Monitoring Schedules for Randomly Deployed Sensor Networks

Gruia Calinescu* Robert B. Ellis†

January 16, 2009

Abstract

Given n sensors and m targets, a monitoring schedule is a partition of the sensor set such that each part of the partition can monitor all targets. Monitoring schedules are used to maximize the time all targets are monitored when there is no possibility of replacing the batteries of the sensors. Each part of the partition is used for one unit of time, and thus the goal is to maximize the number of parts in the partition.

We present distributed algorithms for Monitoring Schedule under the following assumptions: 1) identical sensors can each monitor all targets within a certain radius, 2) the n sensors are randomly distributed uniformly in a large square containing the targets, 3) the number of sensors is high enough given the area the square, and 4) the communication range is twice the sensing range (thus any two sensors which can monitor the same target can communicate in one hop). Our results hold with high probability. With the further assumptions that the sensors are capable (for example, by GPS) of knowing their exact geographic position, and targets fill out the square, our schedule has at least $(1-\epsilon)opt$ parts, where *opt* is the optimum solution. Without geographic position we show that a previously proposed distributed algorithm can be modified to achieve a constant approximation ratio. Our algorithms run in a polylogarithmic number of communication rounds, with the exact running time depending on assumptions on the information a sensor receives when packets collide.

1 Introduction

The input to the Monitoring Schedule problem consists of n sensors, m targets, and a matrix describing which sensor can monitor which target. We call the *range* of sensor u the set of targets that u can monitor. In our model, an active sensor can monitor simultaneously all targets in its range. The output assigns every sensor a time-slot (an integer); we call this assignment a *schedule*. In a distributed environment in which the sensors have limited energy supply, the schedule dictates when a sensor actively monitors its range. More precisely, the sensor will sleep in all time-slots except the one assigned to it. The *quality* of the schedule is given by the maximum integer i such that for all $1 \le j \le i$, the set of sensors assigned time-slot j can monitor all targets. The objective is to maximize the quality of the schedule. In other words, we try to maximize the time all targets are covered, which is one appropriate measure [39] for the lifetime of the network. Several works [9, 32, 46] use "lifetime" to mean either the quality of such a schedule, or the quality of the best possible schedule for a given network.

The problem is motivated by the fact that, as opposed to traditional ad hoc networks, sensor networks are quite limited in power, computational capacity, and memory. A wireless sensor node typically consists

*Department of Computer Science, Illinois Institute of Technology, Chicago, IL, USA. Supported in part by NSF grant CCF-0515088; calinescu@iit.edu.

†Department of Applied Mathematics, Illinois Institute of Technology, Chicago, IL, USA; rellis@math.iit.edu.

of sensing hardware, an embedded processor and memory, a transceiver, and batteries. In most applications, replenishment of power resources is impossible. As noted in [20], the energy density of batteries has doubled only every 5 to 20 years, depending on the particular chemistry, and prolonged refinement of any chemistry yields diminishing returns. Thus energy efficiency will be critical for a long time.

This problem was considered by Cardei and Du [9], who present centralized heuristics. Since the sensors are left unattended after deployment, localized algorithms are preferable. In a localized algorithm, each sensor decides by itself its time-slot, after exchanging information with neighboring sensors. Two sensors are neighbors if they can communicate in one hop. Moscibroda and Wattenhoffer [32] and Calinescu [8] (see also [24]) adapt one Feige et al. algorithm [18] to give a randomized algorithm that, assuming no collisions, computes in one communication round a schedule whose quality is at least $1/O(\ln m)$ of the optimum quality. The results of Feige et al. [18] also immediately imply that, for arbitrary sensor-target matrices, obtaining a significantly better approximation ratio with a polynomial-time algorithm (even a centralized one) is very unlikely. The survey by Cardei and Jie [12] discusses, among others, this problem and cite more works on it, all heuristics without proven bounds: Slijepcevic and Potkonjak [37] and Cardei et al. [11]. Also, Cardei and Du [9] prove that, if the given sensor-target matrix is arbitrary, it is NP-Hard to determine if a schedule of quality 2 exists.

In practice, the given sensor-target matrix might not be arbitrary. An important special case is the *Euclidean model*, when sensors and targets are points in the two-dimensional plane, and the range of sensors is given by disks of the same radius. Whether a centralized algorithm can achieve significantly better than $O(\ln m)$ in approximating the optimum schedule in the Euclidean model is an open problem. In this paper, we concentrate on the *random Euclidean model*, which further assumes that the sensors are randomly deployed uniformly in an area slightly larger than the square containing the targets.

To our best knowledge, this is the first work to consider computing monitoring schedules in this model. If the sensors are capable (for example, by GPS) of knowing their exact geographic position, and the targets fill out a large square, we present a very simple distributed algorithm whose output has quality at least $(1-\epsilon)opt$, where opt is the quality of the optimum solution. Without geographic position we show that a previously proposed distributed algorithm can be modified to achieve a constant approximation ratio. Our algorithms run in a polylogarithmic number of communication rounds, with the exact running time depending on assumptions the information a sensor receives when packets collide. In the next subsection we discuss the assumptions under which our results hold, together with related work.

Other works with different definition of lifetime and random deployment of sensors include Maleki and Pedram [31], Xue and Ganz [40, 41], and Chen et. al. [15]. Heterogeneous sensors are considered by [29].

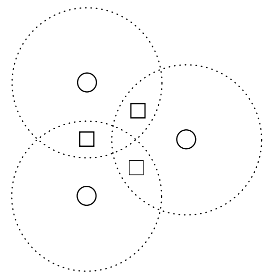
1.1 Related Results, Preliminaries, and Models

We start by describing the related Domatic Number problem and the existing results for it. Then we succinctly describe related work on randomly deployed sensor networks. We continue with the model of communication used by our algorithm, and explain why the number of communication rounds is the appropriate measure for localized algorithms in this setting, together with the quality of the produced schedule.

A set of vertices in a graph is a dominating set if every vertex outside the set has a neighbor in the set. The Domatic Number problem is that of partitioning the vertices of the graph into the maximum number of disjoint dominating sets.

Let n be the number of vertices in the graph and δ the minimum degree. As is standard, in this paper $o(1)$ means ϵ_n, with $\epsilon_n \to 0$ as $n \to \infty$; and whp (with high probability) means with probability converging to 1 as the indexing variable goes to infinity. Feige et al. [18] gave a polynomial-time algorithm to obtain a domatic partition with $(1-o(1))(\delta+1)/\ln n$ dominating sets. This implies a $(1+o(1))\ln n$-approximation algorithm for the Domatic Number, since the domatic number is always at most $\delta+1$. Moreover, they show that this is essentially best possible: for every $\epsilon > 0$, a $(1-\epsilon)\ln n$-approximation implies that $NP \subseteq DTIME(n^{O(\ln\ln n)})$.

Figure 1: Lifetime can differ from δ: sensors are small circles, targets are squares, and the big circles are the coverage area of each sensor.



Their algorithm as well as their hardness result carry over to computing monitoring schedules.

If we take δ to be the minimum over all targets of the number of sensors covering a given target, then δ is an obvious upper bound on the quality of any monitoring schedule. Even in the Euclidean model, achieving a schedule of quality δ may not always be possible, as shown in this simple example: three sensors having sensing radius 1 and located at the vertices of an equilateral triangle of side length $2 - \epsilon$, and three targets at the midpoints of the sides of the triangle – see Fig. 1. Each target is covered by two sensors, but no monitoring schedule of quality 2 exists. It is open as to whether any Euclidean instance has a schedule with quality at least a constant times δ [34].

In the random Euclidean model, Zhang and Hou [46] explore the issue of what density of sensors is needed for each target to be monitored by at least k sensors, finding bounds for δ. Further research in the same random model was done by Kumar et al. [28] and Wan and Yi [44]. Brass [6] defines "coverage" as he ratio of the area covered by a sensing disk to the total area and presents near-optimal results (see also Liu and Towsley [30]).

Another approach considered in the literature is to periodically rebalance the work of sensors [38, 5, 7, 16]. This method applies not just to monitoring, but also to clustering and communication [45, 22, 13, 23, 36, 45, 21]. Papers considering rebalancing typically have fewer sensors with more energy each. Assuming no changes in the monitoring situation, each such sensor can be replaced by a number, corresponding to the initial energy, of identical sensors, after which algorithms for computing the monitoring schedule can be used. This transformation may not preserve randomness, if present in the initial instance. If we ignore rebalancing costs and schedule in a centralized manner, a constant approximation ratio follows immediately from the results of [3] and [19], combined, for example, as in [5]. Monitoring schedule protocols have also been proposed in [43], Yan et al. [42] and recently in [10]; these papers present simulations without theoretical results.

In particular Berman et al. [5] use an algorithm called LBP (Load Balancing Protocol) which we adapt in our paper. LBP works as follows: in each rebalancing, each sensor starts without a status and repeatedly executes until its status becomes "on" or "off": 1) if sensor s has a target t covered only by s from the set of sensors without status or with status "on", then s changes its status to "on" and stays on until the next rebalancing, and 2) if sensor s detects that all its targets are covered by sensors with higher energy which are not "off", s changes its status to "off" and becomes idle until the next rebalancing. Similar schemes were proposed previously for clustering by Basagni [4] and Chatterjee et al. [14], and were called "generic" by Younis and Fahmy [45].

None of the localized rebalancing protocols or monitoring protocols mentioned above have proofs of both approximation ratio and running time (defined in the next paragraph). Also, to the best of our knowledge, there are no proven approximation guarantees for LBP nor non-trivial bounds on its running time.

Since we are considering localized approximation algorithms, we measure the performance of the algorithm by:

- The approximation ratio, which is the quality of the optimum schedule divided by the the quality of the output (so the approximation ratio is at least 1).
- The number of rounds required to prepare the schedule. This is a reasonable estimate for the overhead of the algorithm. As customary in the distributed algorithms setting [35], we call this number the running time of the algorithm.

We do not focus on the total number of packets sent for the computation of the schedule, as a good total number does not guarantee that certain sensors are not overloaded by sending a large number of packets and thus left without enough energy to monitor the slots assigned to them in the schedule. A small number of communication rounds, on the other hand, does guarantee a bound on the maximum energy spent computing the schedule.

Continuing with the assumptions, sensors are identical, starting with the same amount of energy and having communication radius twice the sensing radius. At the moment of deployment, sensors know the total number of sensors, and each sensor knows its own unique ID. Thus any two sensors which can monitor the same target can communicate in one hop. In fact, as we will see later, having larger communication range can be detrimental to our algorithms' running time due to more packet collisions. We also assume the communication rounds are synchronized.

A packet broadcast by a sensor is received by all active sensors within the sender's communication range. Depending on hardware, certain information is available to sensors when packets collide. In the most beneficial model, there are no collisions. In the hardest model, no sensor receives information on collisions. Various other models have been considered in the literature [1]; for example, the receiver could be aware of the collision while the sender is not.

To simplify matters, we assume all targets are in a large $B \times B$ square, and sensors have sensing radius 1 and are deployed uniformly and independently at random in a larger $(B + 4) \times (B + 4)$ square. The latter assumption allows us to ignore boundary effects. One can verify that all of our results, assuming no boundary effects, hold for all "fat" convex target areas (not necessarily a square). Here "fat" means that the ratio of the perimeter to the area is a at most $\epsilon/10$. For non-fat convex area, our algorithms have the slightly worse approximation ratio of $(2\pi\sqrt{3})/9 + \epsilon$ when sensors have GPS information, and again a large constant approximation ratio when they do not.

Define $q = \frac{2\sqrt{3}}{9}(B + 4)^2$ (the significance of $\frac{2\sqrt{3}}{9}$ comes from being the reciprocal of the area of a regular hexagon with side length 1). If direct placement of sensors is allowed, then clearly $\Theta(q)$ sensors are required to cover the whole $B \times B$ square containing the targets. Under uniform random placement, by recasting parameters for Theorem 2 of [44], $(9(2\pi\sqrt{3})^{-1} + o(1))q \ln q$ sensors are required to cover all but a vanishing fraction of the area with high probability. Throughout, we use q to give explicit bounds on the number of sensors n required by, and the running times of, our algorithms.

1.2 New Results

First, we consider the case when the sensors are capable (for example, by GPS) of knowing their exact geographic position, and the whole (large) square must be covered for each time-slot of the schedule. Let r be a positive integer. When $n \geq f(\epsilon)(r + 1)^2 q \ln[(r + 1)^2 q]$, with probability at least $1 - \frac{1}{q^2}$, our algorithm produces a schedule with quality at least $\frac{r^2}{(r+1)^2}(1 - \epsilon)opt$, where opt is the quality of the optimum solution and $f(\epsilon)$ is a function we make explicit later, but does not exceed $6(1/\epsilon)^2$. Thus the higher the density of sensors the better the approximation we obtain. With no collisions, our localized algorithm runs in constant time. In the hardest collision model, our algorithm requires each sensor to participate in $O(\ln^2 n)$ rounds of communication. This algorithm requires that nodes know (or have a good approximation of) n and B. The algorithm is very simple and is presented in Section 2. The same results hold if instead of requiring full area coverage the instance has at least $(16/\epsilon^2)q(\ln q)$ targets randomly distributed uniformly and independently in the square.

Second, we consider the case when the sensors do not have any geographic information. We make here the assumptions that the targets can be identified, in that the sensors are capable of assigning a locally unique ID to each target, meaning that (1) each target is assigned the same ID by all sensors covering it, and (2) no sensor covers two targets with the same ID. Also we assume the targets are distributed so that each sensor can broadcast the targets in its range in a polylogarithmic number of rounds (this is true whp if the number of targets is at most $q \cdot \text{polylog}(q)$, and they are randomly distributed). We must use a different algorithm here: Randomized Load-Balancing Protocol (RLBP), a randomized variation of the LBP algorithm mentioned above in which balancing occurs in one step. When $n \geq q \ln^2 q$, with probability at least $1 - \frac{1}{n^2}$, our algorithm produces a schedule with quality at least $c \cdot opt$, where c is a (small) universal constant. With no collisions, our localized algorithm requires each sensor to participate in $O(\ln^2 q)$ rounds of communication. In the hardest collision model, each sensor participates in $O(\ln^5 q)$ rounds of communication.

Although we can only prove a large constant approximation ratio, some experiments reported in [8] suggest a randomized version of the LBP algorithm may produce whp schedules close to δ in quality on random instances. We present and analyze the RLBP algorithm in Section 3, and make concluding remarks in Section 4.

2 Distributed algorithm with geographic information

We put on top of the $(B+4) \times (B+4)$ square a lattice of disjoint regular hexagons, which we call *cells*, of side length $1/(r+1)$, where r is a positive integer (See Fig. 2). We are only interested in *relevant* cells, defined as those fully contained in the bigger $(B+4) \times (B+4)$ square. Note that every target or point which need to be covered is inside a relevant cell, and in fact any point within distance 1 of the smaller $B \times B$ square is inside a relevant cell. We disregard the other cells, i.e., those not fully contained in the $(B+4) \times (B+4)$ square.

Since the area of a cell is $(9/(2\sqrt{3}))(1/(r+1)^2)$, the total number of relevant cells N is $N = B^2(r+1)^2 2\sqrt{3}/9 + O(B)$. As every relevant cell is contained in the $(B+4) \times (B+4)$ square, we deduce $N \leq (r+1)^2 q$; also note that $q = (1/(r+1)^2)N(1 + O(N^{-1/2}))$.

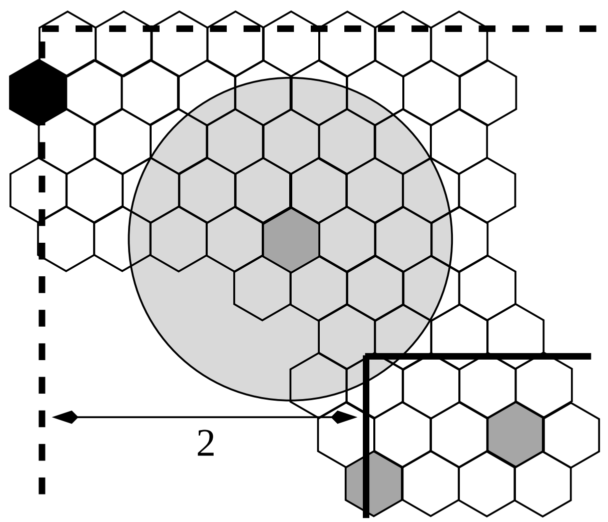
Each of the n sensors is placed uniformly and independently at random within the $(B+4) \times (B+4)$ square. With perfect information on geographic position, every sensor knows its own cell. First assume no collisions. Then in one communication round each sensor broadcasts its own ID and geographic position. The total number of packets sent is $O(n)$, each having $O(\log n)$ bits.

The centers of each cell sit on the intersection points of a small triangular lattice (with nearest-neighbor distance $\sqrt{3}/(r+1)$), and also on the intersection points of a large (r-times scaled up) triangular lattice or translate thereof (See Fig. 3). We define cells/centers lying on the intersection points of the same large lattice to be in the same *class*, and arbitrarily choose one center as the *class representative*. In Fig. 3, the thick circle and thick circle-with-"x" centers are in the same class with the thick circle as representative. Similarly, the thick square centers form a second class. Nearest-neighbor distance within a class of centers is $\sqrt{3}r/(r+1)$.

By considering translates by integer combinations of the basis vectors generating the large triangular lattice, we may take a full set of class representatives to lie in 2 abutting triangles as in Fig. 3. By an easy computation, there are r^2 classes. An ordering of these r^2 classes can be determined easily and computed by each sensor based on its location – the ordering method does not matter, so long as each sensor uses the same one.

The algorithm consists of the following: (1) each sensor uses GPS to determine its class and gathers the IDs and position of all the sensors in its cell; and (2) if the sensor is in class g, and it is the k^{th} sensor in

Figure 2: Here part of a regular hexagonal tiling is depicted close to the corner of a $(B+4)\times(B+4)$ square, bounded by the dashed line. The inner $B\times B$ square containing the targets is bounded by the solid line. For this lattice $r=4$, so that each hexagon has side length (and radius) $1/5$, and the distance between centers of abutting hexagons is $\sqrt{3}/5$. The three cells shaded gray are relevant – one completely outside the inner square, one partially intersecting the inner square, and the other completely inside the inner square. The gray circle of radius 1 indicates that a sensor having sensing radius 1 and lying in a relevant cell completely outside the inner square might still cover a target in the inner square. However, the black cell which is partially outside the outer $(B+4)\times(B+4)$ square is not relevant, and none of the sensors inside this non-relevant cell can cover a target.



its cell according to the ID, the sensor becomes active in time-slot $(1-\epsilon)(n/(q(r+1)^2))(g-1)+k$, where $\epsilon=o(1)$ is chosen (later in this Section) so that whp the minimum number of sensors in a relevant cell is at least $(1-\epsilon)n/(q(r+1)^2)$.

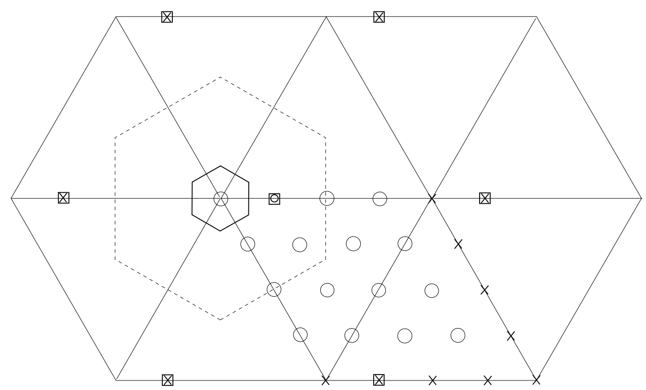
Letting $t=\lfloor r^2(1-\epsilon)n/(q(r+1)^2)\rfloor$, we have the following bound on schedule quality.

Claim 1 *Let $\epsilon>0$ and let $n\geq f(\epsilon)(r+1)^2 q\ln[q(r+1)^2]$, with $f(\epsilon)$ sufficiently large. Then with probability at least $1-o(1)$, each relevant cell contains at least $(1-\epsilon)n/(q(r+1)^2)$ sensors, and so for each time-slot from 1 to t, all points in the target region are covered by some active sensor.*

Proof. We defer the precise formula for $f(\epsilon)$ and for the probability bound until after the proof of Theorem 2. Given at least $(1-\epsilon)n/(q(r+1)^2)$ sensors per relevant cell, for any time-slot from 1 to t, we have one class (which we call *active* class) such that in every relevant cell from the class, a sensor is active. Because of the lattice arrangement of cells in a class, the hexagons of radius $r/(r+1)$ with centers in the active class completely cover (partition) the target area. The dotted hexagons centered at the thick circles of Fig. 3 are three members of such a class. The radius of a small cell is $1/(r+1)$. By the triangle inequality, the distance from some point x in the target area to any sensor in the nearest relevant cell from the active class is at most $r/(r+1)+1/(r+1)=1$, the sensing radius. ■

To obtain the $f(\epsilon)$ and the probability bound of the claim, we proceed as follows. Let $Y_i^{(j)}$ be the probability that sensor i lies in relevant cell j, where $1\leq i\leq n$ and $1\leq j\leq N$. For a fixed j, the $Y_i^{(j)}$'s are independent

Figure 3: Here again is depicted a partition of the space into "small" regular hexagons with sides length (and radius) $1/(r+1)$, with $r = 4$. Four abutting small hexagons are depicted here along with centers of many others. A sensor in a small hexagon covers any target inside the containing dotted hexagon of side length $r/(r+1)$ with the same center. There are r^2 small hexagons not containing an "x" in the two abutting triangles, called "class representatives". If a sensor is active in a class representative and each of its translates along the depicted triangular lattice with nearest neighbor distance $\sqrt{3}r/(r+1)$, then these active sensors cover the entire area.



random variables with $\Pr(Y_i^{(j)} = 1) = p$ and $\Pr(Y_i^{(j)} = 0) = 1 - p$, where $p := 3\sqrt{3}/(2(r+1)^2(B+4)^2) = 1/(q(r+1)^2)$. By considering the minimum number of sensors $\eta_{(1)} := \min_j \sum_i Y_i^{(j)}$ in a relevant cell, we obtain the following.

Theorem 2 *Let $n = xN \ln N$, where $x > 1$ is constant. Let $\beta(x)$ be the root of the equation*

$$\gamma + x(\ln\gamma - \gamma + 1) = 0 \tag{1}$$

which lies in $(1, \infty)$. Then with probability $(1 - o(1))$ as $N \to \infty$, the minimum number $\eta_{(1)}$ of sensors in a cell from placing n sensors independently and uniformly at random into N cells is at least $x \ln N/\beta(x)$.

The proof depends on the following distributional result (see, for example, [26, p.112]) on placing particles into cells. Following [26], define $\alpha = n/N$, and $p_k = \alpha^k e^{-\alpha}/k!$ for any nonnegative integer k (p_k is the probability for a Poisson random variable with intensity α equaling k).

Theorem 3 *If $\alpha/\ln N \to x > 1$ as $n, N \to \infty$ and $r' = r'(\alpha, N)$ is chosen so that $r' < \alpha$ and $Np_r \to \lambda$, where λ is a positive constant, then $\alpha/r' \to \beta(x)$ and*

$$\Pr(\eta_{(1)} \le r') \ge 1 - \exp\left(-\frac{\lambda\beta(x)}{\beta(x) - 1}\right), \tag{2}$$

where $\beta(x)$ is the root of (1) in the interval $1 < \beta(x) < \infty$.

We note that $\beta(x) = \frac{x}{1-x} W_{-1}\left(\frac{1-x}{ex}\right)$, where $W_{-1}(\cdot)$ is the -1 branch of the Lambert function, which is the inverse of $f(x) = xe^x$ (see, for example, [17, p.331]).
Proof. [Proof of Theorem 2] We show that by setting $r^* = x \ln N/\beta(x)$, Theorem 3 implies $\Pr(\eta_{(1)} \le r^*) \to 0$ (as $N \to \infty$). As a consequence, every cell has at least r^* sensors.

Following the proof of Lemma 3 of [26, p.99], for any $\lambda > 0$, if $r' = r'(\alpha, N)$ is chosen so that $Np_{r'} \to \lambda$ as $N \to \infty$, by using the Stirling approximation for $r'!$, taking the log of both sides, and multiplying through by $\alpha/(r' \ln N)$,

$$\frac{\alpha}{r'} + \frac{\alpha}{\ln N}\left(\ln\frac{\alpha}{r'} - \frac{\alpha}{r'} + 1\right) \;\to\; \frac{\beta(x)}{\ln N}\ln\lambda\,.$$

Similarly,

$$\frac{\alpha}{r^*} + \frac{\alpha}{\ln N}\left(\ln\frac{\alpha}{r^*} - \frac{\alpha}{r^*} + 1\right) \;\to\; \frac{\beta(x)}{\ln N}(-\ln(x\ln N))\,.$$

Both right-hand sides converge to 0 as $N \to \infty$. However, $\frac{\beta(x)}{\ln N}\ln\lambda > \frac{\beta(x)}{\ln N}(-\ln(x\ln N))$ for N sufficiently large, and $f(\gamma) = \gamma + \frac{\alpha}{\ln N}(\ln\gamma - \gamma + 1)$ is decreasing in the neighborhood of $\beta(\alpha/\ln N)$. Thus for N sufficiently large, $\alpha/r' < \alpha/r^*$, forcing $r' > r^*$ and $\Pr(\eta_{(1)} \le r^*) \le \Pr(\eta_{(1)} \le r')$. Since this is true for all $\lambda > 0$, by inspection of the right-hand side of (2), $\Pr(\eta_{(1)} \le r^*) \to 0$.

We obtain $f(\epsilon)$ by setting $x = f(\epsilon)$ and $N = q(r+1)^2$ in Theorem 2, so that $(1-\epsilon) = 1/\beta(f(\epsilon))$. Plugging this for γ into Equation 1 we get $1/(1-\epsilon) + f(\epsilon)(\ln(1/(1-\epsilon)) - 1/(1-\epsilon) + 1) = 0$. Using the Taylor expansion $\ln(1+y) \le y - y^2/2 + y^3/3$ which holds for all $y > 0$, and simplifying, we obtain that $f(\epsilon) = 6(1-\epsilon)^2/(3\epsilon^2 - 5\epsilon^3)$ suffices to have $(1-\epsilon) \le 1/\beta(f(\epsilon))$. If ϵ is small enough, $f(\epsilon) \le 3/\epsilon^2$. Although the probability convergence in Theorem 2 is only guaranteed to be $\Pr(\eta_{(1)} \le r^*) = o(1)$, we believe that a careful extension of the proofs of [26] would provide at least exponential convergence of the form $\Pr(\eta_{(1)} \le r^*) = O(\exp(-const \cdot N))$, since the limiting distribution of the number of sensors in a relevant cell is Poisson with intensity α. Alternatively, a straightforward application of the Chernoff bound of [2, Theorem A.1.13] yields convergence $\Pr(\eta_{(1)} \le r^*) \le 1/q^2$ for $f(\epsilon) = 6(1/\epsilon^2)$ and q sufficiently large.

A theorem of Kershner [25] shows that at least $(1+o(1))2\sqrt{3}B^2/9$ disks of radius 1 are needed to cover any convex shape of area B^2; this limit is asymptotically achieved by dividing the $B \times B$ square into hexagons (cells) of **radius** 1 (it is not surprising a perfect cellular placement would be optimal). Thus the optimum must use at least $(1-o(1))q$ sensors to cover the area (in this and next paragraph $o(1) = O(1/B)$), and therefore $opt \le (1+o(1))n/q$. Thus our algorithm has approximation ratio $\frac{(r+1)^2}{r^2}(1+\epsilon)$, as promised.

We have replaced requiring target coverage with requiring full area coverage. However, for $\epsilon > 0$, if the number of targets is $m = (1/\epsilon^2)q(\ln q + \omega(1))$, then with high probability coverage of targets placed uniformly and independently at random also requires $q(1-o(1))$ sensors. Indeed, given a radius ϵ hexagon tiling completely inside the $B \times B$ square, this m guarantees with high probability that each tile contains a target, and so expanding the radii of a disc cover of the targets by a factor of $(1+2\epsilon)$ gives full area coverage of a smaller $(B-4\epsilon) \times (B-4\epsilon)$ target area, after which Kershner's theorem can be applied.

Finally, we discuss the case when collisions are detected by neither sender nor receiver. We treat here r as a constant that does not depend on n, as r influences the approximation ratio. Or, if the density is very high, one could trade better approximation for larger time, or decrease if possible the transmission range to $2/(r+1)$, in effect eliminating r from the computation below. Only nodes in the same cell need to hear each other!

When collisions are detected by neither sender nor receiver, it is beneficial to partition the set of sensors into parts $S_1, S_2, \ldots, S_l$, all but S_l of size $f(\epsilon)(r+1)^2 q\ln[q(r+1)^2]$, and S_l having size between $f(\epsilon)(r+1)^2 q\ln[q(r+1)^2]$ and $2f(\epsilon)(r+1)^2 q\ln[q(r+1)^2]$, such a partitioning can be done based on sensor IDs prior to deployment and preserves the fact that each part consists of sensors placed uniformly at random. Moreover, for each part, we have a sharp estimate of the quality of the schedule our algorithm produces. We process the parts separately as follows: sensors from S_j sleep for $(j-1)r^2(1-\epsilon)f(\epsilon)\ln[q(r+1)^2]$ units of time, then communicate with each other and cover the area for the next $r^2(1-\epsilon)f(\epsilon)\ln[q(r+1)^2]$ units of time, after which, except for the case $j = l$, the sensors still alive from S_j stop communicating to allow the sensors from S_{j+1} to operate.

Now we concentrate on only one S_j. From Chernoff's bound ([2], Theorem A.1.11) we obtain that in each S_j, with probability at least $1 - \frac{1}{q^6}$, each sensor has at most $200(r+1)^2 f(\epsilon) \ln q$ other sensors within twice the communication range. Consider $200(r+1)^2 f(\epsilon) \ln q$ time-slots, and each sensor picking one time-slot at random to communicate its coordinates and ID. With constant probability, this packet will be received by all the destinations. Repeating this random process $\Theta(\ln q)$ times ensures that with high probability each sensor has one packet received by all sensors in its communication range. Thus each sensor sends $\Theta(\ln q)$ packets and is involved in $\Theta(\ln^2 q)$ rounds of communication, and with high probability manages to broadcast its ID and position. The total number of packets sent is $\Theta(n \ln n)$ and each packet has $O(\log n)$ bits.

3 Distributed algorithm without geographic information

We present the RLBP algorithm and prove that, with high probability, it has a constant approximation ratio, and each sensor participates in a polylogarithmic number of communication rounds.

We use $r = 1$ when constructing the hexagonal grid; thus the cells have sides of length $1/2$ and **diameter 1**. Note (this is also implicitly proved in the previous section) that every sensor in a cell covers the whole cell. Define $q' := 4q$.

Note that (as is implicitly proved in the previous section) the number of relevant cells is $N = q'(1 - O(q'^{-1/2}))$. We assume below that $n = q' \ln^2 q'$, with the case $n > q' \ln^2 q'$ treated by partitioning the sensors into parts of roughly $q' \ln^2 q'$ sensors each as done for our first algorithm. Since we only prove a large constant approximation ratio, we will assume without loss of generality whenever necessary that q' (and thus n) are large.

The RLBP algorithm works as follows. The first version assumes no collisions. Recall that each sensor is able to monitor one time unit. Initially all sensors are *alive*. At the beginning of a each time unit, all alive sensors wake up and execute the following:

1. Each sensor s produces a random number $rank(s)$ uniformly between 1 and n^5, and broadcasts it. This is the only communication during one time unit.

2. Each sensor s verifies, for each of its targets, if there is some sensor with smaller $rank$-value covering that target. If this holds for all its targets, sensor s goes to sleep and wakes up after one unit of time. If the condition does not hold (i.e., there is a target t such that s has the smallest $rank$-value among the sensors covering t), then sensor s monitors for one unit of time, after which it runs out of energy and dies.

For each target, the sensor with the smallest rank among those covering the target will not go to sleep, and thus the algorithm is correct as long as there are enough sensors alive to monitor all targets.

The size of each packet sent is $O(\log n)$ bits, and each node sends $O(\log^2 n)$ packets – at most one per time unit. The total number of packets sent is $O(n \log^2 n)$.

Both the instance and the algorithm are random. "Bad" events may happen - events which make the algorithm perform poorly. As shown below, the algorithm will be successful (that is, will cover all targets for time within a constant of the optimum time) if no bad event happens. We define what exactly are bad events later, and will prove that the probability of any bad event happening is small.

The average number of sensors per relevant cell (cells are defined in the previous section) is $n \cdot (1/q') = \ln^2 q'$. Note that the sensors in the algorithm are unaware of the cells; only the proof uses the cells extensively. The first bad event, A, is that there is a relevant cell with more than $1.01 \ln^2 q'$ or with less than $0.99 \ln^2 q'$ sensors.

Lemma 4 *For q' sufficiently large, $Pr[A] \leq 1/q'^3$*

Proof. The probability that sensor s is in relevant cell a is the same for all sensors and relevant cells: $p = 1/q'$. Define Y_a to be the (random) set of sensors in relevant cell a; $|Y_a|$ is a random variable. Then $E[|Y_a|] = n \cdot p = \ln^2 q'$. The Chernoff bound ([2], Corollary A.1.14) states that

$$Pr[||Y_a| - E[|Y_a|]| > \epsilon E[|Y_a|]] < 2e^{-c_\epsilon E[|Y_a|]}, \tag{3}$$

where $c_\epsilon > 0$ depends only on ϵ. In our case, $\epsilon = 0.01$. We can pick q' large enough to have $\ln q' > 10 \cdot \frac{1}{c_\epsilon}$ and therefore

$$Pr[||Y_a| - E[|Y_a|]| > \epsilon E[|Y_a|]] < 2e^{-10 \ln q'} < \frac{1}{q'^4}, \tag{4}$$

There are at most q' relevant cells a, and so the probability that there exists an a with Y_a out of the desired range is at most $1/q'^3$. ■

For i a nonnegative integer, define Y_a^i to be the (random) set of sensors which are in cell a and which have not been used for monitoring during time-slots $1, 2, \ldots, i$; here Y_a^0 is Y_a as defined in the proof of the previous lemma. Define $Y^i = \cup_a Y_a^i$. Define the random variable $\alpha_i = \min_a |Y_a^i|$, where the minimum is over relevant cells a; we are very interested in α_i as the quality of the schedule is at least i if $\alpha_i > 0$ (recall that every target is in a relevant cell).

Define X_a^i to be the (random) set of sensors from cell a that are used to monitor in time-slot i, for $i > 0$. Thus we have $Y_a^i = Y_a^{i-1} \setminus X_a^i$.

A second bad event, C, is that there are two sensors which pick the same rank in the same time unit. We have

$$Pr[C] \leq \ln^2 q' \cdot n \cdot n \cdot \frac{1}{n^5} \leq \frac{1}{n^2}, \tag{5}$$

since there are at most $\ln^2 q'$ interesting time units, n^2 pairs of sensors, and the probability two sensors pick the same rank is $1/n^5$. When C does not occur, the ranking of sensors gives a random permutation Π_i of the sensors, with s_1 before s_2 iff $rank(s_1) < rank(s_2)$; it is immediate that Π_i is uniformly random from the set of $n!$ permutations.

Call two cells *adjacent* if there is a sensor in each cell covering the same target; a cell is adjacent with itself if it contains a sensor covering a target. By elementary geometry a cell is adjacent to at most 21 cells (including itself).

Lemma 5 *Fix a sensor s and a relevant cell a, and assume $\beta > 0.51$ We have: $Pr[s \in X_a^{i+1} \mid s \in Y_a^i \wedge (\alpha_i > \beta \ln^2 q') \wedge \bar{C}] \leq \frac{21}{(1/2)\ln^2 q'}$.*

Proof. If $s \in X_a^{i+1}$, then there exists a target t in a nearby relevant cell a' for which s is the first sensor in Y^i covering t. The sensors $Y_{a'}^i \cup \{s\}$ certainly cover t. Conditioned on the event $s \in Y_a^i \wedge (\alpha_i > \beta \ln^2 q') \wedge \bar{C}$, the probability that s ranks first among $Y_{a'}^i \cup \{s\}$ is at most $|Y_{a'}^i|^{-1} \leq \alpha_i^{-1} \leq 2(\ln^2 q')^{-1}$. By subadditivity over the at most 21 nearby cells potentially containing targets that s could cover, the result follows. ■

For distinct cells a, a' adjacent after round i, define $Q_a^i(a')$ to be the set of sensors $s \in Y_a^i$ such that s ranks first in $Y_{a'}^i \cup \{s\}$. Define s_a^i to be the sensor in Y_a^i with lowest rank in Π_i. By step 2 of the algorithm and the definition of adjacency,

$$X_a^{i+1} \subseteq \{s_a^i\} \cup \bigcup_{a' \neq a, a' \text{ a relevant cell adjacent to } a} Q_a^i(a'). \tag{6}$$

We bound $|X_a^{i+1}|$ in terms of $|Q_a^i(a')|$ as follows.

Lemma 6 *Let a be a relevant cell, $i \geq 1$ a time-slot, $\beta > 0.51$, and d a nonnegative integer. Then*

$$Pr[|X_a^{i+1}| \geq d \mid \bar{C} \wedge \bar{A} \wedge (\alpha_i > \beta \ln^2 q')] < 21(2/3)^d. \tag{7}$$

Proof. If $d = 1$, the lemma is immediate. Thus we assume below that $d > 1$. Let a' be a relevant cell adjacent to but distinct from a. Let Π'_i be the restriction of the rank permutation Π_i to $Y_a^i \cup Y_{a'}^i$. The event $|Q_a^i(a')| \geq d$ occurs iff the first d elements of Π'_i all come from Y_a^i. The proportion of such permutations Π'_i on $Y_a^i \cup Y_{a'}^i$ is

$$\frac{|Y_a^i| \cdot (|Y_a^i| - 1) \cdots (|Y_a^i| - d + 1) \cdot (|Y_a^i| + |Y_{a'}^i| - d)!}{(|Y_a^i| + |Y_{a'}^i|)!} \leq \frac{|Y_a^i|^d}{(|Y_a^i| + |Y_{a'}^i|)^d}.$$

The conditioning on $\alpha_i > 0.51 \ln^2 q'$ implies that $|Y_{a'}^i| > 0.51 \ln^2 q'$, and on $\bar{A}$ implies that $|Y_a^i| \leq 1.01 \ln^2 q'$; thus $|Y_a^i|/(|Y_a^i| + |Y_{a'}^i|) < 2/3$ and so $Pr[|Q_a^i(a')| \geq d \mid \bar{C} \wedge \bar{A} \wedge (\alpha_i > \beta \ln^2 q')] < (2/3)^d$. For a', a'' relevant cells distinct from a, one of $Q_a^i(a')$, $Q_a^i(a'')$ is always a subset of the other; this is because $Q_a^i(a')$ is determined solely by the lowest rank sensor in $Y_{a'}^i$. Therefore if $X_a^{i+1} = d > 1$, then for some relevant cell a' adjacent to but distinct from a, $|Q_a^i(a')| \geq d$. The result follows by subadditivity over (6). ■

If $\beta > 0.51$, by a direct application of Lemma 6 we have that for all i and relevant cells a:

$$Pr[|X_a^{i+1}| \geq 18 \ln q' \mid \bar{C} \wedge \bar{A} \wedge (\alpha_i > \beta \ln^2 q')] \leq \frac{1}{q'^7}(1 + o(1)),$$

and thus

$$Pr[\alpha_{i+1} < \alpha_i - 18 \ln q' \mid \bar{C} \wedge \bar{A} \wedge (\alpha_i > \beta \ln^2 q')] \leq \frac{1}{q'^6}(1 + o(1)).$$

Using the equation above and letting $k = 0.01 \ln q'$ and $\gamma > 0.74$,

$$Pr[\alpha_{i+k} < \alpha_i - k \cdot 18 \ln q' \mid \bar{C} \wedge \bar{A} \wedge (\alpha_i > \gamma \ln^2 q')] \leq \frac{k}{q'^6}(1 + o(1)),$$

Fix one relevant cell a below, and one i. Assume $\gamma > 0.74$. Let nonnegative integers $d_1, d_2, \ldots, d_k$ satisfy $\sum_{j=1}^k d_j = d$. We have

$$\begin{aligned} Pr[|X_a{}^{i+1}| = d_1 \wedge |X_a^{i+2}| = d_2 \wedge \ldots \\ \wedge |X_a^{i+k}| = d_k \mid \bar{C} \wedge \bar{A} \wedge (\alpha_i > \gamma \ln^2 q')] \leq 21(2/3)^d, \end{aligned}$$

using Lemma 6, which we note holds conditioned on all the events $Y_{a'}^j$ for all cells a' and all $j \leq i$, and thus can be used to bound $Pr[|X_a^{i+2}| = d_2| \mid |X_a^{i+1}| = d_1]$, and so on. Here we used the fact that independent rank functions are used in each time unit.

There are $\binom{d+k-1}{k-1}$ ways of writing d as a sum of nonnegative $d_1, d_2, \ldots, d_k$, and we note that for $d > 20k$, $\binom{d+k}{k-1} < 1.1\binom{d+k-1}{k-1}$. Let $d = \delta k$, where δ is a (large) constant to be chosen later. Therefore for all i and all relevant cells a we have:

$$Pr[\sum_{j=1}^k |X_a^{i+j}| = \delta k \mid \bar{C} \wedge \bar{A} \wedge (\alpha_i > \gamma \ln^2 q')] \leq 21(2/3)^{\delta k} \frac{(\delta k + k)^k}{k!}.$$

Using the Stirling approximation, we obtain:

$$Pr[\sum_{j=1}^k |X_a^{i+j}| = \delta k \mid \bar{C} \wedge \bar{A} \wedge (\alpha_i > \gamma \ln^2 q')] \leq 21((2/3)^\delta (\delta + 1)e)^k.$$

As remarked above, $\binom{d+k}{k-1} < 1.1\binom{d+k-1}{k-1}$, and thus $21(2/3)^{d+1}\binom{d+k}{k-1} < 21(3/4)(2/3)^d\binom{d+k-1}{k-1}$. It follows that the numbers $21(2/3)^d\binom{d+k}{k-1}$ decrease in geometric progression for large enough d, and therefore

$$Pr[\sum_{j=1}^k |X_a^{i+j}| \geq \delta k \mid \bar{C} \wedge \bar{A} \wedge (\alpha_i > \gamma \ln^2 q')] \leq 4 \cdot 21((2/3)^\delta (\delta + 1)e)^k.$$

For a very large δ, taking into consideration $k = 0.01 \ln q'$, we can obtain for all relevant cells a:

$$Pr[\sum_{j=1}^{k} |X_a^{i+j}| \geq \delta k \mid \bar{C} \wedge \bar{A} \wedge (\alpha_i > \gamma \ln^2 q')] \leq \frac{1}{n^4},$$

and therefore

$$Pr[\alpha_{i+k} < \alpha_i - \delta k \mid \bar{C} \wedge \bar{A} \wedge (\alpha_i > \gamma \ln^2 q')] \leq \frac{1}{n^3}.$$

Start with $i = 0$ and repeat the argument above $r := \frac{1}{4k\delta} \ln^2 q'$ times to obtain that, with high probability, $\alpha_{kr} > 0.74 \ln^2 q'$, and therefore the number of time units used is at least kr. Recall that the optimum cannot exceed $(1 + o(1))4n/q' = (1 + o(1)) \ln^2 q'$. Therefore the approximation ratio is $(4 + \epsilon) \ln^2 q'/(kr) = 4(4 + \epsilon)\delta$.

Assume now that collisions are detected by neither sender nor receiver. Each sensor has $\Theta(\ln^2 q')$ sensors within twice its communication range. As described in the last paragraph of Section 2, $\Theta(\ln^3 q')$ time-slots are enough for a randomly repeated packet to be received whp by all its destinations: for $\Theta(\ln q')$ phases, a node randomly picks just one time-slot out of the $\Theta(\ln^2 q')$ time-slots of a phase. There are at most $\ln^2 q'$ time units of the algorithm above, and thus a sensor can be involved in at most $O(\ln^5 q')$ rounds of communication. The size of each packet sent is $O(\log n)$ bits and each node sends $O(\log^3 n)$ packets - $O(\log n)$ per time unit. The total number of packets sent is $O(n \log^3 n)$.

4 Conclusions and Open Problems

We showed that deploying a large number of sensors randomly is, in relation to certain lifetime issues, very close to the optimal deployment. Indeed, with high probability, our algorithm using geographic information produces a monitoring schedule with quality at least $(1 - o(1))$*superopt*, where *superopt* is the quality of the best monitoring schedule obtained by any placement of the same number of sensors. The first open question is to obtain similar results with a smaller number of sensors, e.g., $n \leq q \ln q$. Both algorithms presented can be used for values of n smaller than what we used, but we cannot analyze the quality of their output.

Regarding boundary effects, it is fairly easy to generalize our proofs to give a constant approximation if the sensors are only distributed in a convex target area (previously the sensors were distributed in a slightly larger area). For example, for a square target area and with n large enough, by using a single partition of the target $B \times B$ square into squares of diameter 1 and area 1/2 we get a quality of almost $n/(2B^2)$. A corner of the target area is covered by circa $\pi n/(4B^2)$ sensors and thus the approximation ratio is $(1 + o(1))\pi/2$. With a large number of sensors and a large square target area, one can use a clever partition (which we omit for lack of space) into little squares to get a quality within $(1 - \epsilon)$ of the degree of the corner – thus again a $(1 + \epsilon)$ approximation.

The methods we presented extend to the following cases:

- the range of a sensor is given by l_p metric rather than just l_2 (Euclidean distance),
- the region to be monitored, and where the sensors are randomly placed, is three dimensional.
- k-coverage is needed – each target must be covered by k sensors.

In regard to the communication range, our work suggests that having communication range exactly twice the sensing range is beneficial in that a localized algorithm exists, and larger communication range would only result in more collisions.

We made many assumptions to make our proofs work. RLBP needs a communication range at least twice the sensing range, but with GPS and enough sensors a small communication range is enough to compute the schedule – only nodes in the same cell need to communicate. When targets are mobile, the GPS-based algorithm is the same as it covers the whole area, while RLBP needs to acquire and communicate targets in each round, before deciding which node sleeps – and still runs the risk of a target leaving the covered area.

Both algorithms can be easily made to work without knowing n, the total number of sensors, in the case that there are no collisions. More research is needed to make algorithms that work without knowing n, with collisions, and for both scenarios, without synchronization.

Future work might assume only partial geographic information (e.g., angle of signal arrival [33, 27]) and obtain results similar to those presented here with full geographic information, as well as getting better running time with intermediate models of the information that is available to sensors when packets collide.

If the sensoring work is evenly distributed in the plane, as is the case with our algorithms in the case of randomly distributed sensors, algorithms for collecting the data in a balanced manner are straightforward. Note that our first algorithm, using geographic information, does create, with high probability given our assumptions, a connected network: there is a node in each cell and nodes in adjacent cells are one-hop neighbors. The RLBP algorithm, without geographic information, can be made to produce a connected network with high probability if we insist that each sensor is also a target. Doing so does not change anything in Section 3.

References

[1] I. F. Akyldiz, W. Su, Y. Sankarasubramanian, and E. Cairici. Wireless sensor networks: A survey. *Computer Networks*, 38:393–422, 2002.

[2] Noga Along and Joel H. Spencer. *The Probabilistic Method (second edition).* Wiley Interscience, 2000.

[3] Christoph Ambühl, Thomas Erlebach, Matus Mihalak, and Marc Nunkesser. Constant-factor approximation for minimum-weight (connected) dominating sets in unit disk graphs. In *APPROX 2006*, pages 3–14, 2006.

[4] S. Basagni. Distributed clustering algorithm for ad-hoc networks. In *International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN)*, 1999.

[5] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky. Power efficient monitoring management in sensor networks. In *Proc. IEEE Wireless Communications and Networking Conference*, 2004.

[6] Peter Brass. Bounds on coverage and target detection capabilities for models of networks of mobile sensors. *ACM Trans. Sen. Netw.*, 3(2):9, 2007.

[7] D.C. Brinza, G. Calinescu, S. Tongngam, and A. Zelikovsky. Energy-efficient continuous and event-driven monitoring. In *IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, pages 167–169, 2005.

[8] G. Calinescu. A fast localized algorithm for scheduling sensors. *Journal of Parallel and Distributed Computing*, 66:507–514, 2006.

[9] M. Cardei and D.-Z. Du. Improving wireless sensor network lifetime through power aware organization. *ACM Wireless Networks*, 11(3):333–340, 2005.

[10] M. Cardei, M.T. Thai, Y. Li, and W. Wu. Energy-efficient target coverage in wireless sensor networks. In *Proc. of IEEE Infocom*, 2005.

[11] Mihaela Cardei, David MacCallum, Maggie Xiaoyan Cheng, Manki Min, Xiaohua Jia, Deying Li, and Ding-Zhu Du. Wireless sensor networks with energy efficient organization. *Journal of Interconnection Networks*, 3(3-4):213–229, 2002.

[12] Mihaela Cardei and Jie Wu. Energy-efficient coverage problems in wireless ad-hoc sensor networks. *Computer Communications*, 29:413–420, 2006.

[13] A. Cerpa and D. Estrin. Ascent: Adaptive self-configuring sensor networks topologies. In *Proceedings of IEEE INFOCOM*, June 2002.

[14] M. Chatterjee, Das S.K, and D. Turgut. Wca: A weighted clustering algorithm for mobile ad hoc networks. *Cluster Computing*, pages 193–2004, 2002.

[15] Xi Chen, Hongxing Bai, Li Xia, and Yu-Chi Ho. Design of a randomly distributed sensor network for target detection. *Automatica*, 43(10):1713–1722, 2007.

[16] Adil I. Erzin, Vyacheslav V. Zalyubovskiy, and Hyunseung Choo. Maximizing lifetime for a sensor network. In *ICUIMC '08: Proceedings of the 2nd international conference on Ubiquitous information management and communication*, pages 436–440, New York, NY, USA, 2008. ACM.

[17] Corless et al. On the Lambert W function. *Adv. Computational Maths.*, 5:329–359, 1996.

[18] Uriel Feige, M. Halldorsson, G. Kortsarz, and A. Srinivasan. Approximating the Domatic Number. *SIAM J. Comput.*, 32(1):172–195, 2003.

[19] Naveen Garg and Jochen Konemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *IEEE Symposium on Foundations of Computer Science*, pages 300–309, 1998.

[20] R. Hahn and H. Reichl. Batteries and power supplies for wearable and ubiquitous computing. In *Proc. 3rd Int. Symp. Wearable Computers*, 1999.

[21] T. He, S. Krishnamurthy, J. Stankovic, T. Abdelzaher, L. Luo, T. Yan, L. Gu, J. Hui, and B. Krogh. Energy-efficient surveillance systems using wireless sensor networks. In *Mobisys 2004*, June 2004.

[22] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient routing protocols for wireless microsensor networks. In *Proc. Hawaii Int. Conf. on System Sciences, Jan.*, 2000.

[23] W. Heizelman, A. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1(4):660–670, 2002.

[24] Yan Jin, Ju-Yeon Jo, Ling Wang, Yoohwan Kim, and Xiaozong Yang. Eccra: An energy-efficient coverage and connectivity preserving routing algorithm under border effects in wireless sensor networks. *Comput. Commun.*, 31(10):2398–2407, 2008.

[25] R. Kershner. The number of circles covering a set. *American J. Mathematics*, 61:665–671, 1939.

[26] Valentin F. Kolchin, Boris A. Sevastyanov, and Vladimir P. Chistyakov. *Random allocations*. V. H. Winston & Sons, Washington, D.C., 1978. Translated from the Russian, Translation edited by A. V. Balakrishnan, Scripta Series in Mathematics.

[27] K. Krizman, T. Bieda, and T. Rappaport. Wireless position location: fundamentals, implementation strategies, and source of error. In *Veh. Tech. Conf*, pages 919–923, 1997.

[28] Santosh Kumar, Ten H. Lai, and József Balogh. On k-coverage in a mostly sleeping sensor network. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 144–158, New York, NY, USA, 2004. ACM.

[29] Jason H. Li and Miao Yu. Sensor coverage in wireless ad hoc sensor networks. *Int. J. Sen. Netw.*, 2(3/4):218–229, 2007.

[30] Benyuan Liu and D. Towsley. A study of the coverage of large-scale sensor networks. In *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pages 475–483, Oct. 2004.

[31] Morteza Maleki and Massoud Pedram. Qom and lifetime-constrained random deployment of sensor networks for minimum energy consumption. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 39, Piscataway, NJ, USA, 2005. IEEE Press.

[32] Thomas Moscibroda and Roger Wattenhofer. Maximizing the Lifetime of Dominating Sets. In *5th International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN), Denver, Colorado, USA*, 2005.

[33] A. Nasipuri and K. Lim. A directionality based location discovery scheme for wireless sensor networks. In *WSNA 2002*, 2002.

[34] J. Pach, 2006. Personal Communication.

[35] David Peleg. *Distributed Computing - A Locality-Sensitive Approach*. Society for Industrial and Applied Mathematics, 2000.

[36] M. Perillo and W. Heinzelman. Dapr: A protocol for wireless sensor networks utilizing an application-based routing cost. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, March 2004.

[37] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In *Communications, 2001. ICC 2001. IEEE International Conference on*, volume 2, pages 472–476 vol.2, 2001.

[38] D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proc. ACM WSNA02*, 2002.

[39] S. Tilak, N. Abu-Ghazaleh, and W. Heinzelman. A taxonomy of wireless micro-sensor network models. *ACM Mobile Computing and Communications Review (MC2R)*, 6(2), 2002.

[40] Qi Xue and A. Ganz. Maximizing sensor network lifetime: analysis and design guidelines. In *Military Communications Conference, 2004. MILCOM 2004. IEEE*, volume 2, pages 1144–1150 Vol. 2, Oct.-3 Nov. 2004.

[41] Qi Xue and Aura Ganz. On the lifetime of large scale sensor networks. *Computer Communications*, 29:502–510, 2005.

[42] T. Yan, T. He, and J. Stankovic. Differentiated surveillance for sensor networks. In *Proceedings of Sensys*, 2003.

[43] F. Ye, G. Zhong, S. Lu, and L. Zhang. Peas:a robust energy conserving protocol for long-lived sensor networks. In *Proc. of IEEE ICNP*, 2002.

[44] Chih-Wei Yi and Peng-Jun Wan. Coverage by randomly deployed wireless sensor networks. *IEEE Transactions on Information Theory*, 52:2658–2669, 2006.

[45] O. Younis and S. Fahmy. Distributed clustering in ad-hoc sensor networks: a hybrid energy-efficient approach. In *IEEE/INFOCOM*, 2004.

[46] Honghai Zhang and Jennifer C. Hou. On the upper bound of α-lifetime for large sensor networks. *ACM Trans. Sen. Netw.*, 1(2):272–300, 2005.