Bidding Protocols for Deploying Mobile Sensors

Guiling Wang, Guohong Cao, Piotr Berman and Thomas F. LaPorta
Department of Computer Science & Engineering
The Pennsylvania State University
University Park, PA 16802
Email: {guiwang,gcao,berman,tlp}@cse.psu.edu

Abstract

Adequate coverage is very important for sensor networks to fulfill sensing tasks. In many working environments, it is necessary to make use of mobile sensors to provide the required coverage. We propose to deploy a mix of mobile and static sensors to achieve a balance between sensor coverage and sensor cost. We design two bidding protocols to guide the movement of mobile sensors. In the protocols, static sensors detect coverage holes locally by using Voronoi diagrams, and bid mobile sensors to move. Mobile sensors accept the highest bids and heal the largest holes. Simulation results show that our protocols achieve suitable tradeoff between coverage and sensor cost.

I. Introduction

Wireless sensor networks greatly enhance our ability to monitor and control the physical environment. Sensor networks are revolutionizing the traditional methods of data collection, bridging the gap between the physical world and the virtual information world [18], [23], [31], [34]. Sensor nodes must be deployed appropriately to reach an adequate coverage level for the successful completion of the issued sensing tasks [5], [27].

In many potential working environments, such as remote harsh fields, disaster areas and toxic urban regions, sensor deployment cannot be performed manually. To scatter sensors by aircraft is one possible solution. However, using this technique, the actual landing positions cannot be controlled because of the existence of wind and obstacles, such as trees and buildings. Consequently, the coverage may be inferior to the application requirements no matter how many sensors are dropped. Moreover, in many cases, such as during in-building toxic-leaks [20], [21], chemical sensors must be placed inside a building from the outside. In these scenarios, it is necessary to make use of mobile sensors which can move to the correct places to provide the required coverage. One example of a mobile sensor is the Robomote [33]. These sensors are smaller than 0.000047m^3 and cost less than 150 dollars.

Most previous research efforts on deploying mobile sensors are based on centralized approaches. For example, the work in [36] assumes that a powerful cluster head is available to collect the sensor locations and determine the target locations of the mobile sensors. However, in many sensor deployment environments such as disaster areas and battle fields, a central server may not be available. It may also be hard to organize sensors into clusters due to network partitions. Further, centralized approaches introduce a single point of failure. Sensor deployment has also been addressed in the field of robotics [20], where sensors are deployed iteratively one by one, utilizing the location information obtained from the previous deployment. Since sensors are deployed one by one, the deployment time is very long which can significantly increase the network initialization time.

Most previous research efforts on deploying mobile sensors assume that all the nodes are mobile [16], [36], [20], [21]. However, to equip every sensor with a motion base increases the network cost and is unnecessary when the coverage requirement is not very strict, or if sensors can be scattered in the target field relatively uniformly. We propose to deploy a mixture of mobile and static sensors to construct sensor networks, such that a balance between sensor cost and coverage can be achieved.

In this paper, we design two distributed bidding protocols for the placement of mobile sensors in a sensor network composed of both mobile and static sensors: a basic bidding protocol and a proxy-based bidding protocol, which is an improvement on the basic bidding protocol. In the protocols, mobile sensors are treated as *servers* to heal *coverage holes*. Coverage holes are locations not covered by any sensor. Each mobile sensor has a *base price*, which is related to the size of any new hole generated by its movement. This represents the cost of its movement in terms of coverage. Static sensors detect coverage holes locally and estimate their sizes as *bids*. The static sensors *bid* the mobile sensors that have a base price lower than the hole to be covered. In the *basic bidding protocol*, mobile sensors choose the highest bids and thus move to heal the largest coverage holes. Using this process, sensors will only move to cover holes larger than those generated by their movements. After moving to the holes, mobile sensors raise their base prices to reflect the new coverage cost, and re-enter the bidding process. This process iterates until no static sensor can give a bid higher than the base price of any mobile sensor.

To reduce the moving distances of mobile sensors, the *proxy-based* bidding protocol proposes that mobile sensors perform virtual movements from small holes to large holes and only perform physical movements after the final destinations are identified. Simulation results show our protocols can achieve high sensor coverage at low cost.

The rest of the paper is organized as follows. Section II introduces some preliminaries. Section III analyzes this sensor placement problem from theoretical perspective. We present the basic bidding protocol in section

IV, and present the proxy-based bidding protocol in section V. Section VI evaluates the performance of the proposed protocols. We conclude the paper in section VIII.

II. Preliminaries

In this section, we present the necessary background on localization techniques, path planning, the sensing model and Voronoi diagrams.

A. Localization Techniques

Location awareness is important for wireless sensor networks since many applications such as environment monitoring and target tracking depend on knowing the locations of sensor nodes. Due to the ad hoc nature of such networks, each node must determine its location. For outdoor systems, Global Positioning System (GPS) [3] is one method for this purpose. Because GPS may not be cost effective or work well indoors, other techniques have been proposed to enable each node to determine its location indoors with only limited communication with nearby nodes. Most of these methods exploit received signal strength [28], time difference of arrival of two different signals [32], and angle of arrival [7]. Hu et al. [22] have provided detailed discussion of these techniques. In the subsequent discussions of this paper, we assume that sensor nodes know their locations.

B. Path Planning

In systems that exploit mobile sensors, finding paths on which these mobile sensors can move to desired destinations, especially when there exist obstacles in the field, is an important problem. The problem has been studied in the area of robotics [6], [24]. Recently, Li et al. [25] studied the problem in sensor networks. They combined the above methods to find the best motion path and to exploit the distributed nature of sensor networks. In this paper, we do not study this problem further; we assume that mobile sensors can move to any location where they are asked to move based on the existing techniques. We comment more on the impact of this assumption in Section VI.

C. Sensing Model

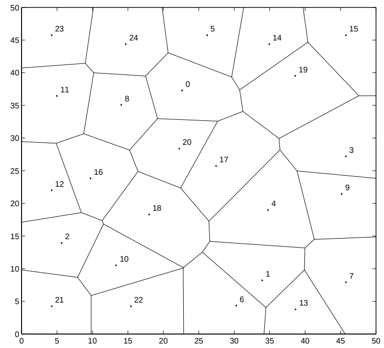
Each type of sensor has unique sensing model characterized by its sensing area, resolution and accuracy. The sensing area depends on multiple factors such as the strength of the signals generated at the source, the distance between the source and the sensor, the attenuation rate in propagation, and the desired confidence level of sensing. Let us consider an application [10] in which a network of acoustic sensors is deployed for detecting mobile vehicles. Due to signal attenuation, sensors closer to a vehicle can detect higher strength of acoustic signals than sensors farther away from the vehicle, and thus have higher confidence for detecting the vehicle. Therefore, given a confidence level, we can derive a sensing range surrounding each sensor. In this paper, we only consider isotropic sensing models. Each sensor node is associated with a sensing area which is represented by a circle with the same radius. This is a common assumption when comparing algorithms for sensing coverage [26], [27].

D. Voronoi Diagram

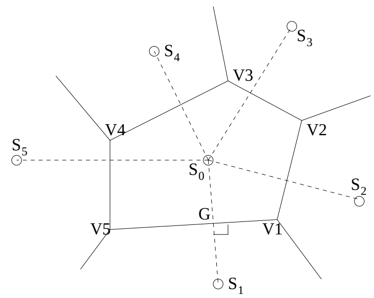
The Voronoi diagram [4], [11] is an important data structure in computational geometry. It represents the proximity information about a set of geometric nodes. The Voronoi diagram of a collection of nodes partitions the space into polygons. Every point in a given polygon is closer to the node in this polygon than to any other node. Figure 1(a) is an example of the Voronoi diagram, and Figure 1(b) is an example of a Voronoi polygon. We define the Voronoi polygon of s_0 as $G_0 = \langle \mathcal{V}_0, \mathcal{E}_0 \rangle$, where $\mathcal{V}_0$ is the set of Voronoi vertices of s_0, and $\mathcal{E}_0$ is the set of Voronoi edges. As shown in Figure 1(b), $\mathcal{V}_0 = \{V_1, V_2, V_3, V_4, V_5\}$, and $\mathcal{E}_0 = \{V_1V_2, V_2V_3, V_3V_4, V_4V_5, V_5V_1\}$. We use $\mathcal{N}_0$ to denote the set of Voronoi neighbors of s_0. In Figure 1(b), $\mathcal{N}_0 = \{s_1, s_2, s_3, s_4, s_5\}$. The Voronoi edges of s_0 are the vertical bisectors of the line passing s_0 and its Voronoi neighbors, e.g., V_1V_5 is s_0s_1's bisector.

Our sensor deployment protocols are based on Voronoi diagrams. As shown in Figure 1, each sensor, represented by a number, is enclosed by a Voronoi polygon. These polygons together cover the target field. The points inside one polygon are closer to the sensor inside this polygon than the sensors positioned elsewhere. If this sensor cannot detect the expected phenomenon in its Voronoi polygon, no other sensor can detect it. Therefore, to examine coverage holes, each sensor only needs to check its own Voronoi polygon. If its sensing area cannot cover the polygon, there are some coverage holes.

To construct the Voronoi polygon, sensors first calculate the bisectors of their neighbors and themselves. These bisectors (and possibly the boundary of the target field) form several polygons. The smallest polygon encircling the sensor is the Voronoi polygon of this sensor.



(a) Voronoi diagram



(b) Voronoi polygon G_0 of s_0

Fig. 1. Voronoi diagram

III. Theoretical Analysis

When a portion of deployed sensors are mobile, the deployment problem can be described as follows: given a target field covered by a number of circles (the sensing circles of the static sensors), but still having some uncovered areas, how to place a certain number of additional circles (the sensing circle of the mobile sensors) to maximize the overall coverage.

This problem is a NP-hard problem, which can reduce to the vertex covering problem [29]. The detailed proof of NP-Completeness is shown in the APPENDIX.

Although our problem is a fundamentally difficult problem, and there is no optimal solution, we can still find some practical solutions to approximate the optimal solution based on heuristics. Similar to the greedy algorithm, which is a commonly used heuristic for the vertex covering problem, we can place mobile sensors to the largest coverage holes.

IV. Basic Bidding Protocol

In this section, we present the basic bidding protocol. We evaluate its performance in terms of coverage, energy consumption, and deployment time in Section VI. The description of this protocol provides a basic understanding and insight into our solution. We improve upon this protocol with optimizations described in

Section V to reduce the required moving distance for each mobile sensor.

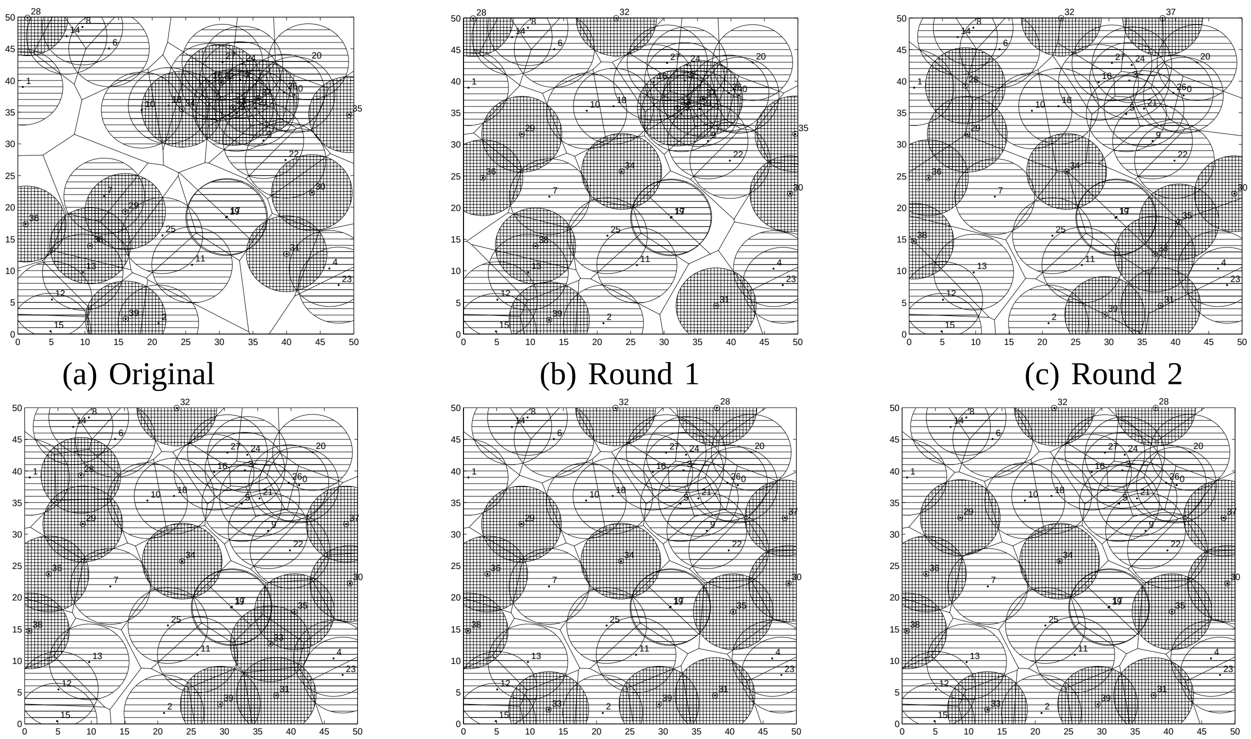
A. Bidding Protocol Overview

According to the greedy heuristic for this NP-hard problem, mobile sensors should move to the area where the most additional coverage can be obtained. After a mobile sensor leaves its original location to cover (heal) another coverage hole, it may generate a new hole in its original location. Thus, a mobile sensor only moves to heal another hole if its leaving will not generate a larger hole than that to be healed. However, due to lack of global information, mobile sensors may not know where a coverage hole exists. Even with the location of the coverage hole, it is still a big challenge to find the target position inside the coverage hole which can bring the most additional coverage when a mobile sensor is placed there compared to other positions. We propose to let the static sensors detect the coverage holes locally, estimate the size of these holes, and determine the target position inside the hole. Based on the properties of the Voronoi diagram, static sensors can find the coverage holes locally and provide a good way to estimate the target location of the mobile sensors.

The roles of mobile and static sensors motivate us to design a bidding protocol to assist the movement of the mobile sensors. We view a mobile sensor as a hole healing server. Its service has a certain base price, which is the estimate of any generated coverage hole after it leaves the current place. Static sensors are the bidders of the coverage hole healing services. Their bids are the estimated sizes of the holes they detect. Static sensors bid mobile sensors that have a base price lower than their bid. Mobile sensors choose the highest bid and move to the target locations provided by the static sensors.

The bidding protocol runs round by round after the initialization period. During the initialization period, all static sensors broadcast their locations and identities locally. We choose the broadcast radius to be two hops, with which sensors can construct the Voronoi diagram in most cases. After the initialization period, static sensors broadcast this information again only when new mobile sensors arrive and need this information to construct their own Voronoi cells.

Each round consists of three phases: *service advertisement*, *bidding*, and *serving*. In the advertisement phase, mobile sensors broadcast their base prices and locations in a local area. The base price is set to be zero initially. By the end of the service advertisement phase, each static sensor has a *service list*, which is a list of mobile sensor IDs along with their location and base price. In the bidding phase, static sensors detect coverage holes locally by examining their Voronoi cells. If such holes exist, they calculate the bids and the target locations for the mobile sensors. Examining the service list, the static sensor chooses a mobile sensor whose base price is lower than its bid, and sends a bidding message to this mobile sensor. We will present how to determine the mobile sensor to bid if there are multiple mobile sensors whose base price is lower than the bid of the static sensor. In the serving phase, the mobile sensor chooses the highest bid and moves to heal that coverage hole. The accepted bid will become the new base price of the mobile sensor. After the serving phase, the mobile sensors broadcast their new locations and new base prices and a new round begins. Because the base price increases monotonically, when no static sensors can give out a bid higher than the base price of the mobile sensors, the protocol terminates.



(a) Original (b) Round 1 (c) Round 2

(d) Round 3 (e) Round 4 (f) Round 5

Fig. 2. Snapshot of the execution of the bidding protocol

Before getting into the technical details of the bidding protocol, we first use an example to show how the protocol works. As shown in Figure 2, the circles with a striped shadow represent the sensing coverage of the static sensors, and the circles with grid shadow are that of the mobile sensors. Initially, 40 sensors are randomly placed in a $50m \times 50m$ flat field, among which 30% are mobile sensors. The initial coverage is 82%. The protocol terminates in the fifth round when the coverage reaches 93%. The sixth round has the same topology as the fifth round.

B. Distributed Calculation of the Voronoi Cell

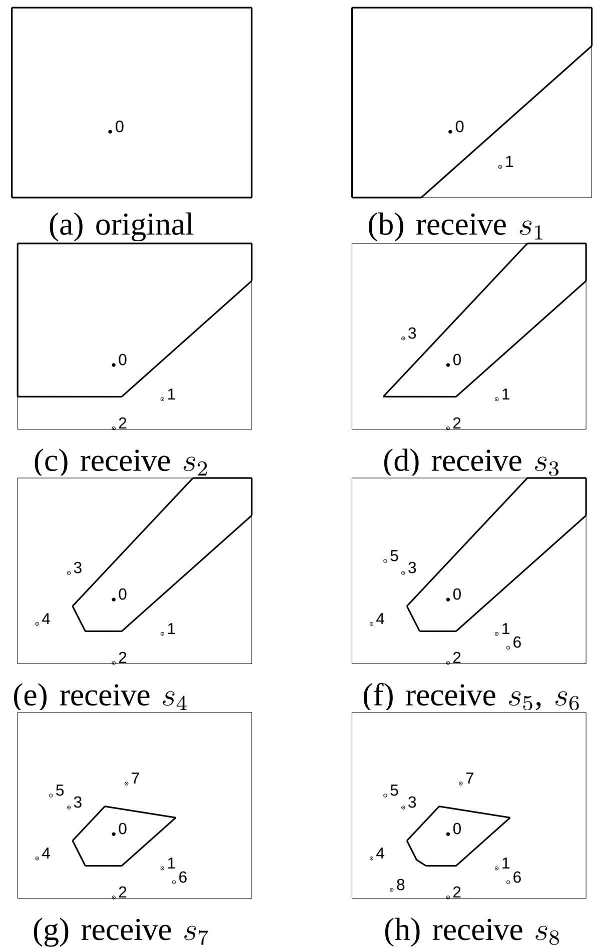


Fig. 3. Computing the Voronoi cell

It is difficult to compute Voronoi diagrams [4]. However, to detect and calculate the coverage hole, each sensor only needs to know its own Voronoi cell, whose calculation can be simplified as follows. We take sensor s_0 as an example. Initially, as shown in Figure 3 (a), the Voronoi cell of s_0 is set to be a large rectangle. After receiving the *hello* message from sensor s_1, s_0 knows the location of s_1 and computes the bisector line of s_1 and itself. This line is added to the original graph and two polygons are generated. Shown in Figure 3 (b), the polygon including s_0 becomes the new view of s_0's Voronoi cell. Later, after s_0 receives the *hello* messages from s_2, s_3, and s_4, its Voronoi cell changes from Figure 3(c) to Figure 3(e) accordingly. The Voronoi cell will not change if the computed bisector line has no intersection with it. As shown in Figure 3(f), knowing s_5 and s_6 does not affect s_0's Voronoi cell. Finally, the true Voronoi cell is generated after s_0 knows the existence of s_7 and s_8.

Static sensors construct Voronoi cells considering only static neighbors and mobile neighbors which are not likely to move. These mobile sensors are detected by examining their base prices. If the base price of a mobile sensor is zero, this mobile sensor has not moved yet and most likely it will move to heal some coverage hole. Thus, when detecting coverage holes, static sensors do not consider mobile sensors which are likely to leave. To find out if a coverage hole exists, a static sensor checks whether its distance to the farthest Voronoi vertex is longer than the sensing range. If yes, then some coverage hole exists and this sensor should prepare to bid some mobile sensor to heal it.

Voronoi cells calculated in this way will not be accurate when Voronoi neighbors are far away from each other and cannot communicate with each other. The accurate calculation of Voronoi cells is not required in these cases because the coverage holes will be large. The algorithm will not mis-detect coverage holes.

C. Bid Estimation

In the bidding message, static sensors provide the estimated coverage hole size as the bid and the target location to which the mobile sensor should move. This information is calculated based on their Voronoi cells. If there exists a coverage hole, the static sensor chooses the farthest Voronoi vertices as the target location of the coming mobile sensor. Inside one coverage hole, there are many positions that a mobile sensor can be located. If the mobile sensor is placed at the position farthest from any nearby sensors, the gained coverage is the highest since the overlap of the sensing circles between this new coming mobile sensor and existing sensors is the lowest. As shown in Figure 4, sensor s_a chooses its farthest Voronoi vertex O as the target location of the mobile sensor for which it bids.

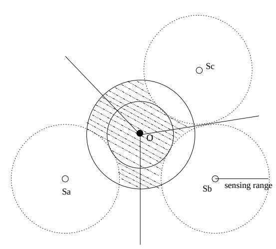


Fig. 4. Bid estimation

From the global point of view, using the greedy heuristic to choose the largest coverage hole may not be optimal in some cases. As shown in Figure 5, A is the farthest Voronoi vertex of s_a. Although a high additional coverage can be obtained by placing a mobile sensor at A, it is not globally optimal since it leaves some scattered coverage holes which are hard to cover by placing additional mobile sensors. To deal with this problem, we propose an optimization which puts a limit on the maximum distance between the calculated target location and the bidder. As shown in Figure 5, by setting this maximum distance, a mobile sensor will be placed at B so that another mobile sensor can move to point C, to achieve better coverage. This maximum distance, denoted by d_{limit}, is a function of sensing range. We choose d_{limit} to be $\sqrt{3} * sensing\_range$, since to place sensing circles in a hexagonal relative position will minimize overlapping and maximize the coverage. Under these conditions, the distance between the centers of the sensing circles is $\sqrt{3} * sensing\_range$.

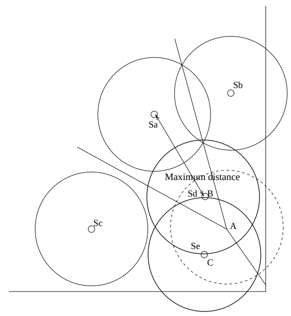


Fig. 5. Optimize the greedy heuristic

Having determined the target location of the mobile sensor it bids, static sensors calculate the bid as: $\pi * (d - sensing\_range)^2$, where d is the distance between the bidder and the target location. As shown in Figure 4, s_a's bid is the area of the inner circle centered at O, which is not the actual additional coverage to be obtained. The actual additional coverage is the shadow area, which is difficult to calculate since it involves the union of circles. Using the inner circle as the bid simplifies the calculation, and can be used to approximate the actual additional coverage, which is the sensing circle minus the overlapping area of the sensing circles. The larger the overlapping area, the smaller the inner circle. Thus, the bid used can represent the relative size of the coverage holes.

Note that the maximum base price (or bid) is $\pi * (d_{limit} - sensing\_range)^2$, which is $\sqrt{3} * \pi * sensing\_range^2/2$.

The property of the Voronoi diagram guarantees that the shadow area is always the additional coverage. This can be explained as follows. The points inside one Voronoi cell are closest to the sensor in this cell. The points in the Voronoi edge are closest to these two sensors besides this edge. The Voronoi vertex is the point closest to the sensors which contribute to the existence of this vertex. The sensing circle centered at the Voronoi vertex must only overlap with the sensing circles of the sensors which contribute to the construction of this Voronoi vertex. Thus we guarantee that the shadow area shown in Figure 4 is always the additional coverage brought by placing a mobile sensor at O.

In addition to the static sensors, mobile sensors with a base price larger than zero also act as bidders. This is necessary because mobile sensors with a relatively larger price are essentially acting as static sensors. At this point, they can assist the movement of other mobile sensors.

D. Criteria of Choosing Mobile Sensors to Bid

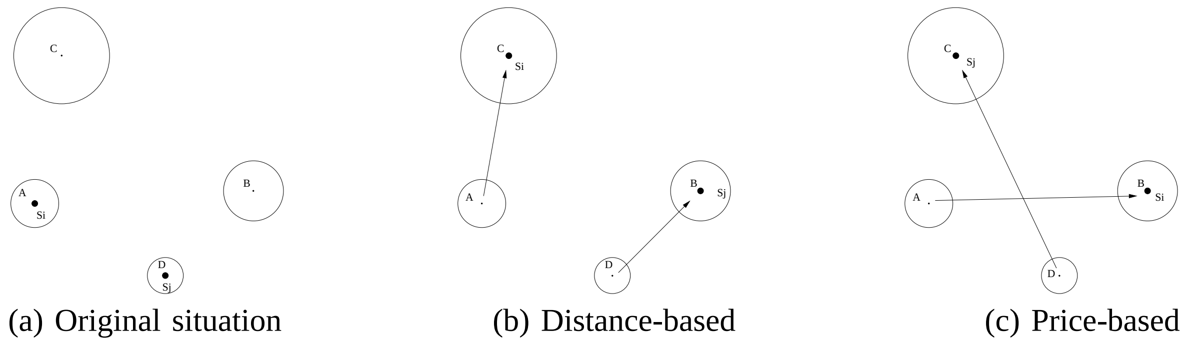


Fig. 6. Distance-based vs. price-based

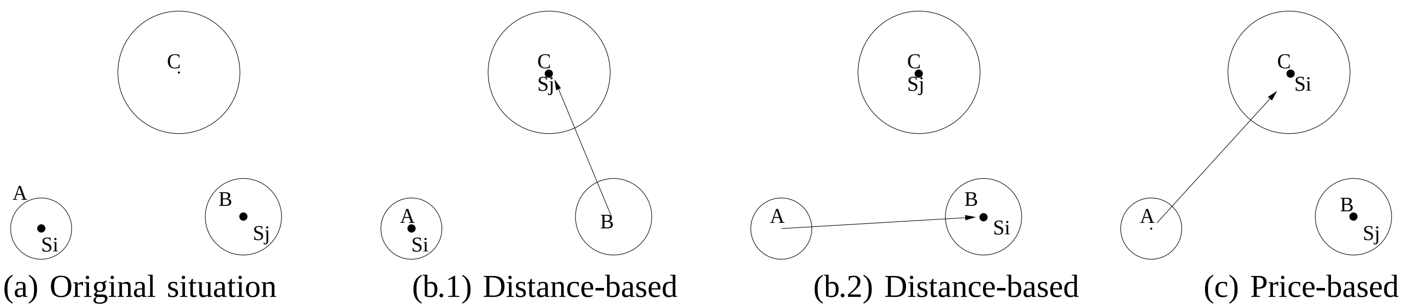


Fig. 7. Distance-based vs. price-based

After the service advertisement phase, each sensor has a list of mobile sensors, their locations and their base prices. A bidder needs to determine which mobile sensor to bid among those having a lower base price than its bid. We propose two criteria for choosing mobile sensors: *distance-based and price-based*. In the distance-based approach, a bidder chooses the closest mobile sensor to bid; in the price-based approach, a bidder chooses the cheapest mobile sensor to bid. The advantage of the distance-based approach is shown in Figure 6. We use a dashed circle to represent the coverage hole. The center of the circle is the target position of the mobile sensor to heal this hole. Initially, s_i is located in hole A and s_j is in hole D. The base price of s_i is higher than s_j, since the size of A is larger than D. In the distance-based approach, hole C will bid s_i and hole B will bid s_j. The movement of s_i and s_j is shown in Figure 6(b). In the price-based approach, both holes C and B will bid s_j and hole C wins. Then hole B bids s_j. Their movement is shown in Figure 6(c). As can be seen, the average moving distance of s_i and s_j is shorter in the distance-based approach, because the distance-based approach helps sensors move to their closest holes.

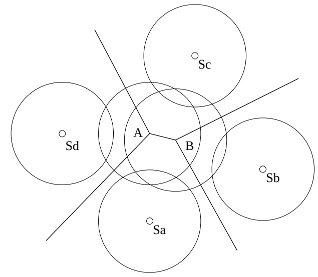
The advantage of the price-based approach is shown in Figure 7. With the price-based approach, hole C bids s_i since it has a lower base price. s_i moves once and no other sensor needs to move. But with the distance-based approach, hole C bids s_j since it is closer. After sensor s_j moves to hole C, hole B needs a sensor and it will bid s_i. In this way, both s_i and s_j have to move.

E. Multiple Healing Detection

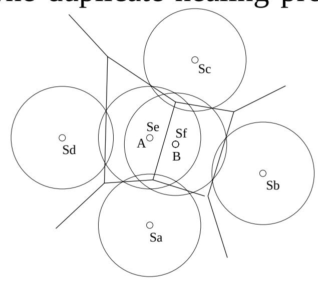
Due to the limited service advertisement radius, static sensors may have different knowledge about the mobile sensors. Therefore, it is possible that several static sensors independently bid different mobile sensors for the same coverage hole since the cheapest mobile sensor or the closest mobile sensor in their views is different. If more than one succeeds in bidding, multiple mobile sensors will move to heal the same hole, which is not necessary. Figure 8(a) shows one example. A is the farthest Voronoi vertex of s_a and B is the farthest Voronoi vertex of s_b. Both s_a and s_b bid mobile sensors to their farthest Voronoi vertices. When both biddings are accepted, a multiple healing occurs.

We propose a self-detection algorithm for mobile sensors to solve this problem. A mobile sensor has a knowledge of the locations and base prices of other mobile sensors in its neighborhood after the service advertisement phase. If it finds out that some other mobile sensors have a higher base price than its own, it will run the detection algorithm to check whether a multiple healing has occurred. If yes, the mobile sensor will lower its base price to zero and most likely some sensor will bid it to cover a different hole.

In the detection algorithm, the detecting mobile sensor calculates a **detecting threshold**, equal to $\pi * (d_{min} - sensing\_range)^2$, where d_{min} is the distance to its closest neighbor. If the detecting threshold is smaller than its new base price, or d_{min} is smaller than the sensing range, a multiple healing has occurred, since without multiple healing, the calculated value should be the same as its new base price. As shown in Figure 8(b), s_e and s_f, located in A and B respectively, are the mobile sensors bid by s_a and s_b. s_f's new base price, the bid put forward by s_b, is calculated without considering s_e, which is $\pi * (d_{b,f} - sensing\_range)^2$, where $d_{b,f}$ is the distance between s_b and s_f. Without a multiple healing, $d_{b,f}$ is just d_{min}, and the calculated detecting threshold should be the same as the new base price. If multiple healing has occurred, $d_{e,f}$ is d_{min}, which is smaller than $d_{b,f}$, and the detecting threshold is smaller than the new base price.



(a) The duplicate healing problem



(b) Fix the duplicate healing problem

Fig. 8. Duplicate healing

V. Proxy-based Bidding Protocol

In this section, we present the proxy-based bidding protocol. This protocol improves the performance of the basic bidding protocol in terms of energy efficiency and load balance. In this protocol, sensors only move after their final location is determined; all calculations with respect to multiple healing detection and optimization are carried out through the exchange of messages before movement. The key tradeoff is the increased number of messages $vs.$ the decreases in required movement. Because movement is typically much more expensive than exchanging messages, this protocol provides a more efficient solution than the basic bidding protocol.

A. General Idea: Logical Movement

Although the basic bidding protocol can achieve a high coverage, there is still room for improvement in terms of energy efficiency and load balance. In the basic bidding protocol, mobile sensors move iteratively to heal large and larger holes. Most likely, mobile sensors will move in an irregular pattern, which consumes more energy than moving directly from their initial location to the final destination. Also, in the basic bidding protocol, some sensors are penalized by being required to move a long distance. These phenomena are illustrated by the following example, shown in Figure 9(a). In the first round, holes A, B, and C bid mobile sensor s_b, and hole D bids s_a. Hole A and hole D win due to their large size, and these two sensors move. In the second round, hole C bids for sensor s_a; hole B does not bid in this round since it does not know the existence of s_a due to the limited advertisement radius. In round three, hole B knows of s_a and bids it. s_a moves the third time to reach its final location, resulting in a much longer moving distance than s_b. Ideally, s_a shall move to heal hole A and s_b move to heal hole B, as shown in Figure 9(b). The comparison between the basic bidding protocol and the ideal solution motivates us to propose the proxy-based bidding protocol to better allocate mobile sensors to coverage holes such that the overall moving distance is shortened and no sensor is penalized.

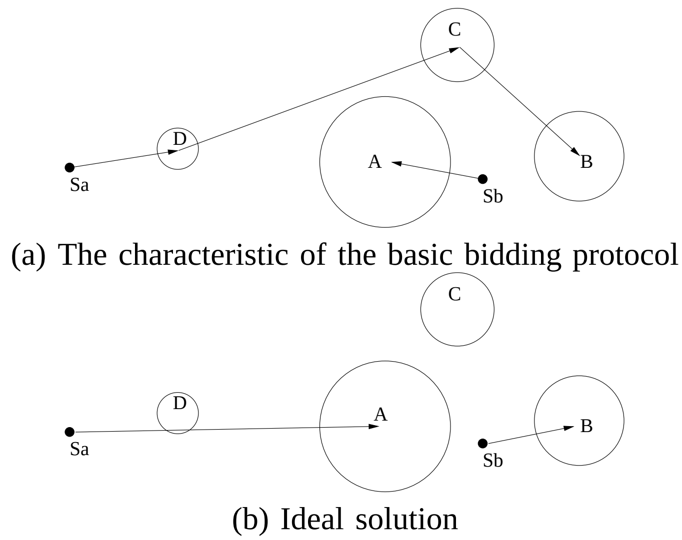


Fig. 9. Motivation of proxy-based bidding protocol

Following the same bidding framework, the proxy-based bidding protocol deploys the idea of *virtual movement*. Instead of moving physically in each round, mobile sensors perform virtual movements once they accept a bid. They only perform physical movements after they determine their final destinations. In this way, mobile sensors will not move in an irregular pattern. Also, virtual movement enables the possibility for mobile sensors to exchange their coverage holes to further shorten the moving distance since it does not matter which sensor heals which hole when all the largest holes are to be healed. For example, as shown in Figure 9, through logical movement, s_a identifies that hole B as its final destination and s_b identifies hole A. Before they perform the physical movements, they can exchange their destinations, i.e., s_a moves to hole B and s_b moves to hole A, such that an ideal allocation of mobile sensors to holes is obtained. In addition, with virtual movement, we can do multiple-healing detection before sensors physically move, and the vain movements of the sensors involved can be saved. In the following sections, we present the details of this protocol.

B. Proxy Sensor

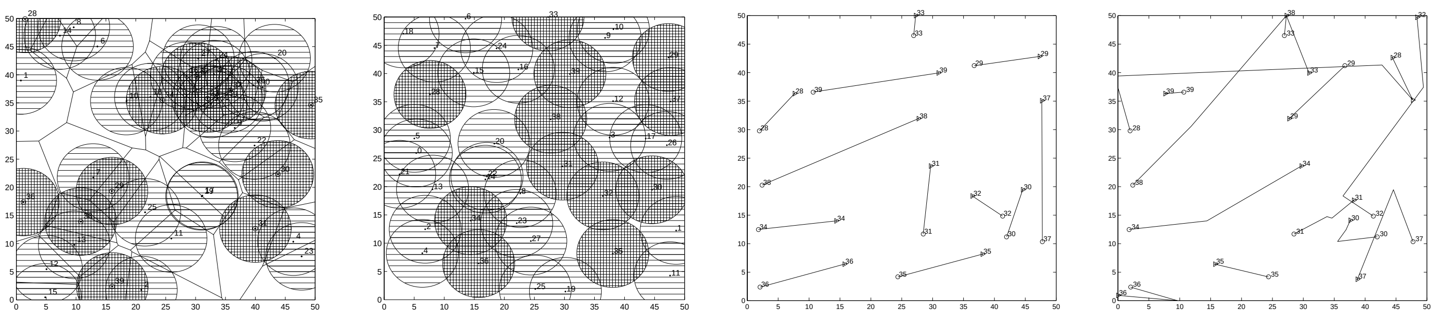
To implement virtual movement, the first problem to be addressed is how to advertise services to the neighborhood of those virtual positions when mobile sensors do not move. One intuitive solution is to perform a network-wide broadcast. However, this may significantly increase the communication overhead. To keep the same communication overhead and let the sensors in the neighborhood of the virtual position of the mobile sensor receive the advertisement messages, we propose to use *proxy sensors*, which are static sensors located closest to the virtual positions of the mobile sensors, to advertise the services and process the bidding messages for those mobile sensors.

The sensor closest to a mobile sensor's virtual position should be the bidder who detects the coverage hole and bids this mobile sensor since sensors detect coverage holes locally by checking their Voronoi polygons. Therefore, we choose the winning bidder as the proxy of the mobile sensor who accepts its bid.

The proxy of a mobile sensor is not fixed during the lifetime of the mobile sensor. When a mobile sensor accepts a bid in the first round, it sends a *delegate* message to the bidder and the bidder becomes the first proxy of this mobile sensor. In the next round, the bidder (proxy) advertises the virtual position and the new base price of the mobile sensor. In the view of other sensors, the mobile sensor has moved to its virtual position and the Voronoi diagram is computed based on the new virtual position of the mobile sensor. Based on the new base price of the mobile sensor, static sensors can still bid for the mobile sensor. Their bidding messages, if any, will be sent to the proxy instead of the real mobile sensor. Based on the received bidding messages, the proxy determines which new hole should be healed. If a new bid is accepted, the proxy delegates the proxy role to that bidder who will become the new proxy of the mobile sensor. In this way, the physical movement of the mobile sensor is replaced by delegating the role of proxies between static sensors, thus realizing virtual movement. When a proxy sensor does not receive any bidding messages for a *waiting threshold* of n rounds, it will notify the mobile sensor to perform physical movement. Experimentally we determined that $n = 2$ provides good results.

In addition to virtual movement, by using a proxy sensor, *multiple healing* can be detected before it happens in many situations. After the service advertisement, proxy sensors have a service list which contains the information of the virtual positions of mobile sensors. A proxy sensor can act as the mobile sensor it represents and detect whether a *multiple healing* would happen by examining its service list, with the same method of *multiple healing* detection presented in Section IV-E. A proxy sensor calculates the Voronoi cell without considering its mobile sensor, as if its bid in the previous round had failed. Then it checks whether the original coverage hole remains; if the same hole exists, no multiple healing has occurred since its mobile sensor is required to heal the hole; otherwise, some neighbor has bid a mobile sensor to heal the same hole and a multiple healing has occurred. If the proxy discovers that a multiple healing has occurred, it reduces the base price of its delegated mobile sensor to zero, and re-advertises the new service in the subsequent rounds.

To avoid all proxies from detecting the same multiple-healing and reducing the base prices of their delegated mobile sensors to zero, the proxies check whether the moving distance of its delegate from its current position to this hole is the shortest among those mobile sensors that heal the same hole. If not, it reduces the base price of its delegate to zero; otherwise, it waits for other mobile sensors to leave.



(a) Initial distribution (b) After deployment (c) Trace of the proxy-based protocol (d) Trace of the basic bidding protocol

Fig. 10. An operational example

C. Coverage Hole Exchange

Coverage hole exchange is proposed to reduce the overall moving distance and to reduce the chance that an individual sensor is penalized by moving a long distance. It is performed by proxy sensors. A proxy sensor checks the service list obtained after the service advertisement phase and determines with which mobile sensor to exchange the virtual position of its mobile sensor. The exchange criteria will be described in depth in the next paragraph. If an exchange is necessary, the proxy sensor sends a request to the proxy of the mobile sensor with which it wants to exchange position. A proxy sensor which receives multiple exchange requests chooses one by the same criteria and sends back a confirm message. Then these two proxy sensors exchange delegation of their mobile sensors, and the two mobile sensors exchange the proxies and their associated coverage holes.

Before presenting the exchange criteria, we introduce the following notations. We use d_i to represent the moving distance of s_i before exchange, and $\hat{d}_i$ the moving distance after exchange. d_{max} is a maximum moving distance threshold. All exchanges between s_i and s_j must satisfy the following prerequisites; otherwise, the exchange will not be performed.

$$\begin{cases} \hat{d}_i + \hat{d}_j \leq d_i + d_j \\ \hat{d}_i \leq max[d_i, d_{max}] \\ \hat{d}_j \leq max[d_j, d_{max}] \end{cases} \tag{1}$$

Shown in (1), all exchanges must reduce the overall distances. Also, the exchanges must not increase the moving distance of a single sensor to be longer than d_{max} if it is not so before the exchange, and must not further increase the moving distance of a single sensor if its moving distance is already longer than d_{max}.

Among the exchanges which satisfy the prerequisites shown in (1), we give higher priority to those which can release or mitigate node penalization. We first check exchanges, in which one or both involved sensors have to move longer than d_{max} before the exchange, and choose the one which can reduce the overall moving distance the most.

Formally, the exchange is chosen as follows:

$$[s_i, s_j] = argmin_{\hat{d}_k + \hat{d}_l - d_k - d_l}\{[s_k, s_l] : d_k \geq d_{max} \vee d_l \geq d_{max}\} \tag{2}$$

Here, $[s_i, s_j]$ indicates mobile sensor s_i exchanges its virtual position with s_j. If there is no such exchange, we choose the exchange which can reduce the overall moving distance the most. That is:

$$[s_i, s_j] = argmin_{\hat{d}_k + \hat{d}_l - d_k - d_l}\{[s_k, s_l] : d_k \leq d_{max} \wedge d_l \leq d_{max}\} \tag{3}$$

Without hole exchange, proxy sensors can notify mobile sensors to move if they do not receive bidding messages for the *waiting threshold* of n rounds. Hole exchange complicates the decision of when to tell a sensor to move. As shown in Figure 9, if s_b moves physically in the third round, s_a has no sensor with which to exchange its virtual position after it virtually moves to hole B. To solve this problem, s_b should wait for more rounds before movement. In general, a mobile sensor that gets a high base price in the first two rounds should wait for additional rounds before physically moving so that other sensors have an opportunity to perform hole exchange. Through extensive experiments, we determined that $n = 5$ for sensors that receive high bid prices in the first two rounds, and $n = 2$ for other sensors, yields good results.

There is an exception to this general principle. For very large holes, i.e., holes bigger than the sensing range of a single sensor, as shown in Figure 5, two mobile sensors (or more) are needed for healing. In Figure 5, static sensor s_a bids mobile sensor s_d to move, and it is s_b which bids another mobile sensor s_e to heal the same hole. Normally, sensors that move first to heal the hole act as bidders in the next rounds to bid more sensors to heal the same hole. These sensors, like s_d, which move first and have the maximum base price $\sqrt(3) * pi * sensing\_range^2/2$ (described in section IV-C), should move immediately because they will act as bidders.

D. Protocol Specification

As with the basic bidding protocol, the proxy-based bidding protocol runs round by round until mobile sensors obtain their final locations and move there directly. Each round consists of four phases: service advertisement, bidding, virtual movement, and hole-exchange. (1) In the service advertisement phase, proxy sensors advertise the virtual locations, physical locations and base prices for their delegated mobile sensors. In the first round, a mobile sensor does not have a proxy and advertises its physical location and base price by itself. (2) In the bidding phase, static sensors calculate their Voronoi polygons based on the virtual positions of mobile sensors. They detect coverage holes by examining the Voronoi polygons, estimate the hole size, choose the closest or cheapest mobile sensor, and send bidding messages to its proxy or the mobile sensor itself if the mobile sensor has no proxy. (3) In the virtual movement phase, proxy sensors (or mobile sensors without a proxy) choose the highest bid and send a delegate message to the bidder. The bidder becomes the new proxy. The base price of mobile sensors is updated by their new proxies. Also, proxy sensors need to check whether hole-exchange is needed. If yes, they choose the mobile sensor suitable for exchange and send out an exchange request to the proxy of that mobile sensor. (4) In the hole-exchange phase, proxy sensors check the received requests, choose one with the highest priority and return the confirm message to the requester. Then the mobile sensors delegated by these two proxy sensors exchange the hole to heal.

The protocol terminates naturally when all the largest holes are healed and no more hole exchanges are necessary. Through the bidding process, when no sensors can raise a bid higher than the lowest base price of mobile sensors, all the largest holes are healed. This process terminates naturally as presented in Section IV. For hole exchange, we require that all the exchanges must reduce the overall moving distance. There is a lower bound of the overall moving distance, and hole exchange will finish naturally. Through this iterative 4-phase process, proxy sensors notify mobile sensors to move and the deployment process terminates. We show the formal algorithm in Appendix A.

We show an operational example to illustrate the advantage of the proxy-based protocol over the basic bidding protocol. 40 sensors, of which 30% are mobile, are randomly distributed in a 50m*50m field. The initial distribution is shown in Figure 10(a); the distribution after deployment is shown in Figure 10(b). In this example (and most others), the proxy-based bidding protocol and the basic bidding protocol get the same distribution of sensors after deployment. Figure 10(c) shows the moving trace of mobile sensors in the proxy-based bidding protocol. The mobile sensors move $13.65m$ on average. Sensor 38 moves the longest distance $27.85m$. Figure 10(d) shows the moving trace of mobile sensors in the basic bidding protocol. The average moving distance is $23.77m$. Sensor 28 has the longest moving distance. It moves 5 times for a total distance of $68.68m$. From this example, we can see that the proxy-based protocol is more energy-efficient and load-balanced.

VI. Performance Evaluations

A. Objectives, Metrics, and Methodology

We implement our deployment protocols in the ns-2 (version 2.1b9a). Our objectives in conducting this evaluation study are three-fold: first, justifying our proposal of constructing sensor networks with both mobile and static sensors to balance cost and sensing coverage; second, testing the effectiveness of our bidding protocols in providing high coverage; finally, comparing the basic bidding protocol and the proxy-based bidding protocol, and giving some insight on choosing deployment protocols.

We analyze the performance of our schemes from three aspects: *sensor cost*, *deployment quality*, and *energy consumption*. Sensor cost is measured by the money used to construct the network. Deployment quality is measured by the sensor coverage and the time (number of rounds) to reach this coverage. Deployment time is determined by the number of rounds needed and the duration of each round. The duration of each round is primarily determined by the moving speed of sensors, which is the mechanical attribute of sensors. Thus, we only use the number of rounds to measure the deployment time. Energy consumption includes two parts, mechanical movement and communication. Message complexity is used to measure the energy consumed in communication. As for movement, the energy consumed in moving a sensor n meters consists of two parts: starting/braking energy and moving energy. Therefore, we use moving distance and the number of movements as the metrics.

We run simulations for different compositions of sensor networks, and determine the coverage that can be reached. In a $60m * 60m$ flat field, we randomly distribute 60 sensors. Among these sensors, we assign a percentage of sensors to be mobile. This percentage varies from 10% to 50%, with an increment of 10%. The mobile sensors are chosen randomly. To evaluate each metric under different parameter settings, we run 50 experiments based on different initial distributions and calculate the average results.

We choose 802.11 as the MAC layer protocol and DSDV as the routing protocol. The physical layer is modeled after the RF MOTE from Berkeley, with 916.5MHZ OOK 5kbps as the bandwidth and 20 meters as the transmission range. Based on the information from [1], we set the *sensing range* to be 6 meters. This is consistent with other current sensor prototypes, such as Smart Dust (U.C.Berkeley), CTOS dust, Wins (Rockwell)[2].

In the following sections, we show the simulation results.

B. Tradeoff between cost and coverage

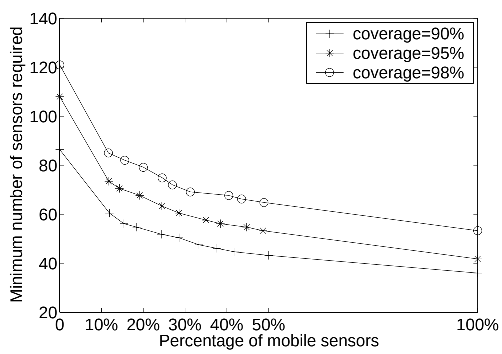
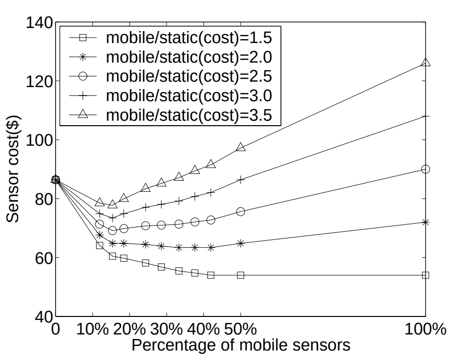
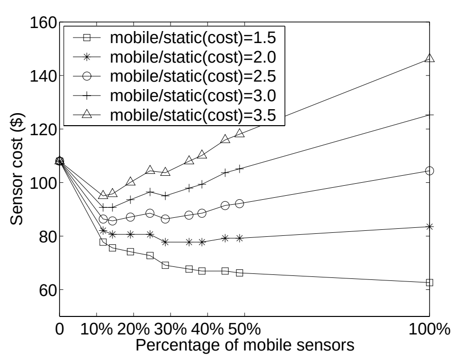


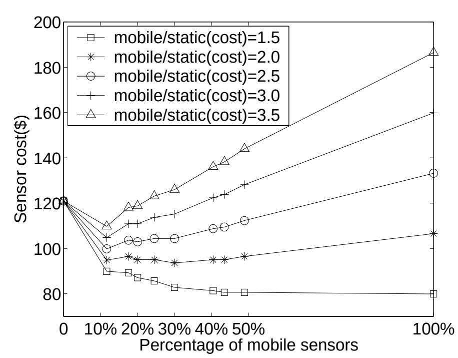
Fig. 11. The number of sensors needed to reach certain coverage under different mobile percentage



(a) To reach 90% coverage



(b) To reach 95% coverage



(c) To reach 98% coverage

Fig. 12. The cost of sensors to reach certain coverage

In order to evaluate the tradeoff between sensor cost and coverage, we consider three cases of network composition: all the sensors are mobile; all the sensors are static; a percentage of sensors are mobile. When all the sensors are static, random deployment is used. When all the sensors are mobile, the VOR protocol [14], [16] is used for sensor deployment. When a percentage of sensors are mobile, our basic bidding protocol (using the distance-based criteria) is used. Figure 11 shows the total number of sensors needed to reach certain coverage with under different network compositions. 0% of mobile sensors means that all sensors are static.

As shown in Figure 11, to reach a certain coverage, random deployment of static sensors uses the most number of sensors; as the percentage of mobile sensors increases, the required number of sensors to reach a certain coverage decreases; a deployment of 100% mobile sensors requires the fewest sensors. However, the cost of mobile sensors may be high.

Compared to random deployment, the basic bidding protocol can significantly reduce the number of sensors required to reach a certain coverage. For example, to reach a 90% coverage, with only 10% of mobile sensors, the basic protocol needs 30% fewer sensors; when 50% of the sensors are mobile, the required number of sensors is reduced by 50%.

Compared to the case in which 100% of the sensors are mobile, to reach 90% coverage, the basic bidding protocol requires 40% fewer mobile sensors in the case in which 50% sensors are mobile. Note that the cost of mobile sensors are higher than static sensors, so the overall cost of using a percentage of mobile sensors may be reduced even though more sensors in total are used.

Figure 12 shows the sensor cost of these three protocols to reach a certain sensor coverage. Based on the *cost ratio* between the mobile sensor and the static sensor, the overall sensor cost of these three protocols may be different. Intuitively, if the cost ratio is low (e.g., 1.5), increasing the percentage of mobile sensors can reduce the overall sensor cost. On the other hand, if the mobile sensors are very expensive, using only static sensors may have the lowest sensor cost (not shown in the figure). When the cost ratio is somewhere in the middle, the basic protocol which has a mix of mobile and static sensors can achieve the lowest sensor cost. For example, when the cost ratio is 3.5, to reach 95% coverage, the basic protocol has the lowest cost when 10% sensors are mobile. Currently, the cost of a static sensor prototype Motes is about $\$100$ and the cost of a mobile sensor prototype is about $\$200$ [8]. The ratio is expected to increase under mass production. Based on this figure, we can see there is a tradeoff between cost and coverage. The basic protocol can achieve a balance between these two most of the time.

C. Comparing the Protocols

We consider four cases: both the proxy and basic bidding protocols using both distance and price-based criteria. In the figures showing the simulation results, we use ’Proxy-distance’, ”Proxy-price’, ’Basic-distance’ and ’Basic-price’ to represent them, respectively.

1) Coverage: Figure 13 shows the coverage obtained by our protocols under different mobile sensor percentage. We can make two observations from the figure. One is that our bidding protocols can increase the coverage significantly. The other is that all four cases we consider achieve very similar coverage. All the four cases follow the same bidding framework and heal the largest holes. In terms of coverage, there is no preference between the basic-bidding protocol and the proxy-based bidding protocol; there is no preference between distance-based criteria and price-based criteria to choose mobile sensors.

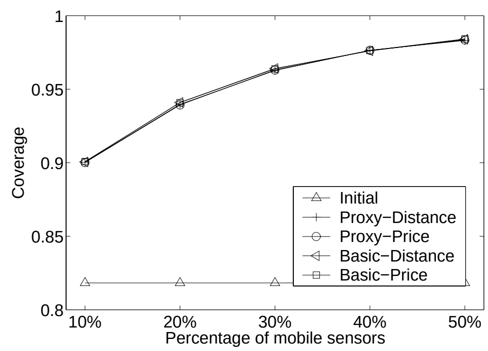


Fig. 13. Coverage

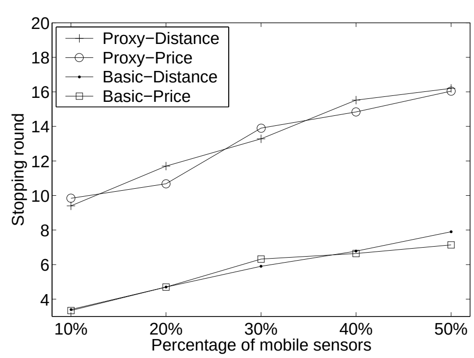


Fig. 14. Termination

2) Termination: When all the largest holes are healed and no sensor can give a higher bid than the lowest base price of mobile sensors, the protocols terminate. Figure 14 shows the number of rounds that the protocols have run when the protocols terminate. As expected, the proxy-based bidding protocol requires more rounds to terminate. In the proxy-based bidding protocol, each sensor waits several rounds before physical movement and sensors spend a number of rounds on exchanging holes. However, because physical movements will likely dominate the recursion time, and the proxy-based protocol reduces movement, it may still terminate in the shortest time. In both the proxy-based protocol and the basic protocol, using distance-based criteria or price-based criteria does not significantly affect the termination time.

The deployment rounds are increased when the mobile sensor percentage increases. With more mobile sensors available, the allocation of mobile sensors to coverage holes is more complicated and needs more rounds.

3) Energy Consumption: Energy consumption includes two parts, mechanical movement and communication. We use message complexity to measure the energy consumed in communication; we use the number of movements and moving distance to measure the energy consumption in movement. We first show the performance of our protocols in these three metrics. Then we show a unified energy consumption considering all these metrics.

Figure 15 shows the moving distance. As expected, the moving distance is much lower when the proxy-based bidding protocol is used. Between the distance-based criteria and price-based criteria, the moving distance is quite similar when using the proxy-based bidding protocol, and it is shorter when the latter criteria is used in the basic bidding protocol. The figure tells us that the phenomena shown in Figure 7 are dominant compared to those shown in Figure 6. In the proxy-based bidding protocol, these two criteria achieve similar performance. For most cases, the hole exchange and virtual movement change the situations illustrated in Figure 7 and Figure 6 to an ideal case. Therefore, these two criteria achieve a similar performance.

When considering the number of movements $vs.$ the percentage of mobile sensors (not shown), we find that the number of movements required does not change as the percentage of mobile sensors increases. In addition, both the distance-based criteria and price-based criteria perform the same when using the proxy-based protocol (about 1.1). When using the basic bidding protocol, the price-based criteria achieves a smaller number of movements (about 1.45) than the distance-based criteria (about 1.6) for the same reason as presented in the above paragraph.

Figure 16 shows the message complexity. The proxy-based protocol has higher message complexity than the basic protocol since it needs more rounds to terminate and needs to negotiate how to exchange holes.

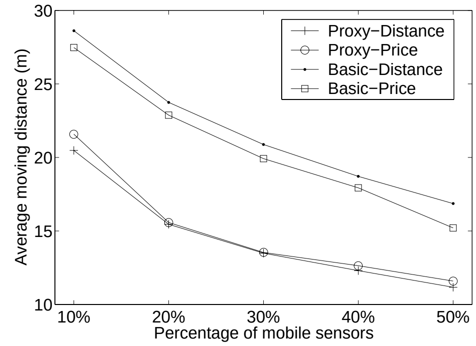


Fig. 15. Moving distance

As mobile sensor percentage increases, the number of rounds increases, and message complexity increases accordingly.

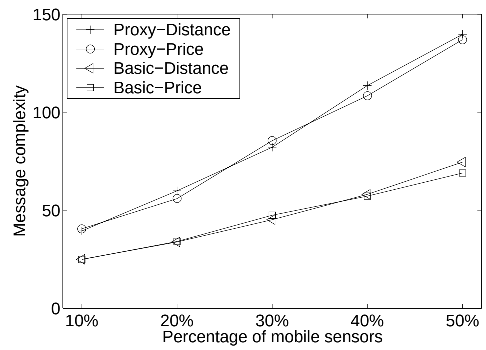


Fig. 16. Message complexity

To get a clear picture of energy consumption, we normalize the moving distance and the number of movements into message complexity. That is, with the same amount of energy consumed in movement, how many messages can be transmitted. Calculated from Robomote [33], approximately, to move a sensor one meter consumes a similar amount of energy as transmitting 300 messages. Also, we set the energy consumption in starting/braking to be the same as that in moving one meter. Figure 17 shows the unified energy consumption. As expected, the proxy-based bidding protocol consumes much less energy than the basic bidding protocol. Though sensors spend more energy in communication, they save much energy in movement. Mechanical movement is the dominant factor in energy consumption. Thus, the proxy-based protocol is much more energy efficient than the basic bidding protocol.

D. Load Balance

The maximum moving distance among the mobile sensors is an indication of whether individual sensors are penalized in terms of moving distance. Our simulations show that the maximum moving distance is about $39m \sim 42m$ in the proxy-based protocol, which is much shorter than the $60 \sim 80m$ in the basic

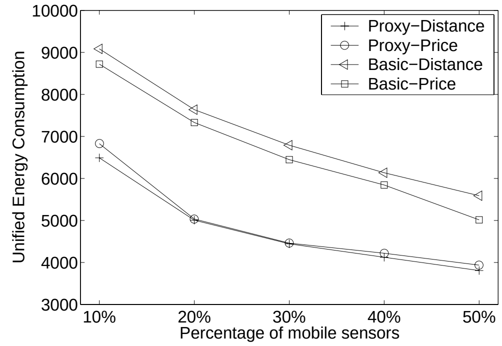


Fig. 17. Unified energy consumption

bidding protocol.

VII. Related Work

In this section, we introduce related work in sensor coverage, static sensor deployment, mobile sensor deployment, and relay node placement in sensor networks.

A. Coverage

Meguerdichian, *et al.*, presented several interpretations of coverage in sensor networks, including deterministic coverage and stochastic coverage [27]. Also, the authors proposed a centralized polynomial time algorithm for coverage calculation. Another metric of sensor coverage, exposure, was defined in [26]. The authors also designed a centralized algorithm for calculating the minimal exposure paths.

B. Sensor Deployment

All previous work on sensor deployment either assumes all sensors are static or assumes all sensors are mobile. In the following, we first introduce papers on static sensor deployment followed by papers on mobile sensor deployment. Deployment of static sensor networks has been addressed in [5], [9]. Clouqueur, *et al.*, proposed to deploy sensors in several steps and assumed random deployment in each step [5]. The number of sensors in each step and the cost of deployment were used as a cost function. The authors proposed algorithms to determine the number of steps of sensor deployment such that the cost is low and the desired distribution is obtained. Dhillon, *et al.*, proposed a centralized polynomial-time algorithm to determine sensor distribution such that a minimum number of sensors are deployed and a minimum amount of data are transmitted.

Deployment of mobile sensors has been addressed in [16], [14], [20], [20], [35], [15], [17]. The work in [35] assumes that a cluster head is available to collect the sensor location and determine the target location of the mobile sensors. Howard, *et al.*, proposed an algorithm to deploy mobile sensors into a building from outside, in which sensors are deployed iteratively one by one, utilizing the location information obtained from the previous deployment [20]. The same authors proposed algorithms based on potential field to maximize the monitoring field in [21]. Wang, *et al.*, proposed three algorithms, VEC, VOR, Minimax, and two protocols, to deploy mobile sensors to increase the coverage considering energy efficiency and deployment time. The authors gave insight on how to choose the algorithm and the protocol under different system requirements [16], [14].

The only work, to our knowledge, that addressed a mixed of mobile and static sensors, is our preliminary result of [15], [17].

C. Other Related Work

Other related work includes the study of heterogeneous networks in which not all sensors are the same, for example, networks that have both sensor nodes and relay nodes, which only have communication capability. Hou, *et al* proposed a centralized polynomial-time heuristic algorithm for relay node placement to increase network lifetime [19]. Patel, *et al* designed centralized deployment strategies for sensor nodes, relay nodes, and base stations considering connectivity and coverage [30].

VIII. Conclusions

In this paper, we proposed to use a mix of mobile and static sensors to construct sensor networks to balance cost and sensor coverage. We identified the problem of deploying mobile sensors in a mixed sensor network as a NP-Complete problem, and designed bidding protocols to tackle this problem in a distributed fashion, in which static sensors act as bidders and mobile sensors act as hole-healing servers. Intensive simulation justified our idea of deploying both mobile and static sensors. Users can determine the percentage of mobile sensors to get the most economical deployment of sensors to construct a network satisfying the coverage requirement. Simulation results also showed the efficiency and effectiveness of our proxy-based bidding protocol in placing mobile sensors to achieve high coverage.

In the future, we will work on the deployment of mobile sensors for non-uniform coverage requirements, or for purposes other than coverage. In many applications, some locations are more important than others and may require more sensors for coverage. The bidding protocols presented here can be adapted to this scenario by modifying the rules of assigning base and bid price. In addition, we believe the bidding protocol can be used in many other applications, such as distributed resource allocation.

APPENDIX A

To prove that this problem in NP-hard, we will reduce the following question:

> Given a cubic planar graph G and integer k, does G have an independent set of size k?

to

> Given a square target field partially covered with a number of unit circles (*i.e.* of radius 1) and integer k, can we obtain a complete coverage of the target field by adding k unit circles?

The first question is NP-complete [13], [12].

The second question can be clearly reduced to our optimization problem.

The first stage in translating the problem is to draw the given graph G on an integer grid. We will request that each node has both coordinates divisible by, say, 10.

For each node of the original graph we have a point $(10i, 10j)$; near that point we center a unit circle and inside we create an uncovered area as shown in the figure below; note that this area has 3 special points, and that nearby are another 3 points, each in distance 1 from a corresponding special point; we call them *outer special points*. Outer special points will be located at lines in which at least one coordinate is divisible by 10.

Each edge of the original graph corresponds to a line in which points have at least one coordinate which is an integer divisible by 10. We cover this path with points that are in distance, say, between 1.5 and 1.75 from each other, so that such a ”trail of points” starts at one of the special points of a node gadget and ends at an outer special nodes of another gadget. The trails of points of each edge must be disjoint, and each must have an odd number of points.

We create a little uncovered area around each points on our trails. We finish the construction by covering all areas that we explicitely did not wish to leave uncovered. It follows from the figure that it can be done.

Now, suppose that we had n nodes and m edges in the original graphs and the trails of points of the edges together contain $2K - m$ points. Then we ask if we can achieve the complete coverage by adding $K + n - k$ circles.

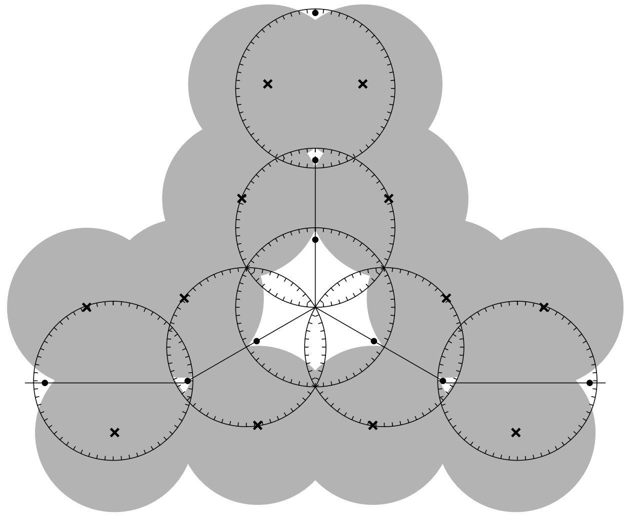


Fig. 18. Gadget for NP-completeness proof. Dots indicate the special points used in the proof; crosses are the locations of already placed sensors.

Suppose that the original question has answer ”YES”, *i.e.* there exists an independent set with k nodes, and thus a vertex cover with $n-k$ nodes. In gadgets corresponding to the vertices of the vertex cover we cover the central area (with the 3 inner special nodes) with a single circle. In gadgets corresponding the the vertices of the independent sets we use three circles to cover the central area and the areas of the outer special points. Now consider an edge; it corresponds to a trail of, say, $2h-1$ points, two of them being outer special points. One of these points is covered by circle used because of the independent set, so we have $2h-2$ points left, and the uncovered areas around these points are placed in such a way that we can cover pairs of them in one circle; thus we use $h-1$ circles; to these circles we can add the circle placed by the independent set rule, so we attribute the use of h circles to this edge. If we add together all circles used that way we get K circles. Hence, we covered the entire area with $K+n-k$ circles and the new question has answer ”YES”.

Now, suppose that the new question has answer ”YES”, so that we obtain a complete coverage by adding $K+n-k$ circles. We will change the placement of the new circles without changing their number to assure some good properties of the placement.

Consider a vertex gadget and its three inner special points.

Suppose that some two of these points are covered with a single circle; then this circle cannot cover any of the outer special points. We move this circle so it covers the entire central area, in particular, all three inner points. We had to have three circles outer special points; we move them so they cover the area of these points as well as the areas of the adjacent points on their respective trails. We call such a vertex a *cover vertex*.

Suppose that the inner special points were covered with three different circles. We move these circles so they cover the central area and the areas of the outer special points. Consider a trail of points of an edge, say, with $2h-1$ points. Suppose that these points are covered with $h+1$ (or more) circles. Then we remove all the circles that cover the areas of these points, we cover them together with the entire inner area and the areas of the outer special points. We call such a vertex an *independent vertex*.

Now consider a trail of points of an edge $\{u, v\}$, say with $2h - 1$ points. Suppose both u and v are independent; in this case both of the outer special points that are at the ends of this trail are covered together with their respective inner points, leaving $2h-3$ points to cover. We remove the circles that cover these points, as well as the outer special point of v that is on the trail; because the latter was covered together with its respective inner point, we are removing the cover of at least $2h - 1$ points, and they are placed in such a way that we surely remove at least h circles. Now we cover $2h - 2$ points on the trail with $h - 1$ circles, and we use one more circle to cover three inner points of the gadget of v. As a result, we changed the classification of v to *cover vertex* without increasing the number of circles.

Now independent vertices form an independent set and cover vertices form a vertex cover. When we consider a trail of points of an edge with $2h - 1$ points, we cover them with h circles, together with an inner special point of the incident independent point (if any). Thus we cover the trails of points of edges, together with the gadgets of the independent points, with K circles, and the remaining $n - k$ circles cover gadgets of the cover vertices, which in turn form a vertex cover. Hence the answer to the original question is ”YES”, we do have an independent set of k nodes.

Appendix B: The proxy-based protocol at sensor s_i

Notations:

S_i: the service list received by sensor s_i.

loc_i, loc'_i: physical position and logical position of s_i, respectively

$\mathcal{P}_i$: the proxy of s_i

(0) *Upon entering Advertising phase:*

set $timer$ to be $advertise\_interval$, and enter *Bidding phase* upon timeout

if s_i is a mobile sensor without proxy **then**

broadcast $service\langle s_i, loc_i, loc_i, base\_price_i, s_i\rangle$

if $s_i = \mathcal{P}_j$ **then**

broadcast $service\langle s_j, loc_j, loc'_j, base\_price_j, s_i\rangle$

(1) *Upon receiving* $service\langle s_j, loc_j, loc'_j, base\_price_j, \mathcal{P}_j\rangle$*:*

add $\langle s_j, loc_j, loc'_j, base\_price_j, \mathcal{P}_j\rangle$ to S_i

(2) *Upon entering Bidding phase:*

set $timer$ to be $bidding\_interval$, and enter *Logical movement phase* upon timeout

calculate bid_i if hole exists

choose the closest/cheapest sensor s_j from S_i where $base\_price_j < bid_i$

send $bidding\langle s_i, loc'_j, bid_i\rangle$ to $\mathcal{P}_j$

(3) *Upon receiving* $bidding\langle s_j, loc'_k, bid_j\rangle$*:*

record it if it has the highest bid

(4) *Upon entering Logical movement phase:*

set $timer$ to be $logical\_interval$, and enter *Hole-exchange phase* upon timeout

if record $bidding\langle s_k, loc'_i, bid_k\rangle$ **then**

send $delegate\langle s_i, loc_i, loc'_i, bid_k\rangle$ to s_k

else if $s_i = \mathcal{P}_j$ and record $bidding\langle s_k, loc'_j, bid_k\rangle$

send $delegate\langle s_j, loc_j, loc'_j, bid_k\rangle$ to s_k

else if $s_i = \mathcal{P}_j$

if hole-exchange with s_m is needed

send $request\langle s_i, s_j, loc_j, priority\rangle$ to $\mathcal{P}_m$

else if it is time for s_j to move **then**

send $notice\langle loc'_j, base\_price_j\rangle$ to s_j

(5) *Upon receiving* $delegate\langle s_j, loc_j, loc'_j, base\_price_j\rangle$*:*

$\mathcal{P}_j = s_i$; record loc_j, loc'_j, $base\_price_j$.

(6) *Upon receiving* $notice\langle loc'_i, base\_price_i\rangle$*:*

move to loc'_i and record $base\_price_i$

(7) *Upon receiving* $request\langle s_j, s_k, loc_k, priority\rangle$*:*

record it if it has the highest priority

(8) *Upon entering Hole-exchange phase:*

if $s_i = \mathcal{P}_m$ and has recorded $request\langle s_j, s_k, loc_k, priority\rangle$

send $confirm\langle s_m, s_k, loc_m\rangle$ to s_j

$\mathcal{P}_k = s_i$; $loc'_k = loc'_m$; $base\_price_k = base\_price_m$

record loc_k

(9) *Upon receiving* $confirm\langle s_m, s_k, loc_m\rangle$*:*

$\mathcal{P}_m = s_i$; $loc'_m = loc'_k$; $base\_price_m = base\_price_k$

record loc_m

References

[1] "Berkeley Sensor and Actuator Center," *http://www-bsac.eecs.berkeley.edu.*

[2] "Wireless Sensing Networks," *http://wins.rsc.rockwell.com*.

[3] "US Naval Observatory (USNO) GPS Operations," *http://tycho.usno.navy.mil/gps.html*, April 2001.

[4] F. Aurenhammer, "Voronoi Diagrams — A Survey of a Fundamental Geometric Data Structure," *ACM Computing Surveys*, vol. 23, pp. 345–405, 1991.

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

[6] D. Koditschek, "Planning and Control via Potenctical Functions," *Robitcs Review I*, pp. 349–367, 1989.

[7] D. Niculescu and B. Nath, "Ad Hoc Positioning Systems (APS) Using AoA," *IEEE INFOCOM*, 2003.

[8] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal and G. S. Sukhatme, "Robomote: Enabling Mobility in Sensor Network," *CRES-04-006 CRES Tech Report*.

[9] S. Dhillon, K. Chakrabarty and S. Iyengar, "Sensor Placement for Grid Coverage under Imprecise Detections," *Proceedings of the International Conference on Information Fusion*, 2002.

[10] F. Zhao and L. Guibas, *Wireless Sensor Networks*, Morgan Kaufmann, 2004.

[11] S. Fortune, D. Du and F. Hwang, "Voronoi Diagrams and Delaunay Triangulations," *Euclidean Geometry and Computers*, 1992.

[12] M. R. Garey and D. S. Johnson, *Computers and Intractability*, 1979.

[13] M. R. Garey, D. S. Johnson, and L. Stockmeyer, *Some Simplified NP-Complete Graph Problems*, 1976.

[14] Guiling Wang, Guohong Cao and Tom La Porta, "Sensor Deployment Protocols," *To appear in IEEE Transaction on Mobile Computing*.

[15] Guiling Wang, Guohong Cao and Tom La Porta, "A Bidding Protocol for Deploying Mobile Sensors," *The 11th IEEE International Conference on Network Protocols (ICNP)*, November 2003.

[16] Guiling Wang, Guohong Cao and Tom La Porta, "Movement-Assisted Sensor Deployment," *IEEE INFOCOM*, March 2004.

[17] Guiling Wang, Guohong Cao and Tom La Porta, "Proxy-Based Sensor Deployment for Mobile Sensor Networks," *IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, October 2004.

[18] W. R. Heinzelman, J. Kulik and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Network," *ACM MobiCom*, 1999.

[19] Y. T. Hou, Y. Shi, H. D. Sherali, and S. F. Midkiff, "Prolonging Sensor Network Lifetime with Energy Provisioning and Relay Node Placement," *Proceedings of IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, 2005.

[20] A. Howard, M. J. Mataric and G. S. Sukhatme, "An Incremental Self-Deployment Algorithm for Mobile Sensor Networks," *Autonomous Robots, Special Issue on Intelligent Embedded Systems*, September 2002.

[21] A. Howard, M. J. Mataric and G. S. Sukhatme, "Mobile Sensor Networks Deployment Using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem," *the 6th International Symposium on Distributed Autonomous Robotics Systems*, June 2002.

[22] L. Hu and D. Evans, "Localization for Mobile Sensor Networks," *ACM MobiCom*, 2004.

[23] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication," *ACM MobiCom*, 2000.

[24] J. Lengyel, M. Reichert, B. Donald, and D. Greenberg, "Real-time Robot Motion Planning using Rasterizing Computer Graphics Hardware," *Proc. SIGGRAPH*, 1990.

[25] Q. Li, M. De Rosa, and D. Rus, "Distributed Algorithms for Guiding Navigation Across a Sensor Network," *ACM MobiCom*, 2003.

[26] S. Meguerdichian, F. Koushanfar, G. Qu and M. Potkonjak, "Exposure In Wireless Ad-Hoc Sensor Networks," *ACM MobiCom*, 2001.

[27] S. Meguerdichian, F. Koushanfar, M. Potkonjak and M. B. Srivastava, "Coverage Problems in Wireless Ad-hoc Sensor Network," *IEEE INFOCOM*, 2001.

[28] N. Patwari and A. Hero III, "Using Proximity and Quantized RSS for Sensor Location in Wireless Location in Wireless Networks," *Workshop on Wireless Sensor Networks and Applications*, 2003.

[29] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Dover Publications, 1998.

[30] M. Patel, R. Chandrasekaran, and S. Venkatesan, "Energy Efficient Sensor, Relay and Base Station Placements for Coverage, Connectivity and Routing," *Proceedings of 24th IEEE International Performance, Computing and Communications Conference*, 2005.

[31] G. J. Pottie and W. J. Kaiser, "Wireless Integrated Network Sensors," *Communications of the ACM*, May 2000.

[32] A. Savvides, C. Han and M. B. Strivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," *ACM MobiCom*, 2001.

[33] G. T. Sibley, M. H. Rahimi and G. S. Sukhatme, "Robomote: A Tiny Mobile Robot Platform for Large-Scale Sensor Networks," *2002 IEEE International Conference on Robotics and Automation*, 2002.

[34] K. Sohrabi, J. Gao, V. Ailawadhi and G. J. Pottie, "Protocols for Self-Organization of A Wireless Sensor Network," *IEEE Personal Communication*, vol. 7, no. 5, pp. 16–27, October 2000.

[35] Y. Zou and K. Chakrabarty, "Energy-Aware Target Localization in Wireless Sensor Networks," *Proc. IEEE Int. Conf. Pervasive Computing and Communications*, 2003.

[36] Y. Zou and K. Chakrabarty, "Sensor Deployment and Target Localization Based on Virtual Forces," *IEEE INFOCOM*, 2003.