Minimal and Maximal Exposure Path Algorithms for Wireless Embedded Sensor Networks

Giacomino Veltri
University of California, Los Angeles
4821 Boelter Hall
Los Angeles, California 90095-1596
1-310-474-8811

gveltri@cs.ucla.edu

Qingfeng Huang
Washington University
Mobile Computing Laboratory
St. Louis, Missouri 63130
1-314-935-6190

qingfeng@cse.wustl.edu

Gang Qu
University of Maryland
1417 A. V. Williams Building
College Park, Maryland 20742
1-301-405-6703

gangqu@eng.umd.edu

Miodrag Potkonjak
University of California, Los Angeles
3531 Boelter Hall
Los Angeles, California 90095-1596
1-310-825-1322

miodrag@cs.ucla.edu

ABSTRACT

Sensor networks not only have the potential to change the way we use, interact with, and view computers, but also the way we use, interact with, and view the world around us. In order to maximize the effectiveness of sensor networks, one has to identify, examine, understand, and provide solutions for the fundamental problems related to wireless embedded sensor networks. We believe that one of such problems is to determine how well the sensor network monitors the instrumented area. These problems are usually classified as coverage problems. There already exist several methods that have been proposed to evaluate a sensor network's coverage.

We start from one of such method and provide a new approach to complement it. The method of using the minimal exposure path to quantify coverage has been optimally solved using a numerical approximation approach. The minimal exposure path can be thought of as the worst-case coverage of a sensor network. Our first goal is to develop an efficient localized algorithm that enables a sensor network to determine its minimal exposure path. The theoretical highlight of this paper is the closed-form solution for minimal exposure in the presence of a single sensor. This solution is the basis for the new and significantly faster localized approximation algorithm that reduces the theoretical complexity of the previous algorithm. On the other hand, we introduce a new coverage problem – the maximal exposure path – which is in a sense the best-case coverage path for a sensor network. We prove that the maximal exposure path problem is NP-hard, and thus, we provide heuristics to generate approximate solutions.

In addition, we demonstrate the effectiveness of our algorithms

through several simulations. In the case of the minimal single-source minimal exposure path, we use variational calculus to determine exact solutions. For the case of maximal exposure, we use networks with varying numbers of sensors and exposure models.

Categories and Subject Descriptors

C.4 [**Computer Systems Organization**]: Performance of Systems – *measurement techniques, performance attributes.*

General Terms

Algorithms, Performance, Design, Theory.

Keywords

Wireless Embedded Sensor Networks, Centralized Optimal Algorithms, Localized Algorithms, Simulation.

1. INTRODUCTION

Wireless sensor networks have the potential to significantly impact not only the way people interact with, use, and think about computers, but also the way people interact, use, and think about the surrounding environment. A wireless sensor network is a collection of sensors that have a limited amount of computational and battery capacity, the ability to communicate with each other, and the ability to sense the environment around them [6]. Sensors can be deployed into dangerous or difficult to reach areas, forming a wireless network, to monitor the environment. Examples include sensors near the edge of a volcano to monitor the seismic activity, sensors placed into the walls of buildings to monitor the structural integrity, and sensors placed in a battlefield providing the capability of monitoring and tracking targets [6]. The cost effectiveness and ease of deployment of wireless sensor networks make them the perfect candidates for such applications. Most existing work focuses on reducing the design and maintenance (including deployment) cost or increasing sensor network's reliability and lifetime. However, we believe that

another crucial problem is to determine how well the sensor network monitors (or "covers") the designated area. This is one of the most important criteria for evaluating the sensor network's effectiveness and we study, in this paper, to understand, formulate, and provide solutions to this fundamental problem.

Specifically, we consider the following scenario: given a sensor network, how does one traverse through the sensor field from one point to another such that the sensors have the least or most coverage of the traveled path. The first part, referred to as the least-covered path problem, corresponds to the sensor network's worst-case coverage. Sensor networks are designed to monitor the sensor field, and the least-covered path measures the ability to move in the sensor field without being discovered. Therefore, how can one design the sensor network to maximize such minimum coverage becomes vital and has attracted significant attention recently [5], [9], [10]. Two popular methods addressing this problem are the maximum breach path and the minimal exposure path. The maximum breach path problem seeks to find a path such that the path's maximum exposure to the sensors at any given point is minimized [10]. The minimal exposure path problem seeks to find a path between two given points such that the total exposure acquired from the sensors by traversing the path is minimized [10], [11].

Using the minimal exposure path problem to evaluate a network is important. Once the minimal exposure path is known, the user can manipulate sensors in the network or add sensors to the network to increase coverage. However, using a centralized method to determine the minimal exposure path means that the moved/added node's position must be reported to a central node or to every other node in the network. If a central node is used, a single-point-of-failure problem arises; if the new data is propagated to every other node in the network, the problem of energy consumption becomes more important. Therefore, we present a localized minimal exposure path algorithm such that only neighboring nodes need to be updated and path information can be calculated on-line in an easier and more efficient manner.

While the least coverage problem takes the pessimistic view of the sensor network design, similar to real-time system designs based on worst-case execution time analysis, it is equally important to study the second part, which we call the maximal exposure path problem. A maximal exposure path is the path by following which the total exposure to the sensors is maximized – that is, the path that is best covered by the sensors. Normally, there is a maximum length constraint on the maximal exposure path or a delay constraint that specifies how long the object can stay in the sensor field. Otherwise, one can keep moving without reaching the ending point or stay at a point with positive exposure forever to accumulate infinite coverage. Unlike the least coverage problem, which measures how well the sensor network monitors its environment, the maximal exposure problem, although it can be interpreted as the best case of coverage, finds applications in the environment that can benefit from the deployed sensors. These applications include, for example, how to manage (such as collecting information from the sensors) the sensor network efficiently and how to get the most benefit, according to some pre-defined benefit function, from the sensors while traversing the sensor field. A more concrete example of using the maximal exposure path to determine the benefit of a sensor network is with respect to a light-detecting network. Assume that a solar-powered autonomous robot is traversing the network and would like to travel the past that accumulates the most light within a certain time frame. By using the maximal exposure path of the light-detecting network, the solar-powered robot can gain as much light as possible within its limited time. Therefore, the maximal exposure path is just as important to a user trying to receive as much benefit (or detection) from the network as the minimal exposure path is to a user trying to receive as little benefit (or detection) from the network.

In this paper, we focus on both the minimal and maximal exposure path problems, which correspond to the worst- and best-case coverage. Our goal is to shed light on the sensor network coverage problem by formulating and providing practical solutions to both extremes. The rest of the paper is organized as follows: in Section 2, we first introduce the exposure model and then discuss the minimal exposure path problem highlighted by an analytic solution to the single-sensor problem and an approximation algorithm to the multiple sensor problem. Section 3 focuses on the maximal exposure path problem, which we show is NP-complete and provide several heuristic methods proposed to tackle this problem. Simulation results are presented to validate our claims in the respective sections. In Section 4, we use the linear programming formulation to show the intrinsic relationship between these two problems. The most directly related work is reviewed in Section 5 before we conclude.

2. Minimal Exposure Path Problem
In this section, we describe our sensor exposure model, formulate the minimal exposure path problem and present our solutions. Specifically, we obtain the optimal solution analytically for the single-sensor case by the method of variational calculus. We also develop an approximation algorithm for the case of multiple sensors based on the Voronoi diagram concept.

2.1 Minimal Exposure Path Problem
Although our solutions will work with an arbitrary exposure model, for simplicity of the discussion, we take the model that measures a single sensor's sensitivity to an object or a point in the sensor field by a function that is inversely correlated with the distance between the sensor and the point being monitored. This model reflects the most important nature of sensor's sensitivity: the closer the object is to the sensor, the better that the sensor can monitor it. The simplest case is when the sensor's sensitivity function $f(s,p)$ takes the form of $1/d^k(s,p)$, where $d(s,p)$ is the distance from point p to the sensor s and k is a positive constant. The constant parameter k, indicating how quickly the sensor loses its sensitivity as the object movers away, adds some flexibility to this model.

In a sensor network with multiple sensors, the exposure of an object, or any point in the sensor field can be defined in the following two ways. The *closest sensor* exposure model uses the largest exposure value of any given point to each individual sensor as the exposure at that point. The *all sensor* exposure model takes the sum of all the individual exposure values as the exposure to the sensor network. When we assume that the sensor's sensitivity is inversely proportional to the distance, the Voronoi diagram can be used to conveniently determine the exposure in the *closest sensor* exposure model. The Voronoi diagram partitions the sensor field into *n* polygonal regions, which are called Voronoi cells, where *n* is the number of sensors. Each Voronoi cell contains exactly one sensor and any point inside the

Voronoi cell is closer to the sensor in the same cell than any of the other sensors.

Given two points $A = (x_a, y_a)$ and $B = (x_b, y_b)$ in the two-dimensional sensor field, a path connecting them is a continuous function $P(t) = (x(t), y(t))$ with boundary conditions $x(t_1) = x_a$, $x(t_2) = x_b$, $y(t_1) = y_a$, $y(t_2) = y_b$. The exposure of this path can be defined as the accumulated exposure to the sensor network as the object moves along the path, which is equivalent to

$$\int_{t_1}^{t_2} f\left(S, (x(t), y(t))\right) \sqrt{\left(\frac{dx(t)}{dt}\right)^2 + \left(\frac{dy(t)}{dt}\right)^2} \, dt \qquad (2.1)$$

where $f(S, (x(t), y(t)))$ is the exposure to the sensor network S at point $(x(t), y(t))$.

We define the minimal exposure path between two points in a sensor network to be a path within the sensor field connecting the two points with the minimum exposure as defined above in Equation (2.1)

2.2 Single-Sensor Optimal Solution

The definition of exposure makes it obviously a path-dependent value. Given two end-points in the field, different paths between them are likely to have different exposure values. Before discussing our solution, we assume that the speed traveled by an object through the sensor network is bounded, and that when dealing with minimal exposure paths, the object is able to and will always travel at the maximum speed (otherwise the exposure would not be minimal). Under these assumptions, the minimal exposure path problem for a sensor field with a single sensor can be optimally solved by the method of variational calculus developed by Euler (1707-1783) and Lagrange (1736-1813).

Theorem 1. ([8]) *Let $J[y]$ be a function(al) of the form $J[y] = \int_a^b F(x, y, y') dx$ defined on the set of functions $y(x)$ which have continuous first derivatives in [a; b] and satisfy the boundary condition $y(a) = A$, $y(b) = B$. Then a necessary condition for $J[y]$ to have an extremum for a given function $y(x)$ is that $y(x)$ satisfy the following Euler's equation, which is also referred as the Euler-Lagrange equation:*

$$\frac{\partial F}{\partial y} - \frac{d}{dx}\frac{\partial F}{\partial y'} = 0 \qquad (2.2)$$

Interested readers may find a proof of this theorem in most variational calculus or functional analysis books. Now we derive an analytic solution to the minimal exposure path problem for a single-sensor field.

Theorem 2. *Given a sensor sensitivity function $f(s, p) = 1/d(s, p)$, where $d(s, p)$ is the distance between the sensor location s and an arbitrary location p, the minimal exposure path from an arbitrary point A to another arbitrary point B is of the following form in polar coordinates:*

$$\rho(\theta) = a e^{\frac{\ln(b/a)}{\alpha}\theta} \qquad (2.3)$$

where constants a and b are the distances from the sensor to the points A and B respectively, and α is the angle formed by (A-sensor-B).

Proof:

We first rewrite the exposure along a path defined in Equation (2.1) into polar coordinates (ρ, θ) by the transformation: $x(t) = \rho(t)\cos\theta(t)$ and $y(t) = \rho(t)\sin\theta(t)$ and we take the sensor's location as the origin

$$\int_{t_1}^{t_2} f\left(S, (\rho(t), \theta(t))\right) \sqrt{\left(\rho\frac{d\theta(t)}{dt}\right)^2 + \left(\frac{d\rho(t)}{dt}\right)^2} \, dt$$

$$= \int_{\theta(t_1)}^{\theta(t_2)} f\left(S, (\rho(t), \theta(t))\right) \sqrt{\rho(t)^2 + \left(\frac{d\rho(t)}{d\theta(t)}\right)^2} \, d\theta(t) \qquad (2.4)$$

Let α be the value of the angle formed by the straight line segments connecting the sensor and points A and B respectively, and a and b be length of these line segments, the above integral can be further simplified to:

$$\int_0^\alpha f\left(S, (\rho, \theta)\right) \sqrt{\rho^2 + \left(\frac{d\rho}{d\theta}\right)^2} \, d\theta \qquad (2.5)$$

with new boundary conditions: $\rho(0) = a$ and $\rho(\alpha) = b$.

When the sensor sensitivity function is $f(s, p) = 1/d(s, p) = 1/\rho$, the minimal exposure path problem becomes to find the path $\rho(\theta)$ that minimizes

$$\int_0^\alpha \frac{1}{\rho} \sqrt{\rho^2 + \left(\frac{d\rho}{d\theta}\right)^2} \, d\theta \qquad (2.6)$$

Define $F = \frac{1}{\rho}\sqrt{\rho^2 + \left(\frac{d\rho}{d\theta}\right)^2}$ and plug into the Euler-Lagrange Equation 2.2 and we obtain $\sqrt{1 + \frac{1}{\rho^2}\left(\frac{d\rho}{d\theta}\right)^2} = c$, which has the general solution in the form of $\rho = c_2 \exp^{\pm c_1\theta}$ for constants c_1 and c_2. Applying the boundary conditions $\rho(0) = a$ and $\rho(\alpha) = b$, we can determine these two constants and find the minimal exposure path as given in Equation (2.3).

Similarly, we can show the following for a more general sense sensitivity function:

```
Initialize(Grid G)                              Search(Grid G)
    Start.Cost = 0                                  Initialize(G)
    Start.Visited = FALSE                           while Finish.Visited = FALSE
    Start.Parent = ∅                                    Let i be the unvisited node with the lowest cost.
    For each node  i in G that is not Start             i.Visited = TRUE
        i.Cost = ∞                                      For each neighbor j of i
        i.Visited = FALSE                                   if i.Cost + Exposure(i, j) < j.Cost
        i.Parent = ∅                                            j.Cost = i.Cost + Exposure(i, j)
                                                                j.Parent = i
```

Figure 1: The functions associated with the centralized minimal exposure algorithm.

Theorem 3. *For the sensor sensitivity function f(s, p) = 1/d^k(s, p), (k ≥ 0, k ≠ 1), the minimal exposure path between two points A and B is given by*

$$\rho(\theta) = a\left(\cos((k-1)\theta) + \sin((k-1)\theta)\frac{(\frac{b}{a})^{k-1} - \cos((k-1)\alpha)}{\sin((k-1)\alpha)}\right)^{\frac{1}{k-1}} \quad (2.7)$$

2.3 Multiple-Sensor Optimal Solution

When there are multiple sensors in the sensor network, or the sensor field is bounded, a closed form solution to the minimal exposure path problem does not exist regardless of the exposure model (*closest sensor* or *all sensor*). This section describes our proposed grid-based approximation method that solves the minimal exposure path problem for sensor network with multiple sensors in a constrained sensor field. The algorithm finds a path whose exposure is within a bounded error of the minimal exposure path. The error is determined by the granularity of the grid.

Figure 1 depicts a centralized method that works similar to Dijkstra's shortest path algorithm. However, applying this algorithm requires that sensor nodes not only to perform the exposure calculation and shortest-path searching in the sensor network, but also know the topography of the network. Both, particularly discovering the network topography, can be costly, in terms of computation and energy consumption. Recall that communication, which is required to discover network topography, in wireless sensor networks is the major energy consumer. Thus, it is important to develop a localized minimal exposure path algorithm so that sensors can estimate the network's minimal exposure path without having to know the entire network's topography. In such a localized algorithm, the number of messages sent across the network and the computation performed at each node should be kept at the minimal level.

To achieve low cost on communication and computation on the wireless sensor network, our localized approximation method necessarily takes the following assumptions: 1) the sensor nodes do not possess the necessary knowledge to compute shortest path locally and thus rely on forwarding messages to their neighbors using a shortest-path heuristic; 2) a sensor node stores topological information it receives and forwards the topological information it knows; and 3) the Voronoi diagram-based minimal exposure path approximation algorithm is used to further reduce the computation (of exposure) at each sensor node. To use the Voronoi diagram in

order to estimate the minimal exposure path, grid points are placed along Voronoi edges, and grid points on the same Voronoi cell are fully connected. The weight of an edge between two Voronoi grid points is the single-sensor optimal solution weight for the sensor corresponding to the Voronoi cell. However, this weight only applies if the shortest path exists entirely within the Voronoi cell. If the path strays from the Voronoi cell, a straight line is used to weight the edges. Note, however, that if one wishes to increase computational complexity, a better approximation can be made using the grid-based centralized method and limiting the path to exist within the Voronoi cell. Furthermore, the single-sensor optimal solution is used to bound areas to search; if the single-sensor optimal solution between two points is larger than an already-found estimated solution, those two points are not investigated during subsequent iterations of the localized algorithm.

The localized minimal exposure algorithm considers the following problem: Given a sensor network *S* with an exposure model *E*, a starting coordinate, and a finishing (or ending) coordinate in a grid imposed on the sensor field, find a minimal exposure path from the starting location to the ending location.

Two types of messages, *Forward* messages and *Search* messages, are passed among sensors in the sensor network. *Search* and *Forward* messages have the same format except that the receiving node will take different action, either search locally or forward to a neighboring sensor as the names suggest, on the receipt of the message.

A sensor node selects its neighboring node that with the largest heuristic value as the recipient of the message. On one hand, one would pick the node that has potentially large number of distinct neighbors so it can quickly learn the network topography. This can be naturally measured by the distance between the sender and the recipient. On the other hand, one would like the recipient to be close to the destination (i.e., the ending location). We combine these two factors and use the following formula to calculate the heuristic value for node *j* with respect to the sender node *i*:

$$\left(\frac{1}{h^k}\right)\frac{D(i,j)}{R} + \left(1 - \frac{1}{h^k}\right)\left(1 - \frac{D(j, Finish)}{D(i, Finish)}\right) \quad (2.8)$$

where *D(i, j)* is the distance between the sender *i* and its neighbor *j*, *R* is the maximum communication radius, *h* is the number of hops that the message has currently been transmitted (assume *h* starts with 1), and *k* is a positive constant. To balance the above two unrelated values, we normalize both as indicated in Equation (2.8), where the first term rewards nodes that are far away from

```
Search*(Grid G)
    Initialize(G)
    while Finish.Visited = FALSE
        Let i be the unvisited node with the lowest cost.
        if i is not on known Voronoi Border
            i.Visited = TRUE
            For each neighbor j of i
                if i.Cost + Exposure(i, j) < j.Cost
                        j.Cost = i.Cost+Exposure(i, j)
            j.Parent = I
```

```
GetBestPotentialNode(Grid G)
    BestP  = ∞
    Best = ∅
    For each node i in G
        P  = i.Cost + Worst(i, Finish)
        if (P < BestP)
            BestP = P
            Best = i
    return Best
```

```
Worst(Node N, Node Dest)
    WorstValue = 0
    For each sensor i in S
        if Exposure(i, Dest) > WorstValue
            WorstValue = Expsoure(i, Dest)
    return WorstValue

ForwardMessage(Node N, SensorNetwork S)
    if N is closest to destination
        SearchMessage(Finish, S)
    else
        Next = GetBestHeuristicNeighbor(N)
        ForwardMessage(Next, S)
```

```
SearchMessage(Node N, SensorNetwork S)
    if N is closest to destination
        Let G be a Voronoi-Based Grid
        Search*(G)
        Next = GetBestPotentialNode(G, S)
        if Next ≠ ∅
            ForwardMessage(Next, S)
    else
        Next = GetBestHeuristicNeighbor(N)
        SearchMessage(Next, S)
```

Figure 2: The functions associated with the localized minimal exposure algorithm.

the sender and the second term penalizes neighbor *j* if it is further to the destination than the sender node *i*, the case when the second term will be negative). The constant *k* reflects how rapid the weight is shifted from picking a neighbor far away from a sensor to picking a neighbor close to the destination. One can see that we tend to pick a sensor closer to the destination as *h*, the number of hops, increases to prevent the message from being circulated endlessly throughout the sensor network. Our experimentation suggests that a good value for *k* is 0.1.

We now outline our Voronoi-based localized approximation algorithm whose main functions are given in Figure 2. First, the sensor that is the closest to the starting coordinate sends a *Search* message to the node that is determined based on the above heuristic value (Equation (2.8)). When this *Search* message reaches its destination sensor (the sensor closest to the ending coordinate), the sensor calculates the minimal exposure path using a Voronoi-based approximation algorithm and the network's topological information it receives. The Voronoi-based approximation algorithm gives the near-optimal exposure path within a Voronoi cell without using the computationally expensive grid-based method and hence reduces the computation requirement. The algorithm then selects the sensor in the location that is most promising to contain the minimal exposure path and sends a *Forward* message there. When the appropriate sensor receives the *Forward* message, it sends a *Search* message back to the destination sensor to acquire more information on sensor network's topography that is needed by the Voronoi-based approximation algorithm. This process repeats until no further

topological information is required by any sensor node or no locations look promising when compared to the current minimal exposure path calculated.

We only need to elaborate the process of finding the sensor that will be the recipient of the *Forward* message. To determine such a node, we first superimpose grid points onto known Voronoi edges. Grid points that are part of the same Voronoi cell are connected via an edge, and the weight of this edge is determined by the single-sensor minimal exposure path weight between the two points (if the closed formula applies, otherwise a straight-line approximation is used). Then, a shortest-path search is calculated from the starting coordinate to the ending coordinate using the edges and weights defined by the superimposed grid point. If the shortest-path algorithm attempts to select a grid point on a Voronoi edge that only one of its two defining sensors has been visited, that node is marked as *potential* and will not be searched any further (because no more Voronoi cells lie in that direction). The best-weighted path from this point to the destination is calculated and stored. To determine the best-weighted path for a grid point, one determines the exposure of the known path to the grid point and estimates the exposure to the destination. To estimate the exposure to the destination, we select the sensor that, given only this sensor in the sensor network, the path between the grid point and the destination would have the most weight. Adding these two values yields the best-weighted minimal exposure path approximation, and if this approximated path is larger than the currently known minimal exposure path, the corresponding Voronoi grid point is ignored. For each *potential*
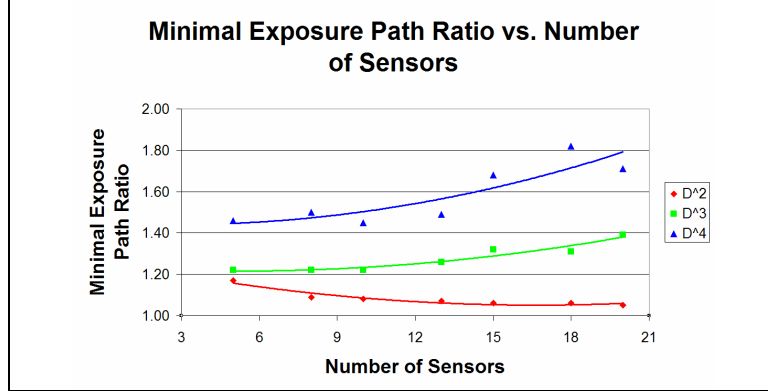
Figure 3: Experimental results for the minimal exposure path algorithm.

grid point a best-weighted path is calculated, and if the best *potential* grid point has a weight lower than the currently known minimal exposure path (that lies completely within the known Voronoi cells), a *Forward* message is sent to the closest sensor to that grid point (which will in turn search for other Voronoi cells near that grid point). If all *potential* grid points have best-weighted paths with more exposure than the current minimal exposure path, the search is terminated and the current minimal exposure path is considered the final result.

2.4 Experimental Results

The goal of Figure 3 is to show how good an approximation our algorithm can achieve with respect to the exposure model and the number of sensors. To acquire the data presented in Figure 3, we ran 1000 tests for each of the 21 different cases. The 21 cases represented each possible combination of the number of sensors (5, 8, 10, 13, 15, 18, and 20) and the exposure models ($1 / d^2$, $1 / d^3$, $1 / d^4$) used. For each case, the 1000 tests were average and compared to the actual minimal exposure path as determined by the centralized optimal numerical approximation method described earlier. This number is called the *minimal exposure ratio*, and is the approximated path's exposure value divided by the minimal exposure path's exposure value. As one can see, the closer to one this value is the better the approximation.

Looking at each exposure model separately, from the $1 / d^2$ exposure model, we see that as the number of sensors increases, the accuracy of the approximation increases until some particular number of sensors threshold is reached. After that threshold, the approximation tends to worsen as the number of sensors increases. This behavior is due to the fact that we use a fixed grid size; in other words, the grid points placed on the Voronoi diagram in the approximation are placed at a specific distance L apart. Thus, when the number of Voronoi cells becomes high enough (and thus the sensor network dense enough), the grid size is no longer accurate enough to adequately approximate the sensor network. With respect to the $1 / d^3$ exposure model, a similar scenario occurs except that the threshold number of sensors appears to have decreased, thus making the approximation algorithm less accurate sooner. Similarly, this threshold appears to have decreased again when changing the model to $1 / d^4$.

Further analysis comes from comparing how the exposure models compare amongst each other. From Figure 3, one can see that the $1 / d^2$ is the most accurate and the $1 / d^4$ is the least accurate. This is due to the fact that the single-sensor minimal exposure path closed formula is not as applicable in the $1 / d^4$ case because the path travels outside the appropriate Voronoi cell more often than in the $1 / d^2$ case. Figure 4 shows how the centralized-optimal minimal exposure path compares with the localized approximation using the single-sensor optimal solution.

To further evaluate our localized minimal exposure path algorithm, we also examine the number of messages sent. This graph, however, is uninteresting, and the results are as expected. The number of messages sent depends on the communication radius, and the larger the radius, the less communications needed. Furthermore, on average, the number of communications required is quite small (approximately an order of magnitude less for networks with a large number of sensors).

3. Maximal Exposure

Given a sensor network with an exposure model, the maximal exposure path between a starting location and an ending location is a path in the sensor connecting the two points such that the exposure received from traveling the path is maximal. We necessarily take one or more of the following assumptions: 1) there is a maximum length constraint on the maximal exposure path so the object cannot keep on moving in the sensor field to accumulate infinite exposure; 2) there is a delay, or maximal time, constraint before which the path has to reach the ending location to prevent the object from staying in the sensor field forever; 3) there is a maximal velocity at which the object can travel (which prevents trivial solutions: jump to the point that has the largest exposure, stay until time expires, and then jump to the ending point); 4) there is an upper bound for the exposure at a given point or there is a minimum distance that the object has to keep away from the sensor node to avoid the sensor sensitivity function to explode to infinity. This last assumption can be imposed to the definition of sensor sensitivity function. In this section, we will show that the maximal exposure path problem is NP-hard and then propose a set of heuristics and compare their efficiency.
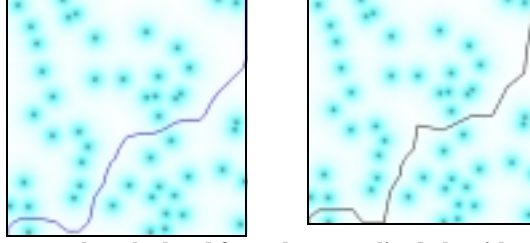
Figure 4: Comparison of minimal exposure paths calculated from the centralized algorithm (*left*) and the localized algorithm (*right*) using a sensor network with 50 sensors and an exposure model of (1/d²).

Theorem 4. *The maximal exposure path problem is NP-hard.*

Proof:

We consider a special case of the maximal exposure path problem and show that it can be reduced to the NP-complete longest path problem [7].

We first restrict the path to be in the Manhattan space (a grid) by defining the exposure function to be positive along the grid and zero otherwise. For each grid point, we associate it with a vertex. Two vertices are connected by an edge if there is a line segment in the grid between the two grid points. The weight of an edge equals to the exposure received from traveling along this line segment. Note that this weight function is independent of the direction of the edge and therefore we construct an undirected weighted graph with positive edge weights. The maximal exposure path from one grid point s to another grid point t has a total exposure that equals to the total edge weights along the path. If we treat the edge weight as its length, finding the maximal exposure path becomes finding the longest path in the undirected graph, which is known as NP-complete.

Before we elaborate the proposed heuristic methods to find a path with large, if not optimal, exposure, we make the following observations about any maximal exposure path:

- When there is a delay constraint on when the destination should be reached, the optimal exposure path should arrive exactly on the deadline. This is because that if there is any time "left over" when the destination is reached, we can always delay the arrival by staying inside the sensor field to accumulate more exposure. This is true because the sense sensitivity function is non-negative.

- Similarly, when there is a maximal path length constraint, the optimal exposure path should always have exactly that maximal length. Otherwise, one can move off the path and then come back to the path at the point it departs. As long as the maximal length constraint is not violated (and the sense sensitivity function is not negative), this gives more exposure.

- If there is any stop along the optimal exposure path, the stop must be made at the point that has the largest exposure value along the path. Stopping at anywhere else will clearly yield a path with less exposure.

- If there is a maximum velocity constraint on how fast one can travel in the sense field, the optimal exposure path achieves the maximal exposure if and only if one either travels at the maximum speed or stops. If one travels a path with a slower speed, one can always speed

up and use the saved time to stop at the point along the path with the largest exposure value. This apparently increases the total exposure along the same path.

- Finally, given a deadline constraint and a maximum velocity at which one can travel, any valid path that can reach the destination before deadline is contained in an ellipse. Note that the ellipse is defined as: *given two foci f_1 and f_2, the sum of the distances from any point on the ellipse to the two foci f_1 and f_2 is a constant.* We can consider the starting and ending locations as the two foci and the constant in the above definition is the maximal length one can travel with the maximum speed at the given time slot. Furthermore, this is true at any point of the path, not only the starting location. We can use the remaining time as the deadline to reach the destination from our current position. This greatly reduces the solution space we need to search for the optimal exposure path.

3.1 Random Path Heuristic

The first, and perhaps the simplest, heuristic that can be used to find a path that has a decent exposure value is to use the *random-path* heuristic. In this method, a random path is created according to a particular model. This model states that the node on the shortest path to the destination is selected with some percentage of the time, and random node is selected otherwise. Nodes on the shortest path are selected to increase the chance that we can reach the destination before time expires. Random nodes are also selected to allow the path to go off the shortest path to collect more exposure. If selecting a random node makes the path too long (that is, the node is outside the ellipse defined by the current location, the destination, and the remaining time as we described earlier. As a result, one won't reach the destination in the required amount of time), the node en-route to the shortest path is used instead of the random node. The advantage of the *random-path heuristic* is that it does not rely on the topography of the network or the exposure model, and hence it is simple to implement and computational inexpensive. Note that we do not compute the ellipse and restrict the selection of the random nodes to be within such ellipse to keep the computation requirement of the algorithm low. The disadvantage of the *random-path heuristic* is that its performance, i.e., the total exposure of the obtained path, is unpredictable. In sum, this method can be used in the case when we need to find a path fast with low computation cost.

3.2 Shortest Path Heuristic

The *shortest path heuristic* reports the shortest path between the starting and ending points as the maximal exposure path. To achieve the maximal exposure on this path, we travel at the
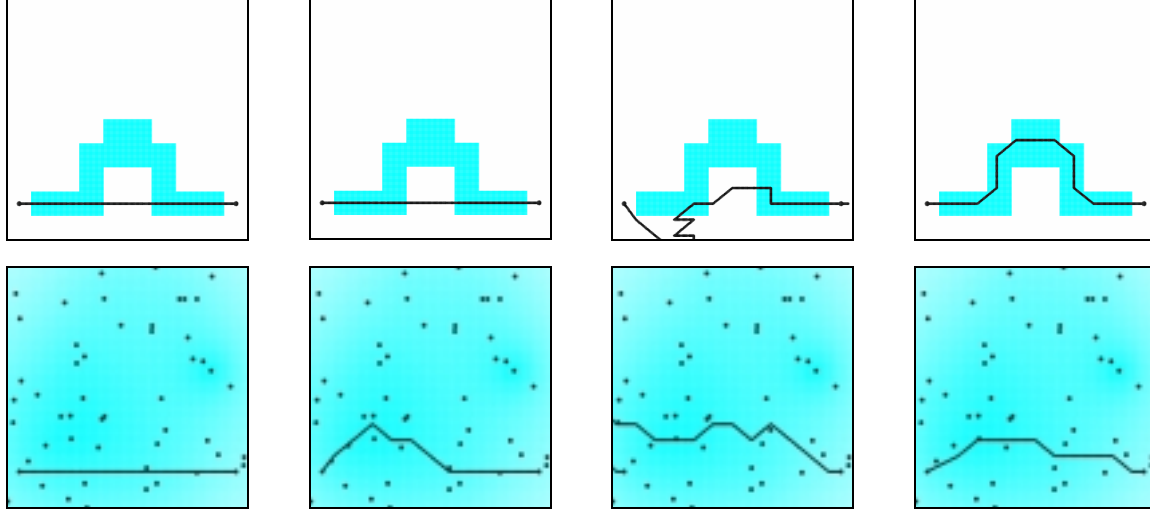
Figure 5: Demonstration of the four heuristics (from left to right: shortest path, best point, random path, adjusted best point) on a discrete (top) and continuous (bottom) exposure model. Note that in the best point heuristic for the continuous case is not two straight lines; this is due to the grid resolution used to generate these examples. As the grid resolution increases, best point solution will approach two straight lines towards the best point.

maximum speed, stop at the point with the highest exposure value along the path, and reach the ending point on the deadline. Again, this heuristic is easy to implement and it has in general better performance and predictability than the random path heuristic, although both are not very good. The shortest path method does require certain knowledge about the sensor network, for example to compute the shortest path. It is also independent of the sensor exposure model. The solution is optimal if we want to keep the path short. However, it may not always yield a good solution because it does not leave the shortest path to explore other fields that may have higher exposure. For example, if the starting and ending locations are fairly close and both are in a region with very low exposure, then the shortest path heuristic gives poor solution. This is particularly true when the exposure model $1/d^k$ has a large k, where exposure increases rapidly as one moves closer to a sensor node instead of staying far away on the shortest path to the destination.

3.3 Best Point Heuristic

Because taking the shortest path to a point does not necessarily yield the highest exposure values, it may also result in poor solutions under some circumstance. In the *best-point method*, we superimpose a grid over the ellipse that defines the search space as we have discussed earlier. We then find the shortest path to each grid point from the start and the destination. Next, we compute the total exposure of the path combining two shortest paths that share the same grid point, where we stop at the point with the highest exposure if there is any "leftover" time. The path that gives the overall best exposure is reported as the optimal exposure path. Clearly, the quality of the solution from this method depends on the granularity of the grid and is almost certainly better than the previous two heuristics. Using the shortest paths to connect the starting and ending points with the grid point ensures that the largest possible amount of time will be spent at that point, thus aiming to maximize the amount of exposure. However, compared to the previous two heuristics, this method is more computationally expensive, as it has to compute

the exposure along all the paths connecting the starting and ending points and each grid point.

3.4 Adjusted Best Point Heuristic

The best-point heuristic cannot guarantee the quality of the solution because it assumes that the optimal exposure path consists of two shortest paths. The *adjusted-best-point method* improves this by considering paths that consists of multiple shortest paths. Figure 5 gives an example where the best-point method reports the same path as the shortest path method. It also shows a random path and one obtained by the adjusted-best-point method, which has more exposure than the rest paths.

For a given path on a sequence of nodes where each two adjacent nodes are connected by the shortest path, we define three basic actions to adjust the path: moving a node, adding a node, and deleting a node.

Moving a node means that we change the position of the node while preserving its order in the sequence of the nodes, and the shortest paths between this node and its two adjacent nodes are changed accordingly. The rest of the path remains unchanged. Each node in the original path is considered as a potential node to move. The position to where a node is moved is selected from its neighboring grid points with higher exposure values.

To add a node, a pair of adjacent nodes is selected and a new node is added between them. The shortest path between the two previously adjacent nodes is replaced by two shortest paths that connecting them with the newly added node respectively. To keep the algorithm simple, we consider the neighboring grid points of the two nodes as candidates for the new node to add. In particular, the one with the largest exposure value is selected.

Deleting a node involves removing the selected node and the two shortest paths connecting it to its two adjacent nodes, as well as adding the shortest path to connect these two nodes. To determine whether a node should be deleted from the path, we evaluate the exposure before and after the deletion to see whether there is any gain in exposure by deleting the node. This can be computed

Table 1: Comparison of maximal exposure path heuristics for varying numbers of sensors and exposure models.

Exposure Model = $1 / d^2$				
Number of Sensors	Adjusted Best Point	Best Point	Shortest Path	Random Path
5	1.05×10^{-1}	1.02×10^{-1}	3.42×10^{-2}	6.39×10^{-2}
10	1.21×10^{-1}	1.15×10^{-1}	4.10×10^{-2}	7.74×10^{-2}
13	1.12×10^{-1}	1.09×10^{-1}	4.00×10^{-2}	7.41×10^{-2}
15	1.11×10^{-1}	1.09×10^{-1}	3.95×10^{-2}	7.44×10^{-2}

Exposure Model = $1 / d^3$				
Number of Sensors	Adjusted Best Point	Best Point	Shortest Path	Random Path
5	7.69×10^{-3}	1.44×10^{-3}	2.07×10^{-4}	3.94×10^{-4}
10	2.34×10^{-3}	1.15×10^{-3}	1.10×10^{-4}	2.23×10^{-4}
13	2.45×10^{-3}	2.01×10^{-3}	2.54×10^{-4}	4.34×10^{-4}
15	7.27×10^{-3}	4.96×10^{-3}	2.55×10^{-4}	5.30×10^{-4}

Exposure Model = $1 / d^4$				
Number of Sensors	Adjusted Best Point	Best Point	Shortest Path	Random Path
5	1.77×10^{-4}	1.44×10^{-4}	3.09×10^{-5}	1.07×10^{-5}
10	7.56×10^{-5}	6.45×10^{-5}	2.84×10^{-5}	2.81×10^{-6}
13	2.44×10^{-4}	1.07×10^{-4}	8.29×10^{-5}	5.97×10^{-6}
15	1.64×10^{-4}	7.40×10^{-5}	1.62×10^{-5}	1.84×10^{-5}

efficiently as for each candidate node, we only need to calculate the exposure along three shortest paths: two connecting the candidate node to its adjacent nodes and one that is going to be added to connect those two adjacent nodes.

These three adjustment actions can be taken iteratively and in any order until we cannot make further improvement of the path's exposure.

3.5 Experimental Results

Figure 5 provides a visual comparison between the different heuristics used to approximate the maximal exposure path. Because the maximal exposure path is NP-hard, in this section we hope to show how well our adjusted-best-point algorithm fares with respect to other heuristics and a derived upper bound. Before discussing the results, we will now describe how the upper bound was calculated. The upper bound was calculated by starting at the specified starting point and checking radii of increasing length until the entire ellipse defined by the maximum length, starting point, and destination point was reached. At each different radii, the point of maximal exposure located on that radii was determined, and these calculated exposure values were summed to yield an upper bound on the maximal exposure path. In essence, this upper bound is calculated by relaxing the constraints on the maximal path such that the points used are not necessarily connected via an edge.

The data values in Table 1 represent the maximal exposure path calculated for the specified scenario divided by the upper bound. Because the upper bound used for comparing the maximal exposure path heuristics is not tight, the values these heuristics determine are much lower than the upper bound. However, it is important to note that for each scenario in Table 1, the adjusted-best-point heuristic outperforms the other three heuristics.

4. Linear Programming Techniques

Linear programming is a powerful tool for solving optimization problems. It takes the form of a set of linear constraints, which defines the solution space for a given problem, accompanied by an objective function, which is used to evaluate and choose the best solution. Due to the wide range of applications of linear programming, there exist well-developed tools to solve the linear programming problem. In this section, we simply formulate both the minimal and maximal exposure path problems as linear programming problems.

Given a sensor network S with an exposure model E, we superimpose a grid containing n grid points (also referred to as nodes) over the sensor field and define the following:

For each grid node i, we define a binary variable X_i, which takes value 1 if and only if the node is on the path.

For each edge between nodes i and j, and the line segment in the grid between the two nodes, we define a binary variable E_{ij} that takes value 1 if and only if this edge is in the path.

For a given pair of starting and ending coordinates, the binary variable S_i takes the value of 1 if and only if node i is the starting node; F_i takes the value of 1 if and only if node i is the ending node.

The positive value $Exposure(i, j)$ is the exposure acquired by traveling from node i to node j. We define C_{ij} (the "cost" from node i to node j to be $Exposure(i, j)$ if nodes i and j are neighbors, otherwise, the cost from i to j will be ∞.

Now, a simple path in the sensor field can be defined as any assignments to the binary variables $\{X_i,\ E_{ij},\ S_i, F_i\}$ satisfying the following:

$$\forall i : S_i + F_i \leq 1 \tag{4.1}$$

$$\sum_{i=1}^{n} S_i = 1 \tag{4.2}$$

$$\sum_{i=1}^{n} F_i = 1 \tag{4.3}$$

$$\forall i : X_i - S_i - F_i \geq 0 \tag{4.4}$$

$$\forall i : S_i + F_i + \sum_{j=1}^{n} E_{ij} - 2X_i = 0 \tag{4.5}$$

where (4.1) implies that node i cannot be both the starting node and the finishing node of the path, (4.2) and (4.3) enforce that there is exactly one starting/finishing node in the path, (4.4) includes both the starting and finishing nodes into the path, (4.5) restricts the path to be a simple path – namely no node or edge can be included in the path more than once. In fact, every node in the path (with the exception of the starting/finishing nodes) has exactly two edges connecting it to other nodes. The exposure along the path is the sum over the exposures of all the edges in the path, which can be expressed as

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}C_{ij}E_{ij} \qquad (4.6)$$

The minimal exposure path problem can be conveniently formulated as minimize (4.6), subject to (4.1)-(4.5). To find the minimal exposure path between two given nodes, we can simply replace the sum on left hand side of Equations (4.2) and (4.3) by these two nodes.

The maximal exposure path problem can be formulated exactly the same fashion by simply changing the objective from minimize (4.6) to maximize (4.6). However, as we have discussed earlier, there are normally additional constraints associated with the maximal exposure path problem. Define D_{ij} to be the length of the edge between nodes i and j if they are neighbors in the sensor network. The maximal length, L, constraint can be expressed as

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}D_{ij}E_{ij} \leq L \qquad (4.7)$$

One may stop at some nodes to accumulate exposure, let T_i be the time that the path halts at node i and T_{ij} be the time to traverse the edge between nodes i and j, then the delay constraint T, or maximal time that one can stay in the sensor field before reaching the finishing node, becomes

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}T_{ij}E_{ij} + \sum_{i=1}^{n}T_i \leq T \qquad (4.8)$$

$$\forall i: \quad T_i - TX_i \leq 0 \qquad (4.9)$$

$$\forall i,j: \quad T_{ij} - TE_{ij} \leq 0 \qquad (4.10)$$

where (4.9) ensures that one cannot stop at a node that is not on the path. That is, if $X_i = 0$, then $T_i = 0$. Similarly, (4.9) guarantees that $T_{ij} = 0$ for the edges that are not along the path, $E_{ij} = 0$. Finally, let C_{ii} be the exposure at node i. Finally, the total exposure along a path becomes

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}D_{ij}E_{ij} \leq L \qquad (4.11)$$

The practicality of the linear programming method is not immediately obvious. In fact, attempting to solve entire minimal or maximal exposure path problems with this formulation yields a significant amount of variables and constraints. However, the linear programming formulation can be applied when the area being searched is uniform, and thus, by using a sparse grid, computationally tractable. Furthermore, the linear programming model can be used to generate optimal solutions for small areas of the network, and these locally optimal solutions can be combined (for example, by using a Dijkstra-style algorithm) to create a global approximation. Moreover, in scenarios where the single-

sensor optimal solution does not apply (i.e. the path travels outside the Voronoi cell), the linear programming model can be used to determine the optimal path through the cell. Another use for the linear programming model is to simplify it by removing variables and constraints. With proper simplification, the linear program can become tractable and thus provide a bound by which we can compare our solutions, which is particularly important for the maximal exposure path formulation.

5. Related Work
There exists a plethora of work to which we are related. This work consists of topics that primarily include coverage in sensor networks (breach, support, and minimal exposure paths, and so forth) and linear programming methods.

The work done by Megerian et. al. in [10] shares similarities to the work presented in this paper. In particular, [10] hopes to classify and categorize sensor networks using the maximal breach path and maximal support path. The maximal breach path solves a problem similar to the minimal exposure path with one key difference. The maximal breach path finds a path such that at any time, the exposure no more than some particular value. Furthermore, the maximal breach path hopes to minimize this particular value. In contrast, the minimal exposure path does not focus on exposure at one particular time, but tries to minimize the exposure acquired throughout the entire time in the network. The maximal support path is somewhat analogous to the maximal exposure path. The goal of the maximal support path is to find a path such that at any given time, the exposure on the path is no less than some particular value, and this value should be maximized. However, the maximal exposure path is slightly different in that again, we do not focus on one particular time but instead aim to take into consideration all the time spent traversing the path. Both works, however, attempt to classify a sensor network by bounding it with some minimal and maximal value related to exposure.

The work done in [10] is similar to our work in that both works attempt to provide a localized means of determining the minimal exposure path. However, the main difference between our works lies in how we calculate the minimal exposure path. In [10], the minimal exposure path is calculated using a grid-based method inside of the area defined by Voronoi cells, and this information is passed between nodes. In our method, we use Voronoi cells in conjunction with the closed-form solution. Doing this enables us to use less gird points and thus less computation by placing the grid-points only on the border of the Voronoi cell, instead of throughout the entire Voronoi cell.

In [4], Deb et. al. present an algorithm they call STEM – Sensor Topology Extraction at Multiple Resolutions. The goal of this algorithm is to tradeoff topological details with resources expended. In other words, STEM extracts the topology of a sensor network at different levels of detail depending on the amount of resources that are allowed to be used. The work by [4] is related to ours in that both works attempt to determine some particular amount of topology, the way by which it is done differs. Our means of evaluating neighbor nodes reduces down to a formula that attempts to find the a neighboring node with the most distinct neighbors and is closest to the destination. The means of acquiring topology in [4] is through the use of the Minimal Virtual Dominating Set. To find the Minimal Virtual Dominating Set, one must partition the network into red and black nodes, with

each red node being connected to at least one black node. Red nodes do not forward information, and the Minimal Virtual Dominating Set is the set of nodes colored black.

The work by Dhillon et. al. in [5] is related to our work in that [5] tries to construct a sensor network that achieves a particular minimum coverage. To construct the network, [5] assumes that the sensors are placed in a grid covering a specified area. The goal of [5] is to turn on the minimum number of sensors such that all points in the grid are covered with some minimal amount of coverage. Our work is related to this in that one aspect of our work is a means by which one can approximate this minimum coverage.

The work done by [9] shows how to use a distributed algorithm to optimally solve the best-coverage problem described in [10]. The best-coverage problem is also known as the maximal support path, and the Delaunay triangulation can be used to solve this best-coverage problem optimally. Our work is related to that of [9] in that we also aim to provide a best-case scenario similar to the best-coverage problem. This best-case scenario is called the maximal exposure problem, and is a different means of classifying a sensor network.

The goal of Couqueur et. al. in [1] is to provide a strategy for deploying sensors such that a target being tracked is subject to some minimal amount of exposure. Furthermore, [1] attempts to maximize this minimal amount of exposure. In addition, [1] also minimize the cost of deploying sensors by deploying and appropriate number of sensors at each step. Our work is similar to this in that both deal with a worst-case scenario with regards to path exposure. However, our work also deals with a best-case scenario with regards to path exposure.

In the last two decades combinatorial optimization in general and integer and linear programming in particular, attracted a great deal of attention. Linear programming and in particular simplex algorithm for solving it were introduced and greatly popularized by Dantzig in 1947. Dantzig's book is still considered as landmark text and must reading. In addition, numerous other textbooks are available, including [1], [3].

6. Conclusion

We introduced our algorithms by beginning with the minimal exposure path algorithm and introducing a new coverage method called the maximal exposure path. The method of using the minimal coverage to quantify a sensor network was solved using several methods including a numerical optimization approach and a linear programming approach. Furthermore, we developed a localized approximation algorithm to enable a sensor network to determine its minimal exposure path. The second coverage problem - the maximal exposure path - was proven to be NP-hard, and thus, we provided heuristics that can be used to generate approximate solutions.

To determine the effectiveness of our localized minimal exposure path algorithm, we compared the results of our algorithm to those produced by the optimal centralized minimal exposure path algorithm. Our results show that our approximation for the minimal path in a sensor network with an exposure model of $1/d^2$

is only 20% sub-optimal. This sub-optimality increases as the exponent in the exposure model increases due to the fact that the single-sensor minimal exposure path solution used in the approximation algorithm no longer stays within the corresponding Voronoi cell.

We also provided heuristics to estimate the maximal exposure path, and have provided an upper bound on the weight of the maximal exposure path. We have shown that our adjusted-best-point algorithm outperforms the other heuristics, including the best-point algorithm, shortest-path algorithm, and the random-path algorithm.

7. Acknowledgements

8. References
[1] Chvatal, V. *"Linear Programming."* W.H. Freeman 1983.

[2] Couqueur, T., Phipatanasuphorn, V., Ramanathan, P., Saluja, K. K. *"Sensor Deployment Strategy for Target Detection."* Proceedings of The First ACM International Workshop on Wireless Sensor Networks and Applications 2002.

[3] Dantzig, G. B. *"Linear Programming."* Springer, 1997.

[4] Deb, B., Bhatnagar, S., Nath, B. *"Multi-Resolution State Retrieval in Sensor Networks."* SNPA, ICC 2003.

[5] Dhillon, S. S., Chakrabarty, K., Iyengar, S. S. *"Sensor Placement For Grid Coverage Under Imprecise Detections."* FUSION 2002.

[6] Estrin, D., Govindan, R., Heidemann, J. *"Embedding the Internet."* Communications of the ACM 2000.

[7] Garey, M.R. and Johnson, D.S. *"Computers and Intractability: A Guide to the Theory of NP-Completeness."* W.H. Freeman and Company, 1979.

[8] Gelfand, M. Fomin, S. *"Calculus of Variations."* Prentice-Hall Inc. 1963.

[9] Li, X., Wan, P., Frieder, O. *"Coverage Problems in Wireless Ad-Hoc Sensor Networks."* IEEE Transactions for Computers 2002.

[10] Megerian, S., Koushanfar, F., Qu, G., Potkonjak, M. *"Exposure in Wireless Sensor Networks."* MobiCom 2001.

[11] Megerian, S., Koushanfar, F., Potkonjak, M., Srivastava, M. *"Coverage Problems in Wireless Ad-Hoc Sensor Networks."* IEEE Infocom 2001.