2-27-2012

Maximizing Network Lifetime on the Line with Adjustable Sensing Ranges

Amotz Bar-Noy
The Doctorate-Granting Institution of the City University of New York

Ben Baumer
The Doctorate-Granting Institution of the City University of New York, bbaumer@smith.edu

Maximizing Network Lifetime on the Line with Adjustable Sensing Ranges

Amotz Bar-Noy and Ben Baumer

The Graduate Center of the City University of New York
365 Fifth Avenue, New York, NY 10016.
amotz@sci.brooklyn.cuny.edu,bbaumer@gc.cuny.edu

Abstract. Given n sensors on a line, each of which is equipped with a unit battery charge and an adjustable sensing radius, what schedule will maximize the lifetime of a network that covers the entire line? Trivially, any reasonable algorithm is at least a $\frac{1}{2}$-approximation, but we prove tighter bounds for several natural algorithms. We focus on developing a linear time algorithm that maximizes the expected lifetime under a random uniform model of sensor distribution. We demonstrate one such algorithm that achieves an average-case approximation ratio of almost 0.9. Most of the algorithms that we consider come from a family based on `RoundRobin` coverage, in which sensors take turns covering predefined areas until their battery runs out.

Keywords: wireless sensor networks, adjustable range, restricted strip cover, lifetime, area coverage

1 Introduction

We consider the following disaster-relief scenario: Suppose you have a highway, supply line, or fence that you want to cover with a wireless sensor network (WSN) for as long as possible. Each sensor has a fixed location along the highway and a unit battery charge that drains in inverse proportion to its sensing radius, which you control. Given a deployment of sensors, what schedule will maximize the lifetime of the network? We analyze both the case where the sensors are placed by an adversary, and the case where they are deployed uniformly at random (e.g. - perhaps they have been dropped from an airplane).

Formally, let $U = [0, 1]$ be a line, and suppose that n sensors are deployed on U with locations $X = \{x_1, ..., x_n\}$. For any time $t \geq 0$, we associate with each sensor i a sensing radius $r_i(t) \in [0, 1]$ and a corresponding coverage interval $R_i(t) = [x_i - r_i(t), x_i + r_i(t)]$, and say that U is covered at time t if for every $x \in U$, there exists an $1 \leq i \leq n$ such that $x \in R_i(t)$. We impose the constraint that each sensor has a unit battery charge that drains at the rate $(r_i(t))^{1/\alpha}$ for some fixed $\alpha > 0$. Our goal is to construct a sensing schedule $S = \{r_i(t)\}_{i=1}^n$ that covers U for as long as possible, and call this value the *lifetime* of the network. That is, the lifetime T of a network is the largest time value t such that for every point $(x, t) \in U \times [0, T]$, there exists some sensor i such that $x \in R_i(t)$.

*Problem 1 (*ADJUSTABLE RANGE RESTRICTED STRIP COVER*).* Given a set of sensor locations X and a battery drainage rate α, compute a schedule $S = \{r_i(t)\}_{i=1}^n$, where $r_i(t)$ is the sensing radius of sensor i at time t, that maximizes T, subject to the constraints that for all pairs $(x,t) \in U \times [0,T]$, there exists an i such that $x \in R_i(t)$, and for all i, $\int_0^T (r_i(t))^{1/\alpha}\, dt \leq 1$.

In this paper, we provide both worst case (adversarial deployment) and average case (random deployment) analysis of several natural algorithms, for the particular situation in which $\alpha = 1$.

1.1 Previous Research

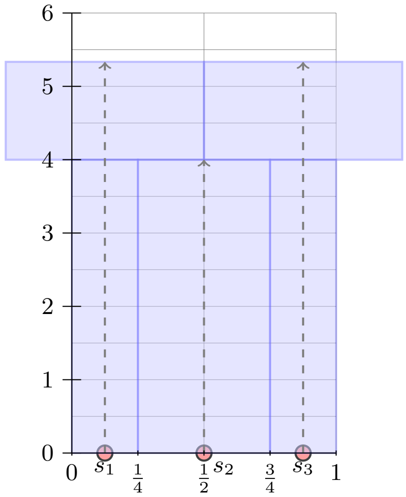
A closely related (and known NP-hard) problem is RESTRICTED STRIP COVER (RSC) [4], in which each sensor has a fixed sensing radius and a fixed duration indicating the length of time that it can be active. Our problem extends RSC by replacing the notion of duration with a that of a finite battery charge, and converting the sensing radius from a fixed input to a variable to be optimized. This introduces considerable complexity to the problem.

To see this, note that in RSC, each sensor can be represented in space-time by a single rectangle of fixed dimensions whose center has a fixed x-coordinate. The only variable to consider is the time (t-coordinate) at which the sensor becomes activated (e.g. - the rectangles can only be moved up and down). In our problem, the regions of space-time occupied by each sensor still have a fixed central x-coordinate and a fixed area, but the height and width may vary as a continuous function of time, so they are not even necessarily rectangles. Furthermore, in general we allow *pre-emptive scheduling*, meaning that a sensor can activate and deactivate more than once, splitting a region into multiple non-contiguous parts. In some cases, pre-emptive scheduling can increase the achievable lifetime. We show one such example in Figure 1.

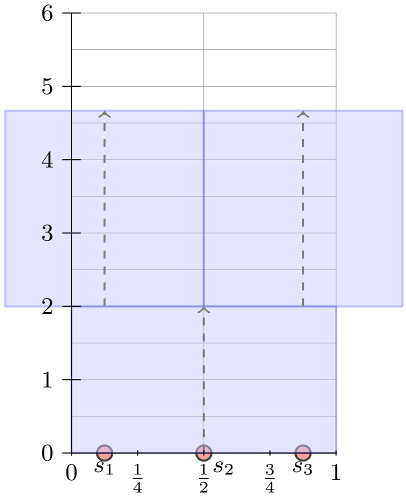
Buchsbaum et al. [4] proved the NP-hardness of RSC and gave an $O(\log\log\log n)$-approximation algorithm. Recently, a constant factor approximation algorithm for RSC was discovered by Gibson and Varadarajan [7].

Much of the related work on network lifetime has focused on *duty cycling*, wherein the goal is to maximize the number of covers k, rather than explicitly maximizing the network lifetime T. The notion of decomposability of multiple coverings can be found in Pach [10]. The connection to sensor networks was made more recently, but it has brought with it increased attention and results. Pach and Tóth [11] showed that a k-fold cover of translates of a centrally-symmetric open convex polygon can be decomposed into $\Omega(\sqrt{k})$ covers. This result was improved to the optimal $\Omega(k)$ covers by Aloupis et al. [1]. Gibson and Varadarajan [7] showed the same result without the centrally-symmetric restriction.

In the plane, Berman et al. [3] gave the first provably good $O(\log n)$-approximation algorithm for the Maximum Lifetime problem with fixed sensing ranges. Wu and Yang [12] initiated the study of area coverage with adjustable sensing ranges, and Cardei et al. [5] pursued a duty cycling approach involving set covers. Dhawan et al. [6] extended the work of [3] to the adjustable range setting.



(a) $T = 5\frac{1}{3}$, pre-emptive



(b) $T = 4\frac{2}{3}$, non-pre-emptive

Fig. 1. Illustration of the advantages of pre-emptive scheduling for $X = \{\frac{1}{8}, \frac{1}{2}, \frac{7}{8}\}$. The lifetime of the network is shown on the vertical axis, while location is shown on the horizontal axis. Each sensor is indicated by a red dot, and each rectangle represents a coverage assignment. The dashed arrows indicate periods of activity. Note that the total area of space-time consumed by each sensor is exactly 2.

In the one-dimensional setting, Peleg and Lev-Tov [8] found an optimal polynomial time solution to the one-time *target coverage* problem using dynamic programming. However, this question was about coverage efficiency, and not explicitly about network lifetime. The running time of the one-dimensional target coverage algorithm was later improved to $O(n + m)$, where m is the number of target points to be covered [2]. A PTAS is known for the area coverage version of the problem (again, for coverage efficiency, not lifetime), but no NP-hardness result is known. These results may offer optimal solutions for one moment in time, but do not necessarily lead to an optimal lifetime.

1.2 Our Contribution

Our extension of Restricted Strip Cover is the first to consider the true lifetime for area coverage on the line with adjustable sensing ranges. For the special case where $\alpha = 1$, any reasonable algorithm is at least a $\frac{1}{2}$-approximation, but we prove tigher bounds for several natural algorithms. However, since a constant factor approximation is trivial, most of our efforts are focused on raising the approximation ratio in the *average case*, which in an application scenario, is likely to be of greater value. Our main result is a constructive proof that a linear time algorithm exists that achieves an approximation ratio of nearly 0.9 in the average case. We accomplish this by employing `RoundRobin` coverage on a hierarchical system of pre-defined coverage areas. Although we allow pre-emptive scheduling, we do not explicitly use it in our algorithms. Thus, our results are also valid for the special case in which pre-emptive scheduling is not allowed. A summary of our results is shown in Table 1.

Algorithm	$\mathbb{E}[T]$	$Var[T]$	AC	WC
`RoundRobin`	1.386	0.078	0.693	2/3
k-`RoundRobin`	1.386	0.078	0.693	2/3
$\log_2$-`RoundRobin`	1.738	0.022	0.869	2/3
Optimized $\log_2$-`RoundRobin`	1.791		0.896	2/3

Table 1. Summary of lifetime results for `RoundRobin` algorithms. T is a random variable describing the per sensor lifetime under uniform random deployment. AC and WC show lower bounds for the average-case and upper bounds for the worst-case approximation ratios, respectively.

2 Preliminaries

For any set of sensor locations X, we assume that there exists some optimal schedule $S = \{r_i(t)\}_{i=1}^n$ that will produce the longest possible lifetime T_{OPT}. As the battery charges are finite, we can bound this value.

Proposition 1. *If n sensors are deployed, then $n \leq T_{OPT} \leq 2n$.*

Proof. The lower bound is immediate since *any* reasonable algorithm achieves $T \geq n$. Consider the case where all of the sensors were located at 0; each could cover U for exactly 1 time unit.

For any time t, each sensor i covers a subinterval of U of width $2r_i(t)$. The total energy consumed is given by $\int_0^\infty r_i(t)\, dt$, which is at most 1 since the battery has unit capacity. Thus, if V_i is the region of space-time consumed by the sensor i, then $|V_i| = \int_0^\infty 2r_i(t)\, dt \leq 2$. The total area of space-time consumed then satisfies

$$\left| \bigcup_{i=1}^n V_i \right| \leq \sum_{i=1}^n |V_i| \leq 2n\,.$$

It is easy to see in this geometric setting that the goal of maximizing T is equivalent to the goal of minimizing coverage overlap (i.e. - intersections $V_i \cap V_j$), and any extraneous coverage outside of U.

In some cases, we can bound T_{OPT} away from $2n$. For any subset $Y = \{x_1, ..., x_m\} \subseteq X$, let $f(Y) = -\frac{1}{2} + \sum_{j=1}^m (-1)^{m-j} x_j$. We show (see the Appendix for a full proof) that if $f(Y) = 0$, then the sensors in Y have the proper spacing to create a *pinned disk* coverage assignment, which has no wasted coverage.

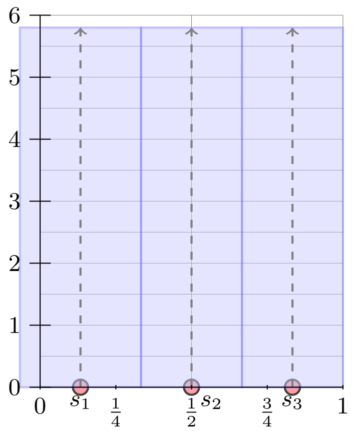
Proposition 2. *A radial assignment that gives perfect coverage over $[0, 1]$ at time t exists if and only if there is a subset $Y \subseteq X$ such that $f(Y) = 0$.*

Corollary 1. *If no subset $Y \subseteq X$ satisfies $f(Y) = 0$, then $T_{OPT}(X) < 2n$.*

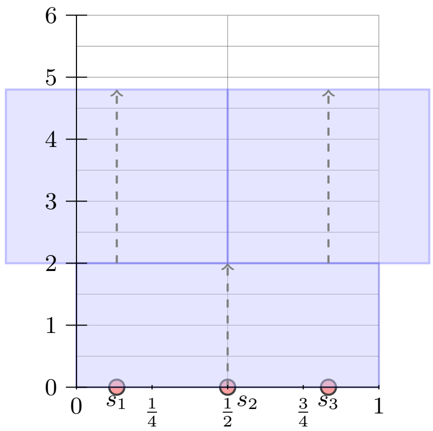
Our work in this paper is focused on `RoundRobin` algorithms, but we show a worst-case approximation bound for `Greedy`, which iteratively schedules the least-wasteful assignment of radii until a sensor runs out of battery life.

Observation 1 *The approximation ratio of* `Greedy` *is at most* $\frac{5}{6}$.

Proof. Consider $X = \{\frac{1}{6} - \epsilon, \frac{1}{2}, \frac{5}{6}\}$, for some $\epsilon > 0$. `Greedy` chooses to activate the middle sensor by itself on U first, since that is the only perfect assigment possible. This produces a T approaching 5 as $\epsilon \to 0$, but $T_{OPT} = 6$ is achievable in the limit (see Figure 2).



(a) $T_{OPT} \to 6$



(b) $T_{Greedy} \to 5$

Fig. 2. Proof that Greedy is at best a $\frac{5}{6}$-approximation. Both diagrams show what happens as $\epsilon \to 0$.

3 Analysis of `RoundRobin` Algorithms

Let $\bar{T} = T/n \in [0, 2]$ be the average network lifetime per sensor. For a group of sensors working simultaneously, it is often convenient to discuss the *normalized* lifetime $\hat{T}$, which is scaled so that $\hat{T} \in [0, 2]$. [1]

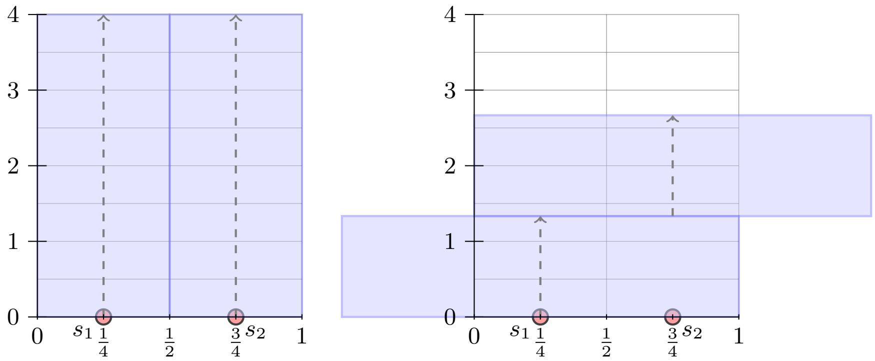
3.1 `RoundRobin`

In its simplest incarnation, `RoundRobin` simply forces each sensor to successively cover all of U for as long as possible. That is, each sensor i is assigned a radius of $r_i = \max(x_i, 1 - x_i)$, and is pushed onto a single queue. It is easy to show that this algorithm is at best a $\frac{2}{3}$-approximation of T_{OPT}.

Lemma 1. `RoundRobin` *is at best a* $\frac{2}{3}$*-approximation.*

[1] This distinction will be made clear in Section 3.2.

Proof. Consider $X = \{\frac{1}{4}, \frac{3}{4}\}$. The only two sensible assignments are shown in Figure 3. But while $T_{OPT} = 4$, `RoundRobin` achieves a lifetime of only $2\frac{2}{3}$.



(a) $T_{OPT} = 4$ (b) $T_{RR} = 2\frac{2}{3}$

Fig. 3. Proof that `RoundRobin` is at best a $\frac{2}{3}$-approximation.

A more complicated argument (presented in the Appendix) shows that `RoundRobin` is at least a 0.548-approximation of T_{OPT}.

Clearly, `RoundRobin` performs best when sensors are located close to 1/2, where the lifetime is close to 2, and poorly for sensors near 0 and 1, where the lifetime is 1. We analyze the average case by assuming that X is a uniform random variable over $[0, 1]$. Then the function $T_{0,1}(X) = \frac{1}{\max(X,1-X)}$ yields a new r.v. giving the lifetime of an *individual sensor*. It is easy to calculate its mean

$$\mu_T \triangleq \mathbb{E}[T_{0,1}(X)] = \int_0^1 \frac{dx}{\max(x, 1-x)} = 2\int_{\frac{1}{2}}^1 \frac{dx}{x} = 2\ln x\Big|_{\frac{1}{2}}^1 = 2\ln 2\,, \tag{1}$$

and variance

$$\sigma_T^2 \triangleq \mathbb{E}[T_{0,1}^2(X)] - \mu_T^2 = \int_0^1 \frac{dx}{(\max(x, 1-x))^2} - \mu_T^2 = 2 - 4\ln^2 2\,. \tag{2}$$

We will develop algorithms that improve on this expected lifetime of μ_T.

Central Limit Theorem. Of course, with n sensors, we are more interested in the distribution of $\bar{T}$, as opposed to that of T. Since we know μ_T and σ_T^2, the Central Limit Theorem implies that the distribution of $\bar{T}$ approaches a normal distribution with mean μ_T and variance σ_T^2/n as $n \to \infty$. For this reason we report the variance but focus most of our attention on the expected average lifetime of each algorithm.

Theorem 1. *The approximation ratio of* `RoundRobin` *is between 0.548 and 2/3, but it achieves at least a 0.693-approximation ratio in the average case.*

3.2 k-`RoundRobin`

A natural extension of `RoundRobin` is to partition U into k equally-spaced subintervals, and run it independently on each of those. Somewhat surprisingly, the performance is no better in either the worst or the average case.

Let k be a fixed positive integer, and let $U_k(i) = [\frac{i-1}{k}, \frac{i}{k}]$ for $i = 1, ..., k$ define a partition of U. We define k-`RoundRobin` to be the algorithm that runs `RoundRobin` independently on each subinterval $U_k(i)$; maintaining k parallel queues. However, over any subinterval $[a, b] \subseteq U$, the r.v. giving the lifetime of a sensor in $U_k(i)$ is simply a rescaling of T from the original `RoundRobin`.

Remark 1. For any interval $[a, b] \subseteq U$, the expected lifetime $T_{a,b}(X)$ of a sensor running `RoundRobin` on $[a, b]$ is $\frac{\mu_T}{b-a}$ with variance $(\frac{\sigma_T}{b-a})^2$.

With $b - a = 1/k$, the expected lifetime of each sensor in k-`RoundRobin` is $\mathbb{E}[T] = k\mu_T$, with a maximum lifetime of $2k$. However, in order to cover the whole line, we have to run k parallel queues, so that the expected *normalized* lifetime of each sensor is $\mathbb{E}[\hat{T}] = \mu_T$. For a set of n sensors, the total expected lifetime is $n\mu_T$, so the expected *average* network lifetime $\mathbb{E}[\bar{T}]$ is μ_T. Similar calculations show that the variance of each sensor's lifetime is $(k\sigma_T)^2$, while the normalized variance is σ_T^2 and the variance of the mean is $Var(\bar{T}) = \sigma_T^2/n$.

Load Balancing. Since we are maintaining k parallel queues that must work together to cover U, our calculations are sensitive to the requirement that the lifetime be the same in each queue.

Following [9], we can think of the observation of each sensor location as an independent Poisson trial, and use a Chernoff bound to ensure that the probability of a sub-interval $U_k(i)$ getting too few sensors is $o(1)$. Let N_i be a r.v. denoting the number of sensors in $U_k(i)$. Then for any $k < \frac{n}{3 \ln n}$, we have that

$$\Pr\left[\left|N_i - \frac{n}{k}\right| \geq \sqrt{\frac{3n \ln n}{k}}\right] \leq 2 \exp\left\{-\frac{1}{3}\frac{n}{k}\frac{3k \ln n}{n}\right\} = \frac{2}{n}\,.$$

In our case, we need to bound the probability that *some* $U_k(i)$ has too few sensors in it, but using a union bound, the probability of this is at most $\frac{2k}{n}$, which still goes to 0 as $n \to \infty$ for a fixed k. This shows that with high probability, the deviations from the mean number of sensors in each interval are on the order of $O(\sqrt{n \ln n})$ for a fixed k.

Set $n = n_1 + n_2$, where $n_1 = k \cdot \min_{1 \leq i \leq k} N_i$. Our scheduler allows the n_1 sensors to run k-`RoundRobin` on perfectly balanced stacks, and then throws the n_2 leftover sensors away. Thus, the actual expected average lifetime of the algorithm is

$$\mathbb{E}[\bar{T}_{actual}] = \frac{n_1}{n} \cdot \mathbb{E}[\bar{T}] + \frac{n_2}{n} \cdot 0 \to \mathbb{E}[\bar{T}] = \mu_T\,, \text{ as } n \to \infty\,,$$

since $n_2 = O(\sqrt{n \ln n})$ and thus $\frac{n_2}{n} \to 0$ as $n \to \infty$.

Observation 2 *k-`RoundRobin` provides the same worst-case and average-case performance as `RoundRobin`.*

3.3 $\log_2$-RoundRobin

Nevertheless, clever applications of RoundRobin can yield efficient algorithms. While the expected lifetime of a sensor in RoundRobin is independent of the length of the interval it covers, it still performs better when it is near the center of the interval. Specifically, the expected lifetime of a sensor covering an interval $[a, b]$, that is located within a subinterval $U_{a,b}(c) = [\frac{b+a}{2} - c, \frac{b+a}{2} + c] \subseteq [a, b]$ for some $c \in [0, \frac{b-a}{2}]$, is given by

$$\mathbb{E}[T_{a,b}(X; c)] = \frac{1}{2c} \int_{\frac{b+a}{2}-c}^{\frac{b+a}{2}+c} \frac{dx}{\max(x-a, b-x)} = \frac{1}{c} \ln\left(1 + \frac{2c}{b-a}\right). \qquad (3)$$

Since the maximum lifetime is $2/(b-a)$, the expected normalized lifetime is $\mathbb{E}[\hat{T}_{a,b}(X; c)] = \frac{b-a}{c} \ln\left(1 + \frac{2c}{b-a}\right)$, and the normalized variance is:

$$Var(\hat{T}_{a,b}(X; c)) = 4\left[1 - \frac{1}{1 + \frac{b-a}{2c}} - \left(\frac{b-a}{2c} \cdot \ln\left(1 + \frac{2c}{b-a}\right)\right)^2\right]. \qquad (4)$$

Within the framework of using RoundRobin on subintervals $[a, b]$, but selecting only those sensors that are closest to the midpoints of those intervals, an algorithm emerges naturally: partition U into subintervals, but employ RoundRobin only on those sensors that are close to the midpoint of each subinterval. To make efficient use of each sensor, we construct a hierarchical series of such partitions. We call this algorithm log_2-RoundRobin, and it is indexed by a depth parameter k, which indicates the number of partitions it employs.

Formally, for a fixed positive integer k, we partition U into 2^k+1 subintervals $U_k(i) = [\frac{i}{2^k} - \frac{1}{2^{k+1}}, \frac{i}{2^k} + \frac{1}{2^{k+1}}] \cap U$ for $i = 0, 1, ..., 2^k$. [2] If sensor $x \in U_k(i)$, then x is responsible for covering the interval around $i/2^k$ with radius $\frac{gcd(i,2^k)}{2^k}$. For example, any sensor that lies within 2^{-k-1} of $\frac{1}{2}$ is assigned to cover all of U. Similarly, sensors within 2^{-k-1} of either $\frac{1}{4}$ or $\frac{3}{4}$ are assigned to cover the subintervals $[0, \frac{1}{2}]$ and $[\frac{1}{2}, 1]$, respectively. A graphical depiction of the normalized sensor network lifetime as a function of location in shown in Figure 4.

For $j = 1, ..., k$, we define $\Gamma_k(j)$ to the be the set of intervals that comprise the j^{th} level of the algorithm. Formally, we denote

$$\Gamma_k(j) = \left\{ \bigcup_{i=1}^{2^k-1} U_k(i) : \log_2(gcd(i, 2^k)) = k - j \right\}.$$

Note that $\Gamma_k(j)$ consists of 2^{j-1} disjoint intervals, each of width 2^{-k}.[3] Thus $\Gamma_k(j)$ occupies 2^{j-k-1} of U. We can compute the expected normalized lifetime

[2] Note that the first and last intervals, $U_k(0) = [0, 2^{-k-1}]$ and $U_k(2^k) = [1-2^{-k-1}, 1]$, respectively, are only half as wide as the others, all of which have width 2^{-k}.

[3] We let $\Gamma_k(0)$ be the set of sensors assigned to $U_k(0)$ or $U_k(2^k)$, and have those cover their respective half-intervals. Their contribution to the network lifetime becomes negligible as $k \to \infty$, so we omit it from our calculations.

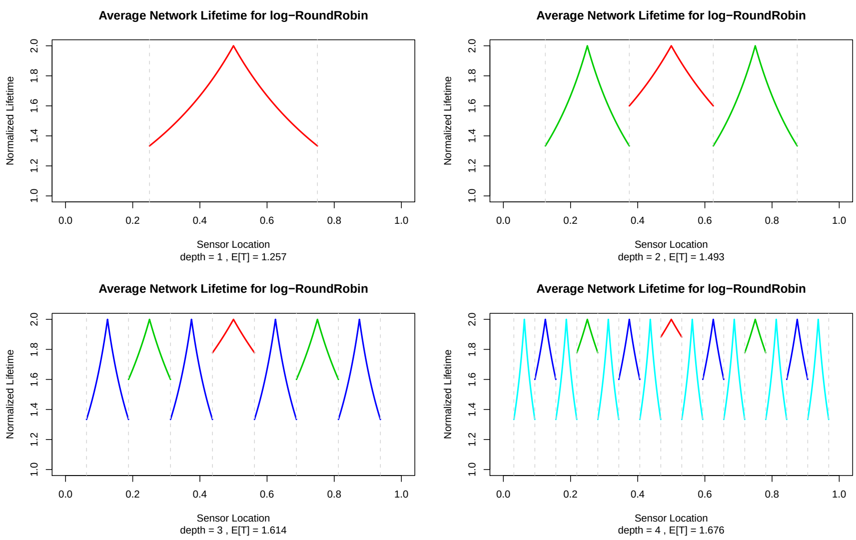


Fig. 4. Normalized Sensor Network Lifetime for $k = 1, 2, 3, 4$ using the log_2-`RoundRobin` algorithm. Each color represents the lifetime of the sensors in $\Gamma_k(j)$. Note that while the actual lifetime of a sensor in $\Gamma_k(j)$ may reach 2^j, it must run in parallel with 2^{j-1} partners, so the normalized lifetime *of the group* is at most 2. The expected average lifetime of the network approaches 1.737752 as $k \to \infty$.

for $\Gamma_k(j)$ using Equation 3

$$\mathbb{E}[\hat{T}_k(j)] = \mathbb{E}[\hat{T}_{0,2^{-j+1}}(X; 2^{-k-1})] = 2^{k-j+2} \ln\left(1 + 2^{j-k-1}\right),$$

and the variance using Equation 4:

$$Var(\hat{T}_k(j)) = 4\left[1 - \frac{1}{1 + 2^{k-j+1}} - \left(2^{k-j+1} \cdot \ln\left(1 + 2^{j-k-1}\right)\right)^2\right].$$

Summing over the $\Gamma_k(j)$'s to find the total expected normalized lifetime, we obtain

$$\mathbb{E}[\hat{T}_k] = \sum_{j=1}^{k} \frac{\mathbb{E}[\hat{T}_k(j)]}{2^{k-j+1}} = 2\ln\prod_{j=1}^{k}\left(1 + 2^{j-k-1}\right) = 2\ln\prod_{\ell=1}^{k}\left(1 + 2^{-\ell}\right). \qquad (5)$$

The analogous infinite product is a q-series [13], denoted here by $\left(-1; \frac{1}{2}\right)_\infty$, for which we can compute an approximate limiting value. This leads directly to the expected average lifetime:

$$\mu_T^* \triangleq \mathbb{E}[\hat{T}] = \lim_{k\to\infty} \mathbb{E}[\hat{T}_k] = 2\ln\left(\prod_{\ell=1}^{\infty} 1 + 2^{-\ell}\right) \approx 1.737752.$$

The mean normalized variance satisfies

$$\mathbb{E}[Var(\hat{T}_k)] = \sum_{j=1}^{k} \frac{Var(\hat{T}_k(j))}{2^{k-j+1}} = 4\left[\sum_{\ell=1}^{k} \frac{1}{1+2^\ell} - 2^\ell \cdot \ln^2\left(1+2^{-\ell}\right)\right],$$

which has the approximate limit of 0.02202547 as $k \to \infty$. Computation of the total variance is omitted, but it will converge to the above as $k \to \infty$.

Furthermore, it is clear from Figure 4 that the worst-case lifetime occurs when a sensor in $\Gamma_k(k)$ lies near one of the endpoints of the interval on which it is active. The normalized lifetime at this point is 4/3, a constant. This provides the same worst-case performance as `RoundRobin`.

Load Balancing, revisited. In $\log_2$-`RoundRobin`, each set $\Gamma_k(j)$ for $j = 1, ..., k$ maintains 2^{j-1} parallel queues. Proper functioning of our algorithm requires balanced loads across these queues, but the hierarchical structure of $\log_2$-`RoundRobin` alleviates the load balancing issue if the $\Gamma_k(j)$'s are pushed onto a central stack in ascending order of j. To see this, suppse that the left half of $\Gamma_k(2)$ runs out, while the right half is still going. U remains covered if the left half of $\Gamma_k(3)$ starts running alongside the right half of $\Gamma_k(2)$. In this manner load imbalances are averaged out over the k levels of the algorithm.

Nevertheless, a Chernoff bound analogous to the one used above for k-`RoundRobin` will show that for $k < \ln n$, with high probability N_i will deviate from its mean of $\frac{n}{2^k}$ by $O(\sqrt{n \ln n})$. Setting $n_1 = 2^k \cdot \min_{1 \le i \le 2^k - 1} N_i$ yields

$$\mathbb{E}[\bar{T}_{actual}] \ge \frac{n_1}{n} \cdot \mu_T^* + \frac{n_2}{n} \cdot 0 \to \mu_T^*, \text{ as } n \to \infty.[4]$$

Theorem 2. *The $\log_2$-`RoundRobin` algorithm is at best a $\frac{2}{3}$-approximation of T_{OPT}, but for sufficiently large n, achieves an average-case 0.869-approximation ratio with high probability.*

3.4 Optimizations

Still, it is clear from Figure 4 that efficiency is highest in $\Gamma_k(1)$ and lowest in $\Gamma_k(k)$. We can show that in fact, the relative efficiency of $\Gamma_k(k)$ is the constant $2 \ln \frac{3}{2} \approx 0.81$. On the other hand, it is easy to see that the relative efficiency of $\Gamma_k(1)$ approaches 1 as $k \to \infty$. Therefore, we can improve the efficiency of $\log_2$-`RoundRobin` by shrinking the intervals over which $\Gamma_k(k)$ is active. Note that since every $\Gamma_k(j)$ for $j = 1, ..., k-1$ borders $\Gamma_k(k)$ on both sides, we maintain balanced loads across each $\Gamma_k(j)$ even as we shrink the width of $\Gamma_k(k)$. Let $\epsilon(k) \in [0, 1]$ be a parameter measuring the inward shift of the boundaries of $\Gamma_k(k)$. Then using Equation 3, the expected normalized lifetime becomes

$$\mathbb{E}[\hat{T}_k(j, \epsilon)] = \mathbb{E}\left[\hat{T}_{0,2^{-j+1}}\left(X; \frac{1+\epsilon}{2^{k+1}}\right)\right] = \frac{2^{k-j+2}}{1+\epsilon} \ln\left(1 + (1+\epsilon)2^{j-k-1}\right)$$

[4] The inequality is justified by the preceding argument that in practice, the actual load balancing will work at least this well.

for $j = 1, ..., k-1$, and

$$\mathbb{E}[\hat{T}_k(k,\epsilon)] = \mathbb{E}\left[\hat{T}_{0,2^{-k+1}}\left(X; \frac{1-\epsilon}{2^{k+1}}\right)\right] = \frac{4}{1-\epsilon}\ln\left(\frac{3-\epsilon}{2}\right).$$

Taking the weighted average again, we have a generalization of Equation 5 that can be expressed as another q-series:

$$\mathbb{E}[\hat{T}_k(k,\epsilon)] = 2\ln\left(\frac{3-\epsilon}{2}\right)\prod_{i=2}^{\infty} 1 + (1+\epsilon)2^{-i} = 2\ln\frac{(3-\epsilon)\left(-(1+\epsilon); \frac{1}{2}\right)_\infty}{(\epsilon+3)(\epsilon+2)}.$$

We can find the optimal $\epsilon(k)$ using elementary calculus, but unfortunately a general solution requires factoring a polynomial of degree $k-1$:

$$T_k'(\epsilon) = 0 \Rightarrow \frac{1}{3-\epsilon} = \sum_{j=1}^{k-1} \frac{1}{2^{j+1}+1+\epsilon}. \tag{6}$$

However, since $T_k'(0) > 0$ for $k > 3$, and $T_k'(1) < 0$ for $k > 0$, the derivative has a root between 0 and 1 for $k > 3$ by the Intermediate Value Theorem. Moreover the Second Derivative Test confirms that for $k > 1$, each of these roots is a local maximum.

Numerical approximations of some relevant roots of this polynomial are shown in Table 2, alongside the expected network lifetime of the optimized algorithm. Our optimizations improve the expected average network lifetime by more than 3% above that of $\log_2$-`RoundRobin`.

k	ϵ	$T_k(0)$	$T_k(\epsilon)$	Gain %	$\lvert U_k(k;\epsilon)\rvert$%
2	0	1.492783	1.492783	0	50.00
3	0	1.614033	1.614033	0	50.00
4	0.211103	1.675576	1.696157	1.23	39.44
5	0.371297	1.706584	1.743439	2.16	31.44
6	0.448178	1.722149	1.767123	2.61	27.59
7	0.485871	1.729946	1.778990	2.84	25.71
8	0.504537	1.733848	1.784931	2.95	24.77
10	0.518459	1.736777	1.789391	3.03	24.08
12	0.521929	1.737509	1.790506	3.05	23.90
15	0.522941	1.737723	1.790831	3.06	23.85
20	0.523081	1.737752	1.790876	3.06	23.85

Table 2. Numerical Approximations for Optimal Choice of ϵ. Note that $T_{20}(0)$ equals $T_\infty(0) = \mu_T^*$ to six digits. The rightmost column shows the percentage of U that is covered by $\Gamma_k(k;\epsilon)$.

Theorem 3. *For sufficiently large n, the optimized $\log_2$-`RoundRobin` algorithm achieves an average-case approximation ratio of 0.895 with high probability.*

Convergence. The Ratio Test, combined with L'Hôpital's Rule, will show that both series $T_k(\epsilon)$ and $T_k'(\epsilon)$ converge as $k \to \infty$ for any fixed $\epsilon \in [0,1]$. As we have not found a closed functional form for either limit, we cannot prove that the optimal ϵ converges to a limit.

4 Open Problems

One obvious variation on this problem is to change the battery drainage rate. If $\alpha > 1$ then larger coverage regions become more expensive, so that, for example, the performance of Γ_1 would decline. Secondly, the average-case analysis could be studied for any probability distribution with finite support.

Another avenue for exploration would be to extend the analysis to higher dimensions, including one in which the sensors are not necessarily located on the line, but rather in the plane, and one in which the sensors remain on the line, but the coverage region extends into the plane.

Lastly, while we allow for pre-emptive scheduling in our definition, we did not actually use it in the case of random deployment. We hope to tackle some of these questions in future research.

References

1. Aloupis, G., Cardinal, J., Collette, S., Langerman, S., Orden, D., Ramos, P.: Decomposition of Multiple Coverings into More Parts. Discrete & Computational Geometry 44-3, 706–723 (2010)
2. Bar-Noy, A., Brown, T., Johnson, M.P., Liu, O.: Cheap or Flexible Sensor Coverage. In: Proceedings of the 5th IEEE International Conference on Distributed Computing in Sensor Systems, pp. 245–258 (2009)
3. Berman, P, Calinescu, G., Shah, C., Zelikovsky, A.: Efficient Energy Management in Sensor Networks, Ad Hoc and Sensor Networks (2005)
4. Buchsbaum, A.L., Efrat, A., Jain, S., Venkatasubramanian, S., Yi, K.: Restricted Strip Covering and the Sensor Cover Problem. In: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1056–1065 (2007)
5. Cardei, M., Wu, J., Lu, M.: Improving Network Lifetime Using Sensors with Adjustable Sensing Ranges. Int. J. Sensor Networks 1-1, 41-49 (2006)
6. Dhawan, A., Vu, CT, Zelikovsky, A., Li, Y., Prasad, SK: Maximum Lifetime of Sensor Networks with Adjustable Sensing Range. In: Proceedings of the International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD), pp. 285–289 (2006)
7. Gibson, M., Varadarajan, K.: Decomposing Coverings and the Planar Sensor Cover Problem. In: 50th Annual IEEE Symposium on Foundations of Computer Science, pp. 159–168 (2009)
8. Lev-Tov, N., Peleg, D.: Polynomial Time Approximation Schemes for Base Station Coverage with Minimum Total Radii. Computer Networks 47-4, 489–501 (2005)
9. Mitzenmacher, M., Upfal, E.: Probability and Computing: Randomized Algorithms and Probabilistic Analysis. Cambridge University Press, New York (2005)
10. Pach, J.: Covering the Plane with Convex Polygons. Discrete & Computational Geometry 1-1, 73–81 (1986)
11. Pach, J., Tóth, G.: Decomposition of Multiple Coverings into Many Parts. Computational Geometry 42-2, 127–133 (2009)
12. Wu, J., Yang, S.: CoverageIissue in Sensor Networks with Adjustable Ranges. In: Proceedings of the 2004 Intl. Conference on Parallel Processing Workshops, pp. 61–68 (2004)
13. Weisstein, Eric W. "q-Series." From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/q-Series.html

Appendix

Pinned Disks. For a fixed time t, let $Y = \{x_1, ..., x_m\} \subseteq X$ be the locations of active sensors, so that $r_i > 0$ for all $1 \leq i \leq m$. The unique radial assignment function $R^*(Y)$ corresponding to pinned disks is then given recursively by

$$(R^*(Y))_i = r_i = \begin{cases} x_1 & \text{if } i = 1 \\ x_i - (x_{i-1} + r_{i-1}) & \text{if } 2 \leq i \leq m \end{cases} .$$

Setting $x_m + r_m = 1$ to ensure a perfect fit yields

$$1 = 2\sum_{j=1}^{m}(-1)^{m-j}x_j \,.$$

We then define the polynomial $f(Y) = -\frac{1}{2} + \sum_{j=1}^{m}(-1)^{m-j}x_j$, and use it in the proof of Lemma 2.

Proof. (of Proposition 2) $\Rightarrow$ From our previous argument, a radial assignment that gives perfect coverage necessarily consists of pinned disks that satisfy $f(Y) = 0$.

$\Leftarrow$ Suppose that there exists $Y \subseteq X$ satisfying $f(Y) = 0$. Then $R^*(Y)$ gives perfect coverage.

RoundRobin. We present the proof of the lower appoximation bound for `RoundRobin` given in Lemma 1.

Proof. (of Lemma 1) To prove the lower bound, let $\beta \in (0, \frac{1}{4})$ be a parameter to be determined later. Let $A = [0, \beta], B = [\beta, 1-\beta]$, and $C = [1-\beta, 1]$ be a division of I into three closed intervals. Let $t_{(1,0,0)}$ denote any block of time in OPT in which sensors from A are active (i.e. - have non-zero radius), but no sensors from B or C are active. Note that in $t_{(1,0,0)}$ (respectively $t_{(0,0,1)}$), exactly one sensor is active in A (resp. C). Thus, for any such time block, `RoundRobin` gives the same solution as OPT. Furthermore, any non-empty time interval in OPT in which only one sensor is active gives the same solution as `RoundRobin`.

It remains to consider the following situations:

- $t_{(0,1,0)}$: The worst position for the sensors are at β and $1-\beta$, where the radii must be set to at least $1-\beta$. So the total network lifetime of `RoundRobin` in this situation is $T \geq \frac{n}{1-\beta}$. Since the maximum network lifetime is $2n$, we know that `RoundRobin` is at least a $\frac{1}{2(1-\beta)}$-approximation in this case.
- $t_{(1,1,0)} \sim t_{(0,1,1)}$: The worst case here is to have $\frac{n}{2}$ pairs of sensors at 0 and β, which then must be assigned radii of 1 and $1-\beta$, respectively, under `RoundRobin`. The lifetime of `RoundRobin` is thus at least $T \geq \frac{n}{2} \cdot 1 + \frac{n}{2} \cdot \frac{1}{1-\beta} = \frac{2-\beta}{2(1-\beta)}n$. The approximation ratio of `RoundRobin` is thus at least $\frac{2-\beta}{4(1-\beta)}$.

- $t_{(1,0,1)}$: With no sensors active in B, the worst case scenario for `RoundRobin` is a lifetime of n, with all n sensors at either 0 or 1. However, note that since $\beta < \frac{1}{4}$, OPT cannot achieve a lifetime of $2n$ under these conditions. [Note in light of Corollary 1, that no subset $Y \subseteq X$ satisfies $f_k(Y) = 0$.] In fact, the maximum network lifetime for OPT occurs when there are $\frac{n}{2}$ pairs of sensors at β and $1-\beta$, each with radii set to $\frac{1}{2} - \beta$. Thus, the lifetime of OPT is at most $\frac{n}{2} \cdot \frac{1}{\frac{1}{2}-\beta} = \frac{n}{1-2\beta}$. The approximation ratio of `RoundRobin` is thus at least $1 - 2\beta$.
- $t_{(1,1,1)}$: Here the worst case is to have $\frac{n}{3}$ sensors at β or $1-\beta$, and corresponding pairs at 0 and 1. The lifetime of `RoundRobin` under this scenario is $T \geq \frac{n}{3} \cdot \frac{1}{1-\beta} + \frac{2n}{3} \cdot 1 = \frac{3-2\beta}{3(1-\beta)} n$. The approximation ratio of `RoundRobin` is then at least $\frac{3-2\beta}{6(1-\beta)}$.

Thus, for any possible arrangement of active sensors, the approximation ratio of `RoundRobin` is at least

$$\rho(\beta) \geq min \left\{ 1, \frac{1}{2(1-\beta)}, \frac{2-\beta}{4(1-\beta)}, 1-2\beta, \frac{3-2\beta}{6(1-\beta)} \right\}$$

Since $1 \geq \frac{1}{2(1-\beta)} \geq \frac{2-\beta}{4(1-\beta)} \geq \frac{3-2\beta}{6(1-\beta)}$ for any $0 < \beta < \frac{1}{4}$, the optimal choice of β occurs when $1 - 2\beta = \frac{3-2\beta}{6(1-\beta)} \Rightarrow \beta = \frac{4-\sqrt{7}}{6} \approx 0.226$. The minimum value of ρ is thus $\frac{\sqrt{7}-1}{3} \approx 0.54857$.