquely located (asymptotically) using sensor placement for single targets.

Theorem 5. *As the number of grid points in a sensor field tends to infinity, the fraction of sets of targets of cardinality exactly l that are uniquely identifiable approaches one if $l = o(\sqrt{N})$.*

Proof. Let $\Pi = \Pi_{i=0}^{l-1}(\frac{N - iV(4)}{N - i})$. It can be easily seen that, for $i \lesssim \sqrt{N}$,

$$\ln \frac{N - iV(4)}{N - i} = \ln\left(1 - \frac{i(V(4) - 1)}{N - i}\right) \sim -\frac{i(V(4) - 1)}{N - i},$$

and $\ln \Pi \sim \sum_{i=1}^{l-1} -\frac{i(V(4)-1)}{N-i}$. Now,

$$\left| \sum_{i=1}^{l-1} \frac{i(V(4) - 1)}{N - i} \right| \leq \frac{(l-1)(V(4) - 1)}{N - l + 1}(l - 1)$$

and $\lim_{N \to \infty} \frac{(l-1)(V(4)-1)}{N-l+1}(l - 1) = 0$ if $l^2/N \to 0$ (since $V(4)$ is constant). $\square$

This underlines the effectiveness of the sensor placement approach for single targets and implies that separate placement algorithms for multiple targets are not necessary.

ACKNOWLEDGMENTS

REFERENCES

[1] J.M. Kahn, R.H. Katz, and K.S.J. Pister, "Mobile Networking for Smart Dust," *Proc. ACM/IEEE Int'l Conf. Mobile Computing and Networks*, 1999.

[2] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proc. ACM/IEEE Int'l Conf. Mobile Computing and Networks*, 1999.

[3] R.R. Brooks and S.S. Iyengar, *Multi-Sensor Fusion: Fundamentals and Applications with Software*. Upper Saddle River, N.J.: Prentice Hall, 1998.

[4] S.S. Iyengar, L. Prasad, and H. Min, *Advances in Distributed Sensor Technology*. Englewood Cliffs, N.J.: Prentice Hall, 1995.

[5] N.S.V. Rao, "Computational Complexity Issues in Operative Diagnosis of Graph-Based Systems," *IEEE Trans. Computers*, vol. 42, no. 4, pp. 447-457, Apr. 1993.

[6] T.R.N. Rao, *Error Control Coding for Computer Systems*. Englewood Cliffs, N.J.: Prentice Hall, 1989.

[7] M. Bae and B. Bose, "Resource Placement in Torus-Based Networks," *IEEE Trans. Computers*, vol. 46, no. 10, pp. 1083-1092, Oct. 1997.

[8] N.S.V. Rao, S.S. Iyengar, B.J. Oomen, and R.L. Kashyap, "On Terrain Model Acquisition by a Point Robot Amidst Polyhedral Obstacles," *IEEE J. Robotics and Automation*, vol. 4, pp. 450-455, Aug. 1988.

[9] J. O'Rourke, *Art Gallery Theorems and Algorithms*. New York: Oxford Univ. Press, 1987.

[10] H.P. Williams, *Model Building in Mathematical Programming*, second ed. New York: John Wiley, 1985.

[11] M. Berkelaar, "lpsolve, version 3.0," Design Automation Section, Eindhoven Univ. of Technology, Eindhoven, The Netherlands, ftp://ftp.ics.ele.tue.nl/pub/lp_solve, 1999.

[12] M.G. Karpovksy, K. Chakrabarty, and L.B. Levitin, "A New Class of Codes for Covering Vertices in Graphs," *IEEE Trans. Information Theory*, vol. 44, pp. 599-611, Mar. 1998.

[13] G.D. Cohen, I. Honkala, S.N. Litsyn, and A. Lobstein, *Covering Codes*. Amsterdam: North Holland, 1997.