Approximation Algorithms for Two Optimal Location Problems in Sensor Networks

Alon Efrat
Department of Computer Science
University of Arizona
Tucson, Arizona 85721
Email: alon@cs.arizona.edu

Sariel Har-Peled
Department of Computer Science
University of Illinois
Urbana, Illinois 61801
Email: sariel@uiuc.edu

Joseph S. B. Mitchell
Applied Mathematics & Statistics
Stony Brook University
Stony Brook, New York 11794-3600
Email: jsbm@ams.sunysb.edu

***Abstract*— This paper studies two problems that arise in optimization of sensor networks: First, we devise provable approximation schemes for locating a base station and constructing a network among a set of sensors each of which has a data stream to get to the base station. Subject to power constraints at the sensors, our goal is to locate the base station and establish a network in order to maximize the life-span of the network.**

Second, we study optimal sensor placement problems for quality coverage of given domains cluttered with obstacles. Using line-of-site sensors, the goal is to minimize the number of sensors required in order to have each point "well covered" according to precise criteria (e.g., that each point is seen by two sensors that form at least angle α, or that each point is seen by three sensors that form a triangle containing the point).

I. Introduction

Consider a wireless sensor network with a large number of deployed sensors, each capturing data on a continuous basis. The sensors may be capturing video data, audio data, environmental data, etc. There is a base station that collects all of the data streams from all of the sensors. Each sensor passes data packets along some route in a network, from sensor to sensor, so that all data arrives at the base station. Since each sensor is generally powered by some form of battery, the duration of the sensor node is determined in large part by its power dissipation rate and energy provision. A fundamental issue associated with wireless sensor networks is maximizing their useful lifetime, given their power constraints. This can be significantly affected by the location of the base station as well as the forwarding protocols used (i.e. which sensor forwards packages of which other sensor) in establishing the network. Hou[1] (see also [1]) has suggested the use of the length of time until the first sensor exhausts its battery as a definition of the *life-span* of the system. In the paper, we show how to find a location of the base station such that the life-span of the system is optimized to within any desired approximation bound. Specifically, we give a method for locating the base station that provably obtains a life-span of at least $(1-\varepsilon)$ times that of the optimal life-span, where $\varepsilon > 0$ is any pre-determined fixed value. The algorithm is based on solving $O(n\varepsilon^{-4}\log^2(n/\varepsilon))$ instances of a linear programming problem. It is simple and easy to implement.

[1]Thomas Hou, personal communication, 2003.

We also study another optimal location problem in sensor networks: How does one choose locations of sensors for line-of-sight coverage of a given region, under the assumption that a point is only "covered" if it is "well seen"? We consider two definitions of "well seen": A point p is well seen (or *robustly covered*) if either (a) there are two sensors that see p, and these sensors are separated by angle at least α with respect to p; or (b) there are three sensors that see p and they form a triangle that contains p. The objective is to minimize the number of sensors to achieve robust coverage. Our results on this problem give efficient approximation algorithms for minimizing the number of sensors.

This second problem is a variant of the classical art gallery problem, in which one is to place the fewest sensors ("guards") to see all points of a certain geometric domain. Art gallery problems have been studied extensively; see, e.g., [2], [3], [4], [5] for surveys. The algorithmic problem of computing a minimum number of guards is known to be NP-hard, even if the input domain, $\mathcal{D}$, is a simple polygon [6]. Thus, efforts have concentrated on approximation algorithms for optimal guarding problems. Ghosh [7] gave an $O(\log n)$-approximation for computing a minimum number of vertex guards for a polygon; his method is based on standard set cover results. More recently, researchers [8], [9] have applied the set cover methods of Brönnimann-Goodrich [10], which exploits finiteness of VC-dimension. In particular, Efrat and Har-Peled [8] obtain an $O(\log k^*)$-approximation algorithm for guarding a polygon with vertex guards, using time $O(n(k^*)^2\log^4 n)$, where k^* is the optimal number of vertex guards. They also apply their technique to guards that may lie at any point of a dense grid; the running time then has a factor polylogarithmic in the grid density. (No approximation algorithm is known if the guards are completely unrestricted and all of the polygon is to be guarded.) Further, the methods can be applied to polygons with holes and to 2.5D polyhedral terrains, still with polylogarithmic approximation factors. Cheong et al. [11] have recently shown how to compute k guards in order to optimize (approximately) the total area seen by the guards. The triangle-guarding coverage problem we study is related to recent work of Smith and Evans [12], as we discuss in Section III-B.

II. Preliminaries

In the following, we consider a wireless *sensor* s to be a point in the plane that has to transmit a certain amount of information from the sensor to a base station. For example, a *sensor* may consist of a thermostat, camera, or other data-gathering device, together with a transmitter and receiver and a battery. Each sensor needs to send its data to the base station. The cost of transmission, in terms of energy drained from the battery, is a function of the distance over which data is transmitted times the amount of information sent. This cost is by far the major component that affects the battery life.

A detailed power dissipation model for each component at a wireless sensor node can be found in [13]. It turns out that, for sensor networks that are sparsely deployed over a wide geographical area, the power consumption by data communication (*i.e.*, receiver and transmitter) is the dominant factor among all the power consumption components [14]. This disparity is illustrated in [15], which shows that, under Rayleigh fading and fourth-power distance loss, the energy cost of transmitting 1 KB a distance of 100 m is approximately the same as executing three million instructions on a 100 MIPS/W processor.

The power dissipation at a receiver can be modelled as in [16]:

$$p_r = \alpha_r r_i \ , \tag{1}$$

where r_i (in bps) is the incoming rate of received data bit-stream and the parameter $\alpha_r = 135$ nJ/b [13].

The power dissipation at the transmitter can be modelled as:

$$p_t(r) = [\alpha_{t1} + \alpha_{t2} r^K] r_o \ , \tag{2}$$

where $p_t(r)$ is the power dissipated in node s_1 when it is transmitting to node s_2, and r is the Euclidean distance between s_1 and s_2. Here α_{t1} is a distance-independent constant term, and α_{t2} is a constant term associated with the distance term, where K is the path loss index[2] with $2 \le K \le 4$ [16], and r_o is the output bit rate of the transmitter, which is the aggregated rate of both local and received data traffic.

The situation becomes considerably more interesting when we have several sensors $s_1, \ldots, s_n$. Each sensor s_i sends a stream of one unit of information per second to the base station (e.g., 10KB/s), and each sensor starts with a battery having b_i units of energy. Here, we speak of each sensor having one *unit* of information to transmit per second; the unit of data is any particular fixed data size, which could be very large (e.g., several images). (The case in which sensors have various different amounts of information to transmit is an easy extension of our discussion; it is deferred to the full version of this paper.) Of course, each sensor can send all of its information directly to the base station, but it can also forward a portion of its transmitted information to other sensors, which can then relay it (directly or indirectly) to the base station.

[2]The exponent K is closer to 4 for a low-lying antenna and near-ground channels [16]. This can be attributed to the partial signal cancellation by a ground-reflected ray.

We follow Hou et al. [1] in optimizing the *time-span* $T(b)$ of the system, defined as the maximal time at which all nodes remain alive (powered), given that the base station is position at a point b. Note that we assume that every sensor has to send all of the information it generates to the base station, and it has to send it in real time, with no ability to buffer it and send only a fraction of it.

Thus, we are looking for a *transmission scheme*; namely, we explicitly assign for each sensor what information it should send to each other sensor (and to the base station), so that: (i) each sensor transmits one unit of information per second, and (ii) all of the information a sensor transmits arrives finally to the base station. Our problem is to find a transmission scheme that maximizes the life-span of the network. Note that using several different transmission schemes, one after the other, instead of a single transmission scheme cannot improve the life-span of the network. Indeed, we can combine all those transmission schemes together into a single transmission scheme with the same life-span. Specifically, if we have schemes $T_1, \ldots, T_k$ serving for $t_1, \ldots, t_k$ units of time, one after the other, then $\sum t_i T_i / L$ is a single transmission scheme with the same life-span, where $L = \sum_i t_i$.

So far, we have not addressed the question of where the base station is located. If we know its location b, then $T(b)$ (and the corresponding protocol) can be solved using linear programming. We briefly describe this solution for the sake of completeness.

Lemma 1: Given a set of sensors $\mathcal{S} = \{s_1, \ldots, s_n\}$, and a base station $\mathbf{b}$ such that each sensor transmits one unit of information to the base station, one can compute a transmission scheme that maximizes the life-span $T(b)$ of the network by solving a single linear programming instance.

Proof: Consider the complete directed graph defined over $\mathcal{S} \cup \{\mathbf{b}\}$. We assign each edge uv in this graph its price according to its length, namely $c_{uv} = p_t(\|uv\|)$. We now define a variable x_{uv} that specifies the amount of information transmitted per second for each edge uv in this graph. It is now straightforward to encode the standard conditions on this graph as a set of linear inequalities. In particular, the amount of data "flow" (information per second) into a sensor is equal to the amount of flow leaving the node, plus one (the one additional unit of flow is the sensor's own information that needs to be transmitted), and the flows are always non-negative. In order to model the life-span objective, we let t be the life-span of the network. Clearly, for a node u, the life-span of the battery of u has to comply with the inequality $\left(\sum_{v \in V} x_{uv} c_{uv}\right) t \le b_u$, where V is the set of vertices of the graph, and b_u is the amount of energy in the battery of u. Dividing by t, we have $\sum_{v \in V} x_{uv} c_{uv} \le b_u / t$. Introducing a new variable $T = 1/t$, we have $\sum_{v \in V} x_{uv} c_{uv} \le b_u T$, which a linear constraint. Thus, we get a linear system of constraints, and our objective function is to minimize the value of the variable T. Solving this system, which is an instance of linear programming, yields the desired network having an optimal life-span. ■

A. A Constant-Factor Approximation

Our goal is to compute a location, b, of the base station and a corresponding transmission scheme in order to maximize the life-span of the network. We have seen that if the location b is known, an optimal transmission scheme is found using linear programming. Now we want to optimize over all choices of b.

For simplicity, we assume in the following that $p_t(r) = r^4$, though this assumption can easily be modified to accommodate different constants used in Eq. (1).

Lemma 2: Given a set of sensors $\mathcal{S} = \{s_1, \ldots, s_n\}$, one can compute a location for the base station and a transmission scheme that approximates up to a constant factor the life-span of the optimal solution by solving n linear programming instances.

Proof: We claim that restricting the base station to be placed at the location of one of the n sensors, and then taking the best from among these n choices, results in a constant-factor approximation. Indeed, consider the optimal placement for the base station $\mathbf{b}_{opt}$, and the optimal transmission scheme T_{opt}. Let s be the closest sensor to $\mathbf{b}_{opt}$, and consider the solution generated by keeping the same flow as T_{opt} while moving the base station to s. Clearly, each edge becomes longer by at most a factor of two, and, as such, the life-span of each battery goes down by at most a factor of $p_t(2r)/p_t(r) = 2^4 = 16$. ■

B. An ε-Approximation Algorithm

We now show how to improve the approximation factor from 16 (given in the proof of Lemma 2) to $(1+\varepsilon)$, for any fixed $\varepsilon > 0$. In the following, we let t_{opt} be the life-span of an optimal solution.

We show that one can discretize the search space for the optimal choice of b. Thus, we would have to check only a small number of possible solutions, the best one of which yields the desired approximation. In particular, we show that the energy used by each sensor to transmit directly to the base station is one of small number of possible levels.

Indeed, if we take an optimal transmission scheme and use it only for $(1-\varepsilon/4)t_{opt}$ time, then the battery of each sensor s_i would still have at least $(\varepsilon/4)b_i$ energy left in it. We use this "leftover" energy as follows: the sensor s_i sends information to several other sensors and to the base station. Let α_i be the total energy of s_i that is used in an optimal transmission scheme to send information directly to the base station. Next, we change the energy allocation of b_i. Each regular link that was allocated energy β_{ij} is now allocated energy $(1-\varepsilon/4)\beta_{ij}$. Furthermore, the link to the base station is allocated energy $\alpha'_i = m \cdot \varepsilon b_i/4$, where $m = \lceil 4\alpha_i/(\varepsilon b_i) \rceil$. Clearly, $\sum_j (1-\varepsilon)\beta_{ij} + \alpha'_i \leq b_i$, and thus this allocation of energy is valid (i.e., no sensor exhausts its battery).

For a technical reason that will become clear shortly, we want to guarantee that no station sends directly to the base station a quantity of information that is "too small."

Lemma 3: Given a transmission scheme T with $\mathrm{lifespan}(T)$, one can compute a transmission scheme, T', such that any sensor s_i, under the transmission scheme T' sends β_i bits of information (a second) to the base station directly, and β_i is either 0 or larger than ε/n^4. Furthermore, $\mathrm{lifespan}(T') \geq (1-\varepsilon)\mathrm{lifespan}(T)$.

Proof: We first note that T can be decomposed into $O(n^2)$ paths, so that the total transmission scheme can be described as the summation of the flows along these paths, where for each path π the flow starts at an initial sensor and ends in the base station. (For "standard" network flow this is a well-known fact.)

To see why this is true in our case, consider a transmission scheme as a network flow f in the graph G, where the sink in G is the base station. We start with an empty flow g, and we increase it until it reaches f. For each edge e, let $f(e)$ denote the original flow along e. Indeed, pick any sensor v in G for which g does not carry all of the flow from it to the sink, with respect to the flow of f. In particular, let η_v be the amount of flow that is not handled yet by g. Compute a path π from v to the sink (i.e., base station), such that each of the edges $e \in \pi$ carries less flow in g than the corresponding flow in T (i.e., $f(e) - g(e) > 0$). Consider the edge e along π that minimizes $f(e) - g(e)$, and let u_π be this capacity. We transmit $\min(u_\pi, \eta_v)$ bits of information along π starting from v. Next, add the flow along π to g. Clearly, at each iteration of this augmentation scheme, either a bottleneck edge (e') is being saturated, or alternatively, a node is being completely served (i.e., all the information that it needs to transmit is being transmitted to the base station). It follows that we can perform only $O(n^2)$ iterations of this algorithm, and this produces the required decomposition of T into $O(n^2)$ path flows.

Returning to the proof of the claim, we need to rechannel every direct transmission to the base station that is smaller than ε/n^4. Thus, assume that s_i sends β_i bits to the base station directly, and $\beta_i \leq \varepsilon/n^4$. Now, we attach β_i to the path flow that goes through (or starts at) s_i with maximum flow on it. Clearly, this path flow must have a flow of at least $1/n^2$ (since s_i sends 1 unit of information to the base station, and there are at most n^2 path flows in T). Doing this for each sensor results in a new transmission scheme T', which is the required transmission scheme.

Indeed, consider a path flow π, starting from a sensor s_i, and observe that, if it was modified, the initial flow along it was at least $1/n^2$. Each modification increased the flow by at most $n \cdot \varepsilon/n^4$. It follows that the flow along the path increased by at most a factor of $(1+\varepsilon/n)$. As such, the life-span of this path flow by itself (i.e., we allocate this path the same energy as in the energy allocated to it by T) shrinks by at most a factor of $1/(1+\varepsilon/n)$, which implies the claim, as this implies that $\mathrm{lifespan}(T') \geq \mathrm{lifespan}(T)/(1+\varepsilon/n) \geq (1-\varepsilon)\mathrm{lifespan}(T)$. ■

Lemma 4: Given a transmission scheme T with $\mathrm{lifespan}(T)$, one can compute a transmission scheme T' such that any station s_i under the transmission scheme T' sends either 0 or $(1+\varepsilon/4)^j \varepsilon/n^4$ units of information (per second) to the base station directly, for $1 \leq j \leq M$, where $M = O((\log(n/\varepsilon))/\varepsilon)$. Furthermore, $\mathrm{lifespan}(T') \geq (1-\varepsilon)\mathrm{lifespan}(T)$.

Proof: We apply Lemma 3 to T with $\varepsilon/3$, and let T'' denote the resulting transmission scheme, such that $\text{lifespan}(T'') \geq (1-\varepsilon/3)\text{lifespan}(T)$. Next, we go over the sensors, and for each sensor that sends $\beta_i > 0$ information directly to the base station, we round it up to the closest value $(1+\varepsilon/4)^j \varepsilon/n^4$ that is larger than β_i. This shortens the life-span of the sensor by at most a factor of $1/(1+\varepsilon/4) \geq (1-\varepsilon/2)$. Thus, for the resulting transmission scheme T', we have $\text{lifespan}(T') \geq (1-\varepsilon/2)\text{lifespan}(T'') \geq (1-\varepsilon/2)(1-\varepsilon/3)\text{lifespan}(T) \geq (1-\varepsilon)\text{lifespan}(T)$. ■

Since by Lemma 2 we can approximate the network life-span within a constant factor of optimal, it follows that we can assume that we know the optimal life-span up to a factor of, say, $1-\varepsilon/4$. Indeed, let t' be the life-span computed by Lemma 2, and consider $t_i = t'(1+\varepsilon/8)^i$. for $i = 1, \ldots, M$, where $M = \left\lceil \log_{1+\varepsilon/8} 32 \right\rceil = O(1/\varepsilon)$, as can be easily verified. Clearly, there exists a ρ, with $1 \leq \rho \leq M$, such that $t_\rho \leq t_{opt} \leq (1+\varepsilon/8)t_\rho$.

By checking all $O(1/\varepsilon)$ possibilities, we can assume that we know t_ρ, where $t_\rho \leq t_{opt} \leq (1+\varepsilon/8)t_\rho$. Similarly, by the above discussion, we know that the energy α'_i used in the transmission from a sensor s_i to the base station is one of $O(1/\varepsilon)$ possible values: $\gamma'_{i1}, \ldots, \gamma'_{il}$, where $\gamma_{ij} = j \cdot \varepsilon b_i/4$ and $l = \lceil 4/\varepsilon \rceil$. Finally, we can also guess the amount of information begin sent, β_i, since it belongs to one of $O(\log(n/\varepsilon)/\varepsilon)$ possibilities. Each of these canonization steps decreases the life-span of the network by, say, a factor of $(1-\varepsilon/10)$. Let $\mathcal{T}$ be the resulting canonical transmission scheme. Clearly, $\text{lifespan}(\mathcal{T}) \geq (1-\varepsilon/10)^3 t_{opt} \geq (1-\varepsilon)t_{opt}$.

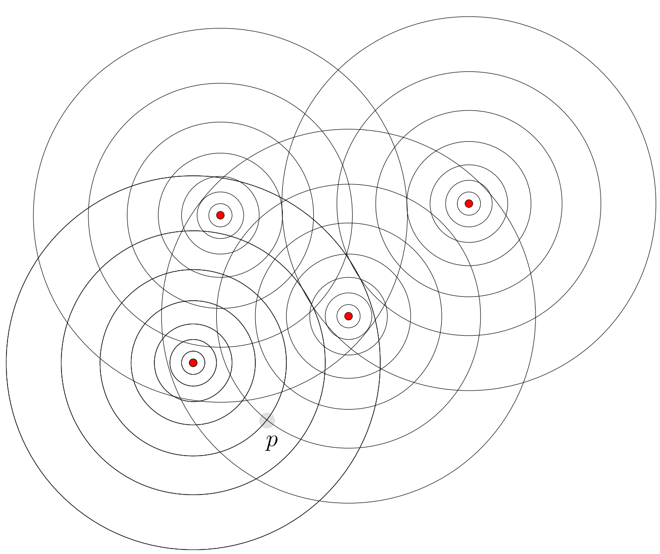

Fig. 1. Sensors and their different level of transmission. An intersection point like p is a candidate location for the base station.

Thus, for each sensor s_i, we have a transmission scheme for which the parameters of the sensor is one of $O(\log(n/\varepsilon)/\varepsilon^3)$ triples of possible configurations for this sensor. Once we know these three parameters, we can compute the maximum radius of transmission of the sensor. Let $r(\alpha_i, \beta_i, t_\rho)$ denote this radius. Imagine now that we draw the set $\mathbf{R}_i$ of disks with these radii around s_i, and let C be the resulting set of circles. See Figure 1. This set has $O((n/\varepsilon^3)\log(n/\varepsilon))$ circles, and it is easy now to verify that the base station of $\mathcal{T}$ can be moved to one of the vertices of $\mathcal{A}(C)$ without changing the life-span of $\mathcal{T}$. Since the arrangement $\mathcal{A}(C)$ has at most $O((n/\varepsilon^3)^2 \log^2(n/\varepsilon))$ vertices, it follows that checking for each one of these vertices the optimal possible solution, using the algorithm of Lemma 1, results in computing a transmission scheme with life-span at least $(1-\varepsilon)t_{opt}$.

Theorem 5: Given a set of sensors $\mathcal{S} = \{s_1, \ldots, s_n\}$ and a parameter $\varepsilon > 0$, one can compute a location of the base station and a transmission scheme such that the network's life-span is at least $(1-\varepsilon)t_{opt}$, where t_{opt} is the life-span of an optimal transmission scheme for $\mathcal{S}$. This algorithm requires $M = O((n/\varepsilon^3)^2 \log^2(n/\varepsilon))$ preprocessing and needs to solve M instances of linear programming.

C. A Faster Algorithm

In this section we present a faster approximation algorithm. Consider the set of disks $\mathbf{R}_i$ discussed in the previous section. For each s_i, and each disk of $\mathbf{R}_i$, we place $4\pi/\varepsilon$ equally spaced points along the boundary of this disk. Let Γ be the resulting set of points (for all disks).

Lemma 6: Let t_0 be the optimal life-span of the system when the base station is constrained to lie at one of the points of Γ. Then, $t_0 > (1-4\varepsilon)t_{opt}$.

Proof: Let v be a vertex of the arrangement of the disks of $\{\mathbf{R}_i\}$ that maximizes the system's life-span, under the optimal transmission scheme. Let D be the smallest disk containing v, and let r be its radius. By shifting the base station from v to v', we increase the transmission distance from each sensor to the base station by a factor of at most $(1+\varepsilon/2)$, thereby draining a factor of at most $(1+\varepsilon/2)^4$ more energy from its battery, and thus decreasing the system life-span by at most $(1+\varepsilon/2)^4$. ■

Motivated by Lemma 6, we solve a linear programming problem for each point of Γ. Since $|\Gamma| = O(n/\varepsilon^4 \log^2(n/\varepsilon))$, the asserted bound is obtained. To summarize, we obtain

Theorem 7: Given a set of sensors $\mathcal{S} = \{s_1, \ldots, s_n\}$ and a parameter $\varepsilon > 0$, one can compute a location of the base station and a transmission scheme such that the network life-span is at least $(1-\varepsilon)t_{opt}$, where t_{opt} is the life-span of an optimal transmission scheme for $\mathcal{S}$. This algorithms requires $M = O(n/\varepsilon^4 \log^2(n/\varepsilon))$ preprocessing, and needs to solve M instances of linear programming.

III. Sensors Placement Problems

In this section we study some problems related to locating sensors in a given region, Q, in a way that guarantees "robust" visibility (line-of-sight) coverage of Q. We are motivated by applications in which it is not enough to ensure that each point $p \in Q$ is seen by a sensor. We require that more than one sensor sees p and that these sensors see p from substantially different directions. We formalize this notion of robust sensing in two different ways, discussed in the two subsections below.

Though not specifically detailed here, our algorithms based on the powerful Brönnimann-Goodrich [10] set cover methodolgy apply also with a wide variety of constraints on the sensors, such as range constraints and view-angle constraints. Also, we focus here on the case in which Q is a simple polygon; however, modifications of the coverage algorithms allow them to apply to more general domains, such as three-dimensional structures (buildings) and terrains.

At a high level, the core of the Brönnimann-Goodrich [10] technique is the idea of distributing weights among candidate locations of sensors, and selecting subsets of these candidate locations at each main iteration of the algorithm. The larger the weight of a candidate location, the greater the chance that this location will be selected, since the algorithm selects new locations randomly according to a probability distribution given by the (normalized) weights. If the resulting set of selected locations satisfies the covering criteria (e.g., Q is 2-guarded at angle α or is triangle-guarded), then we are done. Otherwise, we select a point $q \in Q$ that is not yet "covered," increase the weights of other candidate locations that "see" q, and continue. Brönnimann-Goodrich [10] prove a bound on the number of iterations of the main algorithm and prove that it yields a cover with small size compared to optimal; we refer the reader to their paper for details.

A. Two-Sensors Visibility

Here we assume that we are given an angle $\alpha > 0$, and two regions P and Q, where $Q \subseteq P$. Our goal is to place sensors within P that enable a "robust" coverage of the region Q.

We say that a sensor g *sees* a point $p \in Q$ if the segment connecting them does not cross ∂P. We say that p is *2-guarded at angle* α by sensors g_1 and g_2 that each see p if the angle $\angle g_1 p g_2$ is in the range $[\alpha, \pi - \alpha]$. The motivation for this definition is, for example, use of sensors in localization of an object (e.g., a particular animal) in a terrain, where two sensors that each see the animal are needed to find the coordinates of the animal. Moreover, the location of the animal and these two sensors should not be nearly collinear, as this leads to errors in estimating the location of the animal. Thus, for robust localization, we desire double coverage of each point of Q by two sensors that are well separated angularly with respect to the point being covered.

We constrain all sensors to be located inside the region P, which, for the sake of simplicity, we will assume here to be a simple polygon. We say that a set of sensors $G \subset P$ *guards* Q (under standard visibility) if each point $p \in Q$ is seen by at least one sensor of G. We say that Q is *2-guarded at angle* α by G if for each point $p \in Q$ there are two sensors g_1, g_2 in G such that p is 2-guarded at angle α. We note that there may be no set G of guards that 2-guard Q at angle α, or, in some cases, such a set exists but is infinite in cardinality.

We present the following two-phase algorithm for finding a small set G of approximately smallest cardinality among sets of guards that 2-guard Q. In the fist phase, we find a set $G_1 \subset P$ of approximately smallest cardinality that guards Q (under standard visibility); this can be done, e.g., using the techniques of [8]. In the second phase, we find another set $G_2 \subset P$ such that $G_1 \cup G_2$ 2-guards Q, but with angle $\alpha/2$, rather than α.

The following lemma is used in justifying this approach.

Lemma 8: Let G^* be a set of k^* sensors that 2-guard Q at angle α. Let G_1 be a set of sensors that sees Q (under standard visibility). Then, for any point $p \in Q$ there exist sensors $g_1 \in G_1$ and $g_2 \in G^*$ that 2-guard p at angle $\alpha/2$.

Proof: Let $g_2, g_2' \in G^*$ be two sensors that 2-guard p at angle $\geq \alpha$. Let $g_1 \in G_1$ be a sensor that sees p. If g_1 is (clockwise around p) between g_2 and g_2', then it must form an angle $\geq \alpha/2$ with one of them; otherwise, it must form an angle $\geq \alpha$ with one of them. ■

This lemma justifies the second phase of our algorithm, in which we find a second set of sensors, G_2, that, together with G_1, 2-guard Q at angle $\alpha/2$. Moreover, using the method of [10] we can be assured that the cardinality of what we produce, $G_1 \cup G_2$, is $O(k^* \log k^*)$. Since the algorithm requires that the candidate guards are picked from a discrete set, we impose a grid, Γ, in P, and pick vertices from the grid. The algorithm follows almost identically the one used in [8], so we omit its full details here. We should emphasize, however, that the grid is not maintained explicitly, and is just used to select vertices using an oracle that reports the number of grid points inside certain regions.

The technique of maintaining the weights is essentially the same as in [8], so only modifications are described here. We need also to describe how one can find a witness point q that not 2-guarded by a current set G of guards. For this we establish $2\pi/(\alpha/2)$ equally-spaced rays emerging from each guard of G. These rays subdivide Q into regions. We overlay these cones with the visibility arrangement of P. Let $\mathcal{A}$ be the resulting arrangement. For every pair of two points p, p' in the same region of $\mathcal{A}$, if we create sets of cones around p and p' by placing $2\pi/(\alpha/2)$ equally-spaced cones around them, then a cone in some direction contains a sensor $g \in G$ that is visible to p if and only if the cone of the same direction of p' contains a sensor visible to g'. Thus, by checking whether each of the cones contains a sensor visible to p, we can approximately decide if p is either $\alpha/2$-guarded, not α-guarded, or that the answer is not clear. Moreover, we further subdivide each cell of $\mathcal{A}$ so that the circular order of the sensors around each point of a cell of $\mathcal{A}$ is the same. Such a subdivision can be obtained by passing a line through each pair of sensors of G. Then it is not hard to observe that it is enough to consider the maximal and minimal guards (in angular order) among the guards that share the same cell of $\mathcal{A}$, in order to determine which of the points of the cell are actually $\alpha/2$-guarded.

Theorem 9: Given P and Q as above, an angle α, and a grid Γ of edge-length δ inside P, we can find a set G of sensors in P such that G 2-guards Q at angle $\alpha/2$, and $|G| = O(k_{opt} \log k_{opt})$, where k_{opt} is the cardinality of smallest set of vertices of Γ that 2-guard Q at angle α. The running time of the algorithm is $O(nk_{opt}^4 \log^2 n \log m)$, where m is the number of vertices of $\Gamma \cap P$.

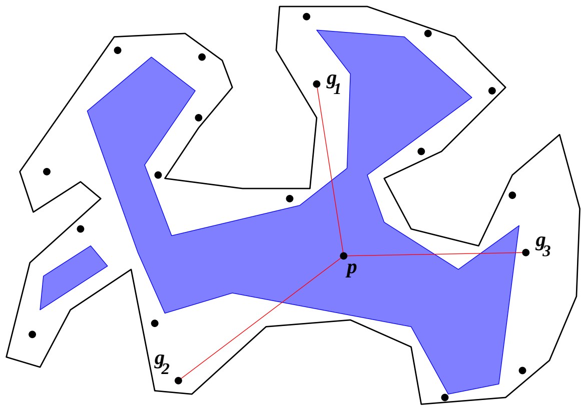

Fig. 2. Example of a point p that is triangle-guarded by g_1, g_2, and g_3.

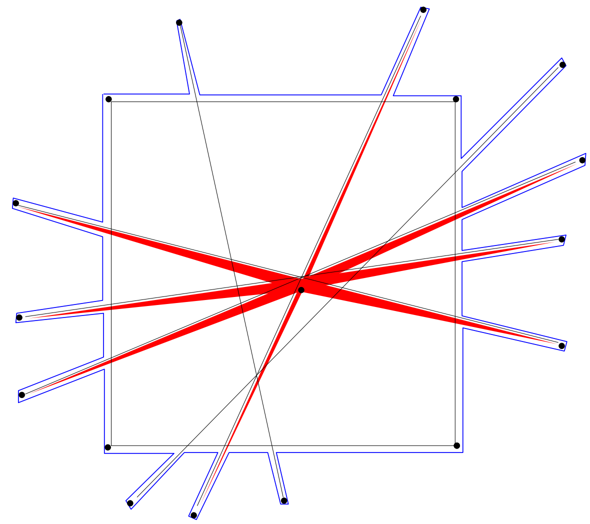

Fig. 3. The proof of hardness showing the reduction from the minimum line covering problem (MLCP). The polygon P is shown in blue; the region Q is a slightly smaller copy of P, just inside P. Just below one vertex in the arrangement of $\mathcal{L}$, we show a black dot representing a guard associated with the vertex; the guard, together with the guards in the "tips" of the "spikes" of P participates in the triangle-guarding of the red region (a union of three skinny triangles).

B. Triangle-Guarding

Given P and Q as before, here we seek a set of guards G such that each $p \in Q$ is seen by three sensors and p is contained in the triangle they span. See Figure 2. This problem can be presented in a slightly different way. We associate with each $p \in Q$ an infinitesimally small opaque disk, and we require that we find sensors that see all points on the perimeter of the disk. We can transform this problem to the following problem. Consider the region $\mathcal{Q} = Q \times [0, 2\pi)$ in a 3-dimensional space. The point $(p, 0)$ is associated with the lowest (i.e., with smallest y-coordinate) point of the tiny disk centered at p. Thus, we say that $(p, \theta) \in \mathcal{Q}$ is seen by g if p is seen by g, and the angle $|\theta - \theta_1| \leq \pi/2$, where θ_1 is the angle between the segment pq and the negative y-axis.

A similar notion of "triangle guarding" is studied in Smith and Evans [12], who consider the case in which the boundary of P is transparent, so does not limit the coverage of a sensor, as it does in our case. They show that a simple polynomial-time greedy algorithm optimally solves this version of the problem. (They also show that if the boundary of Q is opaque and must be triangle-guarded with vertex guards, the problem is NP-hard.)

We begin by showing that the problem we are solving approximately is in fact NP-hard to solve exactly:

Theorem 10: Deciding if k sensors within P suffice to triangle-guard Q is NP-hard.

Proof: Our reduction is from the MINIMUM LINE COVERING PROBLEM (MLCP), which is known to be NP-hard [17]. The MLCP has an input of a set of n non-parallel lines in the plane, and the problem asks us to find a smallest number of points that *cover* the lines, in the sense that each line is incident on at least one point.

Consider an instance of MLCP given by lines $\mathcal{L} = \{\ell_1, \ldots, \ell_n\}$, no two of which are parallel, and let B be a large square that contains all vertices in the arrangement of $\mathcal{L}$. Clip each line ℓ_i outside B, and draw a polygon P just outside the union of B and the ends of the clipped lines. The resulting polygon P is a box with "spikes," having complexity $O(n)$. Define Q to be a slightly smaller copy of P, just inside P. In order to triangle-guard the "tips" of Q, a sensor must be placed at the tip of each spike of P. Also, four sensors must be placed in the corners of B in order to triangle-guard the interior of B. What remains is to decide where to place additional sensors so that they, in conjunction with the sensors at the tips of P, triangle-guard the spikes of Q that extend outside of B into the spikes of P. A sensor placed just below a vertex v in the arrangement allows triangle-guard coverage of all $2i$ of the spikes of Q that correspond to the i lines of $\mathcal{L}$ passing through v. (See Figure 3.) Conversely, in order to triangle-guard a spike of Q, there *must* be a sensor placed (approximately) on the line that corresponds to the spike. Thus, k points suffice to cover the lines $\mathcal{L}$ if and only if $2n + k + 4$ sensors suffice to triangle-guard Q. ■

For our approximation algorithm, we again apply an algorithm based on [10], after imposing a grid Γ in P (not explicitly maintained). Weights associated with grid points and selection of points from the grid are again done as in [10].

In order to determine whether or not a set G of k sensors triangle-guards Q, we do the following.

Consider a point $q \in Q$ that is seen (using standard visibility) by G, and let $G' \subseteq G$ be the set of sensors $\{g \in G | g \text{ sees } q\}$. Then q is triangle-guarded by G' if and only if q is in the convex hull $CH(G')$. Based on this observation, we propose the following algorithm:

1) Compute the visibility arrangement $\mathcal{A}$ of G in P. This is a subdivision of P into regions, so that each each region is seen by the same set of sensors of G. This arrangement is formed by a set of line segments within P, where the segments are determined by "ray shooting" as follows. Shoot a ray from each $s_i \in G$ through each vertex of $v \in \partial P$ that is seen by v, and let $\xi_i(v)$ be the first point on ∂P where the ray exits P. The segments of

the arrangement $\mathcal{A}$ are given by $(s_i, \xi_i(v))$, the (clipped) *visibility rays*. It is known [18] that the complexity of $\mathcal{A}$ is $O(nk^2)$, and it can be computed in time $O(nk^2 + n \log n)$.

2) Compute the overlay arrangement of $\mathcal{A}$ and ∂Q. This can be done by triangulating each face of $\mathcal{A}$ (in time linear in the complexity of $\mathcal{A}$), and then tracing ∂Q within the arrangement $\mathcal{A}$, starting at any vertex of Q. Since each segment of ∂Q can intersect only 2 visibility rays of each sensor $s_i \in G$, the total number of intersection points of ∂Q and triangles of $\mathcal{A}$ is $O(nk)$. (Note that if we discover that no portion of Q lies in a face of $\mathcal{A}$ corresponding to visibility from at least three guards, then clearly Q is not triangle-guarded.)
3) Let $\mathcal{A}'$ denote the resulting overlay arrangement, $\mathcal{A} \cap Q$, restricted to Q. We triangulate each face of $\mathcal{A}'$. For each triangle $c \in \mathcal{A}'$, we find the set G_c of sensors of G that see c. We compute the convex hull, $CH(G_c)$, and in $O(\log k)$ time determine if ∂c intersects $CH(G_c)$. If so, then any point of $c \setminus CH(G_c)$ (e.g., in the neighborhood of one of the intersection points) represents a witness point to the fact that Q is not triangle-guarded by G. Otherwise, we determine if c if fully inside or fully outside $CH(G_c)$. A naive implementation of this idea would run in time $O(nk^3 \log n)$, but a factor of k can be saved by noting that two sets of sensors seeing two neighboring cells of $\mathcal{A}'$ are different by at most one sensor, and hence the convex hulls can be maintained dynamically: we traverse $\mathcal{A}'$ and at each time update the convex hull by adding or deleting one sensor. The details are straightforward and omitted here.

Theorem 11: Given P and Q as above, and a grid Γ of edge-length δ, we can find a set G of sensors in P, that triangle-guard Q, with $|G| = O(k_{opt} \log k_{opt})$, where k_{opt} is the cardinality of smallest set of vertices of Γ that triangle-guard Q. The running time of the algorithm is $O(nk_{opt}^2 \log^2 n \log m)$, where m is the number of vertices of $\Gamma \cap P$.

Acknowledgment

We thank Thomas Hou for introducing us to the problem, and many helpful discussions. We thank Esther Arkin, Will Evans, Matya Katz, and Petr Konecny for valuable input on the sensor coverage problems.

A. Efrat is partially supported by an NSF CAREER award (CCR-0348000) and an ITR/Collaborative Research grant (0312443). S. Har-Peled is partially supported by an NSF CAREER award (CCR-0132901). J. Mitchell is partially supported by grants from NASA Ames Research (NAG2-1620), the National Science Foundation (CCR-0098172, ACI-0328930), Metron Aviation, and the U.S.-Israel Binational Science Foundation (No. 2000160).

References

[1] Y. Hou, Y. Shi, J. Pan, A. Efrat, and S. Midkiff, "Maximizing lifetime of wireless sensor networks through single-session flow routing," The Bradley Dept. of ECE, Virginia Tech, Tech. Rep., 2003.

[2] J. M. Keil, "Polygon decomposition," in *Handbook of Computational Geometry*, J.-R. Sack and J. Urrutia, Eds. Amsterdam: Elsevier Science Publishers B.V. North-Holland, 2000, pp. 491–518.

[3] J. O'Rourke, *Art Gallery Theorems and Algorithms*, ser. The International Series of Monographs on Computer Science. New York, NY: Oxford University Press, 1987.

[4] T. C. Shermer, "Recent results in art galleries," *Proc. IEEE*, vol. 80, no. 9, pp. 1384–1399, Sept. 1992.

[5] J. Urrutia, "Art gallery and illumination problems," in *Handbook of Computational Geometry*, J.-R. Sack and J. Urrutia, Eds. North-Holland, 2000, pp. 973–1027.

[6] J. O'Rourke and K. J. Supowit, "Some NP-hard polygon decomposition problems," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 181–190, 1983.

[7] S. K. Ghosh, "Approximation algorithms for art gallery problems," in *Proc. Canadian Inform. Process. Soc. Congress*, 1987, pp. 429–434.

[8] A. Efrat and S. Har-Peled, "Locating guards in art galleries," in *2nd IFIP International Conference on Theoretical Computer Science*, 2002.

[9] H. González-Banos and J.-C. Latombe, "A randomized art-gallery algorithm for sensor placement," in *Proc. 17th Annual Symposium on Computational Geometry*, 2001, pp. 232–240.

[10] H. Brönnimann and M. T. Goodrich, "Almost optimal set covers in finite VC-dimension," *Discrete Comput. Geom.*, vol. 14, pp. 263–279, 1995.

[11] O. Cheong, A. Efrat, and S. Har-Peled, "On finding a guard that sees most and a shop that sells most," in *Proc. 15th ACM-SIAM Sympos. Discrete Algorithms*, 2004, pp. 1098–1107.

[12] J. Smith and W. Evans, "Triangle guarding," in *Proc. 15th Canad. Conf. Comput. Geom.*, 2003, pp. 76–80.

[13] W. Heinzelman, "Application-specific protocol architectures for wireless," Ph.D. dissertation, MIT, 2000.

[14] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, pp. 393–422, 2002.

[15] G. Pottie and W. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, pp. 51–58, 2000.

[16] T. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall PTR, 2001.

[17] N. Megiddo and A. Tamir, "On the complexity of locating linear facilities in the plane," *Oper. Res. Lett.*, vol. 1, pp. 194–197, 1982.

[18] L. Gewali, A. Meng, J. S. B. Mitchell, and S. Ntafos, "Path planning in $0/1/\infty$ weighted regions with applications," *ORSA J. Comput.*, vol. 2, no. 3, pp. 253–272, 1990.