8-15-2015

Set It and Forget It: Approximating the Set Once Strip Cover Problem

Amotz Bar-Noy
City University of New York

Benjamin Baumer
Smith College, bbaumer@smith.edu

Dror Rawitz
Bar-Ilan University, Israel

Set It and Forget It: Approximating the Set Once Strip Cover Problem

Amotz Bar-Noy
Department of Computer Science
The Graduate Center of the CUNY
New York, NY 10016, USA
amotz@sci.brooklyn.cuny.edu

Ben Baumer
Department of Mathematics & Statistics
Smith College
Northampton, MA 01063, USA
bbaumer@smith.edu

Dror Rawitz*
Faculty of Engineering
Bar-Ilan University
Ramat Gan 52900, Israel
dror.rawitz@biu.ac.il

July 20, 2015

Abstract

We consider the Set Once Strip Cover problem, in which n wireless sensors are deployed over a one-dimensional region. Each sensor has a fixed battery that drains in inverse proportion to a radius that can be set just once, but activated at any time. The problem is to find an assignment of radii and activation times that maximizes the length of time during which the entire region is covered. We show that this problem is NP-hard and that RoundRobin, the algorithm in which the sensors take turns covering the entire region, has a tight approximation guarantee of $\frac{3}{2}$. Moreover, we show that the more general class of *duty cycle* algorithms, in which groups of sensors take turns covering the entire region, can do no better. We present similar results for the more general Strip Cover problem, in which each radius may be set finitely-many times. Finally, we give an $O(n^2 \log n)$-time optimization algorithm for the related Set Radius Strip Cover problem, in which sensors must be activated immediately.

Keywords: wireless sensor networks, strip cover, barrier coverage, network lifetime.

*Supported in part by the Israel Science Foundation (grant no. 497/14).

1 Introduction

Suppose that n sensors are deployed over a one-dimensional region that they are to cover with a wireless network. Each sensor is equipped with a finite battery charge that drains in inverse proportion to the sensing radius that is assigned to it, and each sensor can be activated only once. In the Set Once Strip Cover (OnceSC) problem, the goal is to find an assignment of radii and activation times that maximizes the *lifetime* of the network, namely the length of time during which the entire region is covered.

Formally, we are given as input the locations $x \in ([0, 1] \cap \mathbb{Q})^n$ and battery charges $b \in \mathbb{Q}_+^n$ for each of n sensors (where $\mathbb{Q}_+ = \mathbb{Q} \cap [0, \infty)$). While we cannot move the sensors, we do have the ability to set the sensing radius $\rho_i \in [0, 1] \cap \mathbb{Q}$ of each sensor and the time $\tau_i \in \mathbb{Q}_+$ when it should become active. Since each sensor's battery drains in inverse proportion to the radius we set (but cannot subsequently change), each sensor covers the region $[x_i - \rho_i, x_i + \rho_i]$ for b_i/ρ_i time units. Our task is to devise an algorithm that finds a schedule $S = (\rho, \tau)$ for any input (x, b), such that $[0, 1]$ is completely covered for as long as possible.

Motivation. Scheduling problems of this ilk arise in many applications, particularly when the goal is *barrier coverage* (see [9, 22] for surveys, or [14] for motivation). Suppose that we have a highway, supply line, or fence in territory that is either hostile or difficult to navigate. While we want to monitor activity along this line, conditions on the ground make it impossible to systematically place wireless sensors at specific locations. However, it is feasible and inexpensive to deploy adjustable range sensors along this line by, say, dropping them from an airplane flying overhead (e.g. [8, 19, 21]). Once deployed, the sensors send us their location via GPS, and we wish to send a single radius-time pair to each sensor as an assignment. Replacing the battery in any sensor is infeasible. How do we construct an assignment that will keep this vital supply line completely monitored for as long as possible?

Models. While the focus of this paper is the OnceSC problem, we touch upon three closely related problems. In each problem the location and battery of each sensor are fixed, and a solution can be viewed as a finite set of radius-time pairs. In OnceSC, both the radii and the activation times are variable, but can be set only once. In the more general Strip Cover problem, the radius and activation time of each sensor can be set finitely many times. On the other hand, if the radius of each sensor is fixed and given as part of the input, then we call the problem of assigning an activation time to each sensor so as to maximize network lifetime Set Time Strip Cover (TimeSC). Set Radius Strip Cover (RadSC) is another variant of OnceSC in which all of the sensors are scheduled to activate immediately, and the problem is to find the optimal radial assignment. Figure 1 summarizes the important differences between related problems and illustrates their relationship to one another.

Related work. TimeSC, which is known as Restricted Strip Covering, was shown to be NP-hard by Buchsbaum et al. [7], who also gave an $O(\log \log \log n)$-approximation algorithm. Later, a constant factor approximation algorithm was discovered by Gibson and Varadarajan [13].

Close variants of RadSC have been the subject of previous work. Whereas RadSC requires *area* coverage, Peleg and Lev-Tov [15] studied *target* coverage. In this problem the input is a set of n sensors and a finite set of m points on the line that are to be covered, and the goal is to find the

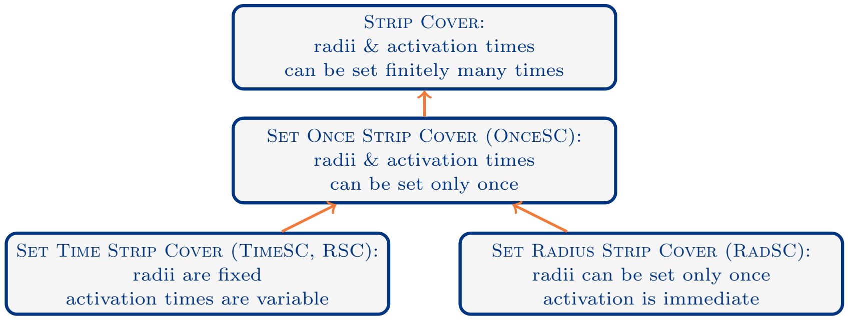


Figure 1: Relationship of Problem Variants.

radial assignments with the minimum sum of radii. They used dynamic programming to devise a polynomial time alorithm. Bar-Noy et al. [5] improved the running time to $O(n+m)$. Recently, Bar-Noy et al. [6] considered an extension of RadSC in which sensors are mobile.

Strip Cover was first considered by Bar-Noy and Baumer [3], who gave a $\frac{3}{2}$ lower bound on the performance of RoundRobin, the algorithm in which the sensors take turns covering the entire region (see Observation 2), but were only able to show a corresponding upper bound of 1.82. The similar Connected Range Assignment (CRA) problem, in which radii are assigned to points in the plane in order to obtain a connected disk graph, was studied by Chambers et al. [12]. They showed that the best one circle solution to CRA also yields a $\frac{3}{2}$-approximation guarantee, and in fact, the instance that produces their lower bound is simply a translation of the instance used in Observation 2.

The notion of *duty cycling* as a mean to maximize network lifetime was also considered in the literature of discrete geometry. In this context, maximizing the number of covers t serves as a proxy for maximizing the actual network lifetime. Pach [16] began the study of decomposability of multiple coverings. Pach and Tóth [17] showed that a t-fold cover of translates of a centrally-symmetric open convex polygon can be decomposed into $\Omega(\sqrt{t})$ covers. This result was later improved to the optimal $\Omega(t)$ covers by Aloupis et al. [2], while Gibson and Varadarajan [13] showed the same result without the centrally-symmetric restriction.

Motivated by prior invocations of duty cycling [20, 18, 1, 8, 10, 11], Bar-Noy et al. [4] studied a duty cycle variant of OnceSC with unit batteries in which sensors must be grouped into shifts of size at most k that take turns covering $[0,1]$. (RoundRobin is the only possible algorithm when $k=1$.) They presented a polynomial-time algorithm for $k=2$ and showed that the approximation ratio of this algorithm is $\frac{35}{24}$ for $k>2$. It was also shown that its approximation ratio is at least $\frac{15}{11}$, for $k \geq 4$, and $\frac{6}{5}$, for $k=3$. A fault-tolerance model, in which smaller shifts are more robust, was also proposed.

Our results. We introduce the Set Once model that corresponds to the case where the scheduler does not have the ability to vary the sensor's radius once it has been activated. We show that OnceSC and Strip Cover are NP-hard (Section 3) and that RoundRobin is a $\frac{3}{2}$-approximation algorithm for both OnceSC and Strip Cover (Section 4). This closes a gap between the best previously known lower and upper bounds ($\frac{3}{2}$ and 1.82, resp.) on the performance of this algorithm. Our analysis of RoundRobin is based on the following approach: We slice an optimal schedule into temporal (horizontal) strips in which the set of active sensors is fixed (see Figure 4). For each such strip we construct an instance with unit batteries and compare the performance of RoundRobin to the RadSC optimum of this instance. In Section 5 we show that the class of duty cycle algorithms cannot improve on this $\frac{3}{2}$ guarantee. In Section 6, we provide an $O(n^2 \log n)$-time exact optimization algorithm for RadSC. We note that the same approach would work for the case where, for every sensor i, the ith battery is drained in inverse proportion to ρ_i^α, for some $\alpha > 0$.

2 Preliminaries

Problems. The Set Once Strip Cover (abbreviated OnceSC) is defined as follows. Let $U \triangleq [0,1]$ be the interval that we wish to cover. Given is a vector $x = (x_1, \ldots, x_n) \in ([0,1] \cap \mathbb{Q})^n$ of n sensor locations, and a corresponding vector $b = (b_1, \ldots, b_n) \in \mathbb{Q}_+^n$ of battery charges. We assume that $x_i \leq x_{i+1}$ for every $i \in \{1, \ldots, n-1\}$. We sometimes abuse notation by treating x as a set. An instance of the problem thus consists of a pair $I = (x, b)$, and a solution is an assignment of radii and activation times to sensors. More specifically a solution (or *schedule*) is a pair $S = (\rho, \tau) \in \mathbb{Q}_+^n \times \mathbb{Q}_+^n$,[1] where ρ_i is the *radius* of sensor i and τ_i is the *activation time* of i. Since the radius of each sensor cannot be reset, this means that sensor i becomes active at time τ_i, covers the *range* $[x_i - \rho_i, x_i + \rho_i]$ for b_i/ρ_i time units, and then becomes inactive since it has exhausted its entire battery.

Any schedule can be visualized by a space-time diagram in which each coverage assignment can be represented by a rectangle. It is customary in such diagrams to view the sensor locations as forming the horizontal axis, with time extending upwards vertically. In this case, the coverage of a sensor located at x_i and assigned the radius ρ_i beginning at time τ_i is depicted by a rectangle with lower-left corner $(x_i - \rho_i, \tau_i)$ and upper-right corner $(x_i + \rho_i, \tau_i + b_i/\rho_i)$. Let the set of all points contained in this rectangle be denoted as $Rect(\rho_i, \tau_i)$. A point (u, t) in space-time is *covered* by a schedule (ρ, τ) if $(u, t) \in \bigcup_i Rect(\rho_i, \tau_i)$. The *lifetime* of the network in a solution $S = (\rho, \tau)$ is the maximum value T such that every point $(u, t) \in [0,1] \times [0, T]$ is covered. Graphical depictions of two schedules are shown below in Figure 2.

In OnceSC the goal is to find a schedule $S = (\rho, \tau)$ that maximizes the lifetime T. Given an instance $I = (x, b)$, the optimal lifetime is denoted by $\textsc{Opt}(x, b)$. (We sometimes use Opt, when the instance is clear from the context.)

The Set Radius Strip Cover (RadSC) problem is a variant of OnceSC in which $\tau_i = 0$, for every i. Hence, a solution is simply a radial assignment ρ. Set Time Strip Cover (TimeSC) is another variant in which the radii are given in the input, and a solution is an assignment of activation times to sensors.

Strip Cover is a generalization of OnceSC in which a sensor's radius may be changed finitely many times. In this case a solution is a vector of piece-wise constant functions $\rho(t)$, where $\rho_i(t)$ is the sensing radius of sensor i at time t. The segment $[0,1]$ is covered at time t, if $[0,1] \subseteq \bigcup_i [x_i - \rho_i(t), x_i + \rho_i(t)]$. The solution achieves a lifetime of T, if $[0,1]$ is covered for all $t \in [0, T]$, and if $\int_0^\infty \rho_i(t)\,dt \leq b_i$, for every i.

Maximum lifetime. The best possible lifetime of an instance (x, b) is $2\sum_i b_i$. We state this formally for OnceSC, but the same holds for the other variants of the problem.

Observation 1. *The lifetime of a* OnceSC *instance* (x, b) *is at most* $2\sum_i b_i$.

[1] If $S \in \mathbb{R}_+^n \times \mathbb{R}_+^n$, our upper bound on the approximation ratio of RoundRobin would be $\frac{3}{2} + \epsilon$, for any $\epsilon > 0$.

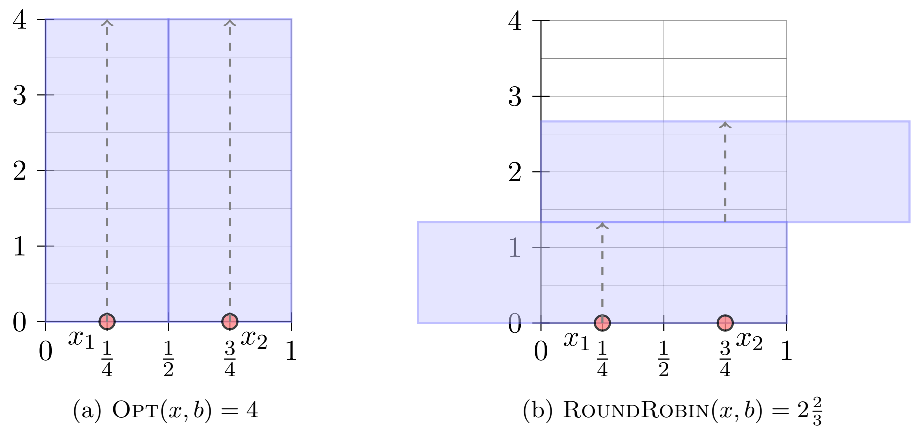


Figure 2: RoundRobin vs. Opt with $x = (\frac{1}{4}, \frac{3}{4})$ and $b = (1,1)$. The sensors are indicated by (red) dots. Each of the (blue) rectangles represents the active coverage region for one sensor. The dashed gray arrow helps to clarify which sensor is active at a particular point in time.

Proof. Consider an optimal solution (ρ, τ) for (x, b) with lifetime T. A sensor i covers an interval of length $2\rho_i$ for $\frac{b_i}{\rho_i}$ time. The lifetime T of the network is at most the total area of space-time covered by the sensors, which is at most $\sum_i 2\rho_i \cdot b_i/\rho_i$. □

Round Robin. We focus on a simple algorithm, we call RoundRobin, which forces the sensors to take turns covering $[0,1]$, namely it assigns, for every i, $\rho_i = r_i \triangleq \max\{x_i, 1 - x_i\}$ and $\tau_i = \sum_{j=1}^{i-1} b_j/\rho_j$. The lifetime of RoundRobin is thus

$$\text{RR}(x,b) \triangleq \sum_{i=i}^{n} \frac{b_i}{r_i} \ .$$

Notice that Observation 1 implies an upper bound of 2 on the approximation ratio of RoundRobin, since $r_i \leq 1$, for every i. A lower bound of $\frac{3}{2}$ on the approximation guarantee of RoundRobin was given in [3] using the two sensor instance $x = (\frac{1}{4}, \frac{3}{4})$, $b = (1,1)$. The relevant schedules are depicted graphically in Figure 2.

Observation 2 ([3])**.** *The approximation ratio of* RoundRobin *is at least* $\frac{3}{2}$.

Given an instance (x, b) of OnceSC, let $B \triangleq \sum_i b_i$ be the total battery charge of the system and $\overline{r} = \sum_i \frac{b_i}{B} \cdot r_i$ be the average of the r_i's, weighted by their respective battery charge. We define the following lower bound on $\text{RR}(x,b)$:

$$\text{RR}'(x,b) \triangleq \frac{B}{\overline{r}} \ .$$

Lemma 3. $\text{RR}'(x,b) \leq \text{RR}(x,b)$*, for every* OnceSC *instance* (x,b)*.*

Proof. We have that

$$\text{RR}(x,b) = \sum_{i=1}^{n} \frac{b_i}{r_i} = \sum_{i=1}^{n} \frac{b_i^2}{b_i r_i} \geq \frac{(\sum_{i=1}^{n} b_i)^2}{\sum_{i=1}^{n} b_i r_i} = \frac{\sum_{i=1}^{n} b_i}{\overline{r}} = \text{RR}'(x,b) \ ,$$

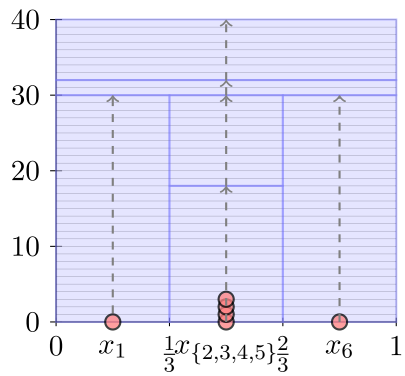


Figure 3: Proof of NP-hardness. $Y = \{1, 2, 3, 4\}$ is a given instance of Partition, and $(x, b) = \big((\frac{1}{6}, \frac{1}{2}, \dots, \frac{1}{2}, \frac{5}{6}), (5, 1, 2, 3, 4, 5)\big)$ is the translated OnceSC instance.

where the inequality is due to the following implication of the Cauchy-Schwarz Inequality: $\sum_j \frac{c_j^2}{d_j} \geq \frac{(\sum_j c_j)^2}{\sum_j d_j}$, for any positive $c, d \in \mathbb{R}^n$. □

3 Hardness Results

In this section we show that OnceSC and Strip Cover are NP-hard. This is done using a reduction from Partition.

Theorem 1. OnceSC *is NP-hard.*

Proof. Let $y = \{y_1, \dots, y_n\}$ be a given instance of Partition, and define $\beta = \frac{1}{2}\sum_{i=1}^{n} y_i$. We create an instance of OnceSC with $n + 2$ sensors, where the n sensors are located at $\frac{1}{2}$, and the two remaining sensors are located at $\frac{1}{6}$ and at $\frac{5}{6}$. Sensor i, for $2 \leq i \leq n + 1$, has a battery charge $b_i = y_{i-1}$, while $b_1 = b_{n+2} = \beta$. That is, the instance of OnceSC consists of sensor locations $x = (\frac{1}{6}, \underbrace{\frac{1}{2}, \dots, \frac{1}{2}}_{n}, \frac{5}{6})$ and batteries $b = (\beta, y_1, \dots, y_n, \beta)$. We show that $y \in$ Partition if and only if the maximum possible lifetime of 8β is achievable for the OnceSC instance.

First, suppose $y \in$ Partition, hence there exists a subset $I \subseteq \{1, \dots, n\}$, such that $\sum_{i \in I} y_i = \sum_{i \notin I} y_i = \beta$. Schedule the sensors that correspond to I to iteratively cover the region $[\frac{1}{3}, \frac{2}{3}]$. Since all of these sensors are located at $\frac{1}{2}$, this requires that each sensor's radius be set to $\frac{1}{6}$, i.e. $\rho_{i+1} = \frac{1}{6}$, for every $i \in I$. Since the sum of their batteries is β, this region can be covered for exactly 6β time units. With the help of the additional sensors located at $\frac{1}{6}$ and $\frac{5}{6}$, whose radii are also set to $\rho_1 = \rho_{n+2} = \frac{1}{6}$, the sensors that correspond to I can thus cover $[0, 1]$ for 6β time units (see Figure 3 for an example). Next, by assigning $\rho_{i+1} = \frac{1}{2}$, for every $i \notin I$, the sensors that correspond to $\{1, \dots, n\} \setminus I$ can cover $[0, 1]$ for an additional 2β time units. Thus, the total lifetime is 8β.

Now suppose that for such a OnceSC instance, the lifetime of 8β is achievable. Since the maximum possible lifetime is achievable, no coverage can be wasted in the optimal schedule. In this case the radii of the sensors at $\frac{1}{6}$ and $\frac{5}{6}$ must be exactly $\frac{1}{6}$, since otherwise, they would either not reach the endpoints $\{0, 1\}$, or extend beyond them. Moreover, due the fact that all of the other sensors are located at $\frac{1}{2}$, and their coverage is thus symmetric with respect to $\frac{1}{2}$, it cannot be the case that sensor 1 and sensor $n+2$ are active at different times. Thus, the solution requires a partition of the sensors located at $\frac{1}{2}$ into two groups: the first of which must work alongside sensors 1 and $n+2$ with a radius of $\frac{1}{6}$ and a combined lifetime of 6β; and the second of which must implement RoundRobin for a lifetime of 2β. The sensor partition induces a solution to the Partition instance. □

A similar approach is used for Strip Cover.

Theorem 2. Strip Cover *is NP-hard.*

Proof. We use the reduction that was used in the proof of Theorem 1. In this proof it was also shown that if $y \in$ Partition, then a set once schedule with a lifetime of 8β is achievable. It remains to show that if a lifetime of 8β is achievable, then $y \in$ Partition.

Consider a schedule that achieves a lifetime of 8β. Recall that this is an optimal schedule, and thus no coverage can be wasted. Since the radii can change a finite number of times, such a schedule can be split into temporal (horizontal) strips in which the radii remain unchanged. Consider such a strip in which sensor 1 participates in the cover. The radius of sensor 1 in such a strip must be exactly $\frac{1}{6}$, since otherwise it would not reach the barrier end-point or extend beyond it. Since other sensors are located at $\frac{1}{2}$, it must be that sensor $n+2$ is active in this strip with a radius of $\frac{1}{6}$.

Change the order of the strips, such that all strips in which sensors 1 and $n+2$ are active are placed first. In the resulting schedule sensors 1 and $n+2$ are set once. Moreover, due to the optimality of the schedule, it partitions the rest of the sensors into two subsets: sensors that work alongside sensors 1 and $n+2$ with a radius of $\frac{1}{6}$, and the sensors that do not. The rest of the proof is similar to the proof of Theorem 1. □

4 Round Robin

We showed in Section 3 that OnceSC and Strip Cover are NP-hard, so here we turn our attention to approximation algorithms. While RoundRobin is among the simplest possible algorithms (note that its running time is exactly n), the precise value of its approximation ratio is not obvious (although it is not hard to see that 2 is an upper bound). In [3] an upper bound of 1.82 and a lower bound of $\frac{3}{2}$ were shown. In this section, we show that the approximation ratio of RoundRobin is exactly $\frac{3}{2}$. The structure of the proof is as follows:

- In Section 4.1 we start with an optimal schedule S, and cut it into disjoint time intervals, or *strips*, such that a fixed set of sensors is active throughout each time interval. Each strip induces a RadSC instance I_j and a corresponding solution S_j. Given a strip I_j and a schedule S_j that has a lifetime of T_j, we remove redundant sensors and decrease radii of extreme sensors while preserving the strip lifetime T_j. (We explain why in the third bullet.) The new instance and radii are denoted by $\hat{I}_j$ and $\hat{\rho}^j$. We note that the RoundRobin lifetime may only decrease, namely $\text{RR}(\hat{I}_j) \leq \text{RR}(I_j)$.

- Next in Section 4.2 we show that for any such pair of a strip $\hat{I}_j$ and a radii vector $\hat{\rho}^j$, there exists a uniform-battery instance I'_j and a radii vector σ^j with the same lifetime T_j. In I'_j each sensor is represented by a set of sensors that depend on the size of his battery. We make sure that each original sensor is represented by at least three new sensors. (Again, this is explained in the third bullet.) Here we show that $\text{RR}'(I'_j) \leq \text{RR}'(\hat{I}_j)$. We note that I'_j may contain sensors outside $[0, 1]$.

- In Section 4.3 we prove a lower bound on the performance of RoundRobin on such uniform battery instances. First, we show that it is enough to consider unit batteries. Then, we construct a unit battery instance I''_j by "stretching" (the unit battery version of) I'_j such that $\text{Opt}_0(I''_j) = \text{Opt}_0(I'_j)$ and $\text{RR}'(I''_j) \le \text{RR}'(I'_j)$ and show that $\text{RR}'(I''_j) \ge \frac{2}{3}\text{Opt}_0(I''_j)$. The stretching operation is based on increasing gaps between sensors such that all gaps become the original maximum gap within $[0, 1]$. Hence, we need to make sure that the gaps outside $[0, 1]$ are not larger than the maximum gap within $[0, 1]$. This is achieved by decreasing the radii of extreme (original) sensors and by representing each original sensor by at least three new uniform sensor.

- By combining these results, we prove in Section 4.4 that $\text{RR}(x, b) \ge \frac{2}{3}\text{Opt}(x, b)$.

4.1 Cutting the Schedule into Strips

Given an instance $I = (x, b)$, and a solution $S = (\rho, \tau)$ with lifetime $T = \text{Opt}(x, b)$, let Ω be the set of times until T in which a sensor was turned on or off, namely $\Omega = \bigcup_i \{\tau_i, \tau_i + b_i/\rho_i\} \cap [0, T]$. Let $\Omega = \{0 = \omega_0, \ldots, \omega_\ell = T\}$, where $\omega_j < \omega_{j+1}$, for every j. Observe that $\omega_j \in \mathbb{Q}_+$, for every j.

Next, we partition the time interval $[0, T]$ into the sub-intervals $[\omega_j, \omega_{j+1}]$, for every $j \in \{0, \ldots, \ell - 1\}$, and we define a new instance for every sub-interval. For every $j \in \{0, \ldots, \ell - 1\}$, let $x^j \subseteq x$ be the set of sensors that participate in covering $[0, 1]$ during the jth sub-interval of time, i.e., $x^j = \{x_i : [\omega_j, \omega_{j+1}] \subseteq [\tau_i, \tau_i + b_i/\rho_i]\}$. Also, let $T_j = \omega_{j+1} - \omega_j$, and let b^j_i be the energy that was consumed by sensor i during the jth sub-interval, i.e., $b^j_i = \rho_i \cdot T_j$. Observe that $b^j_i \in \mathbb{Q}_+$, for every i. Also, observe that $I_j = (x^j, b^j)$ can be seen as a valid RadSC instance, for which ρ^j, where $\rho^j_i = \rho_i$ for every sensor i such that $x_i \in x^j$, is a solution that achieves a lifetime of exactly T_j.

We further modify the battery charges b^j and the radii ρ^j as follows:

- Iteratively remove redundant sensors: starting with $i = 1$, remove sensor i from the instance if the interval $[0, 1]$ remains covered during $[\omega_j, \omega_{j+1}]$ without i.

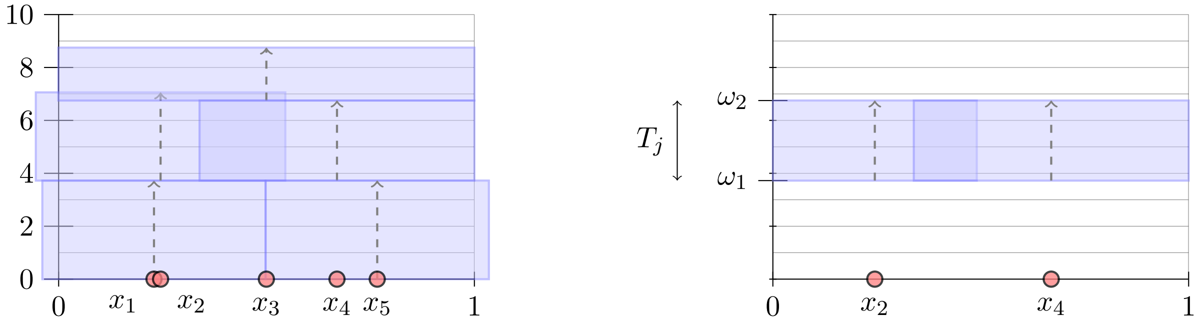
- Decrease the batteries and the radii of the left-most and the right-most sensors as much as possible.

The resulting instance and schedule are denoted by $\hat{I}_j = (\hat{x}^j, \hat{b}^j)$ and $\hat{\rho}^j$. Figure 4 provides an illustration of this procedure.

Observation 4. *$\hat{I}_j = (\hat{x}^j, \hat{b}^j)$ and $\hat{\rho}^j$ satisfy the following:*

1. *Let sensors 1 and m be the leftmost and rightmost sensors in $\hat{x}^j$. Then,*
   - *either $\hat{\rho}^j_1 = \hat{x}^j_1$ or the interval $[0, \hat{x}^j_1 + \hat{\rho}^j_1)$ is only covered by sensor 1.*
   - *either $\hat{\rho}^j_m = 1 - \hat{x}^j_m$ or the interval $(\hat{x}^j_m - \hat{\rho}^j_m, 1]$ is only covered by sensor m.*
2. *$\hat{b}^j, \hat{\rho}^j \in \mathbb{Q}^n_+$.*

For now, it is important to note that $\text{RR}(\hat{x}^j, \hat{b}^j) = \sum_{x_i \in \hat{x}^j} \frac{\hat{b}^j_i}{r_i} \le \text{RR}(x^j, b^j)$ is the RoundRobin lifetime of the jth strip, which can also be seen as a specific RadSC instance I_j with the properties outlined above. Also note that while S is an optimal solution to I, it is not necessarily the case that $\hat{\rho}^j$ (with lifetime T_j) correspond to an optimal solution to $\hat{I}_j$.



(a) An optimal schedule S. (b) The second strip of that schedule.

Figure 4: Cutting an optimal schedule into strips. Note that coverage overlaps may occur in both the horizontal and vertical directions in the optimal schedule, but only horizontally in a strip. At right, the second strip is shown. Note that in the third strip (not shown), the sensor x_2 is redundant (since x_3 covers the interval by itself) and will be removed.

4.2 Reduction to Set Radius Strip Cover with Uniform Batteries

Given the RadSC instance $\hat{I}_j = (\hat{x}^j, \hat{b}^j)$ and a solution $\hat{\rho}^j$, we construct an uniform-size battery instance $I'_j = (y^j, (\beta^j)^n)$, where $\beta^j \in \mathbb{Q}$, and a RadSC solution σ^j, such that the lifetime of σ^j is T_j.

The instance I'_j is constructed as follows. First, let β^j be small enough such that $\hat{b}^j/\beta^j \in \mathbb{N}^m$ and $\hat{b}^j_i/\beta^j \geq 3$, for every i such that $\hat{x}^j_i \in \hat{x}^j$. We replace each sensor i, such that $\hat{x}^j_i \in \hat{x}^j$, with $\hat{b}^j_i/\beta^j$ sensors with battery β^j whose average location is $\hat{x}_i$. To do this, we divide the interval $[\hat{x}^j_i - \hat{\rho}^j_i, \hat{x}^j_i + \hat{\rho}^j_i]$ into $\hat{b}^j_i/\beta^j$ equal sub-intervals, and place a unit battery sensor in the middle of each sub-interval. These new uniform battery sensors are called the *children* of sensor i. Observe that child sensors may be placed outside $[0, 1]$, namely to the left of 0 or to the right of 1. Also, observe that $\sum_{k:k \text{ child of } i} \beta^j = \hat{b}^j_i$. The solution σ^j is defined as follows. For any child k of a sensor i in $\hat{I}_j$, we set $\sigma^j_k = \hat{\rho}^j_i/(\hat{b}^j_i/\beta^j)$. An example is shown in Figure 5.

Lemma 5. *The lifetime of σ^j is T_j.*

Proof. First, the $\hat{b}^j_i/\beta^j$ children of sensor i in $\hat{I}_j$ cover the interval $[\hat{x}^j_i - \hat{\rho}^j_i, \hat{x}^j_i + \hat{\rho}^j_i]$. Also, a child k of i survives $\beta^j/\sigma^j_k = \hat{b}^j_i/\hat{\rho}^j_i = T_j$ time units. □

Next, we prove that the lower bound on the performance of RoundRobin may only decrease.

Lemma 6. $\mathrm{RR}'(y^j, (\beta^j)^n) \leq \mathrm{RR}'(\hat{x}^j, \hat{b}^j)$.

Proof. Let p^j be the RoundRobin radii of y^j. Observe that if $\hat{x}^j_i \leq \frac{1}{2}$, it follows that

$$\sum_{k:k \text{ child of } i} p^j_k = \sum_{k:k \text{ child of } i} \max\left\{y^j_k, 1 - y^j_k\right\} \geq \sum_{k:k \text{ child of } i} (1 - y^j_k) \;=\; \hat{b}^j_i/\beta^j \cdot (1 - \hat{x}^j_i) \;=\; \hat{b}^j_i/\beta^j \cdot r^j_i \;,$$

where the inequality stems from the possibility that a child of i may be located to the right of $\frac{1}{2}$, and following equality is due to an averaging argument. By similar reasoning, if $\hat{x}^j_i \geq \frac{1}{2}$, we have that

$$\sum_{k:k \text{ child of } i} p^j_k = \sum_{k:k \text{ child of } i} \max\left\{y^j_k, 1 - y^j_k\right\} \geq \sum_{k:k \text{ child of } i} y^j_k \;=\; \hat{b}^j_i/\beta^j \cdot \hat{x}^j_i \;=\; \hat{b}^j_i/\beta^j \cdot r^j_i \;.$$

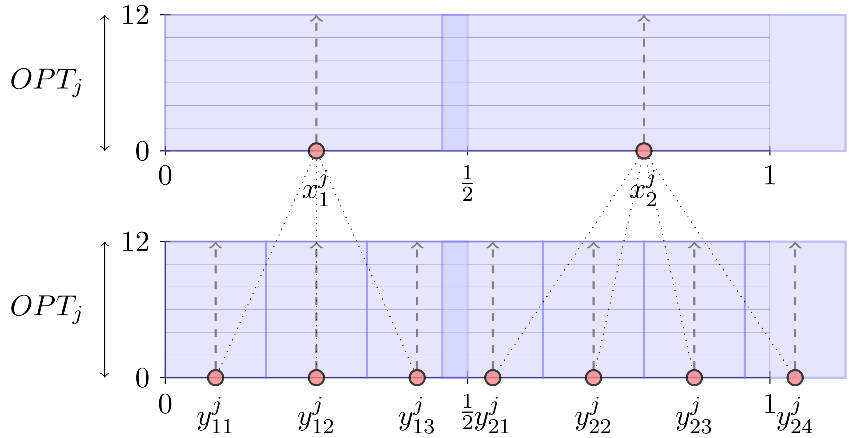


Figure 5: Reduction of a non-uniform battery strip I_j to a uniform battery instance I'_j: At the top, $I_j = \left((\frac{1}{4}, \frac{19}{24}), (3,4)\right)$, while at the bottom, $I'_j = \left((\frac{1}{12}, \frac{3}{12}, \frac{5}{12}, \frac{13}{24}, \frac{17}{24}, \frac{21}{24}, \frac{25}{24}), 1^n\right)$.

Hence,

$$\mathrm{RR}'(y^j, (\beta^j)^n) = \frac{\sum_i \hat{b}_i^j/\beta^j}{\overline{p^j}} = \frac{B^j}{\frac{1}{B^j}\sum_k p_k^j} \leq \frac{B^j}{\frac{1}{B^j}\sum_{\hat{x}_i^j \in \hat{x}^j} \hat{b}_i^j/\beta^j \cdot r_i^j} = \frac{B^j}{\overline{r^j}} = \mathrm{RR}'(\hat{x}^j, \hat{b}^j) \ ,$$

and the lemma follows. □

4.3 Analysis of Round Robin for Unit Batteries

In the previous section we obtained a uniform battery instance $(y^j, (\beta^j)^n)$. Our next step is to show that we may assume that we are given unit charge batteries. Notice that an RadSC solution σ has lifetime t for $(y^j, (\beta^j)^n)$ if and only if it has lifetime t/β^j for $(y^j, \mathbf{1})$. In particular we have that $\mathrm{Opt}_0(y^j, \mathbf{1}) = \mathrm{Opt}_0(y^j, \mathbf{1})/\beta^j$, where Opt_0 denote the optimal RadSC lifetime, and $\mathrm{RR}(y^j, \mathbf{1}) = \mathrm{RR}(y^j, (\beta^j)^n)/\beta^j$. Furthermore, $\mathrm{RR}'(y^j, \mathbf{1}) = \mathrm{RR}'(y^j, 1^n)/\beta^j$.

For the remainder of this section, we assume that we are given a unit battery instance x that corresponds to the jth strip. (We drop the subscript j and go back to x for readability.) Recall that $x \cap [0,1]$ is not necessarily equal to x, since some children could have been created outside $[0,1]$ in the previous step. We show that $\mathrm{RR}'(x) \geq \frac{2}{3}\mathrm{Opt}_0(x)$.

Let $i_0 = \min\{i : x_i \geq 0\}$ and let $i_1 = \max\{i : x_i \leq 1\}$ be the indices of the left-most and right-most sensors in $[0,1]$, respectively.

Lemma 7. $\max_{i \in \{i_0,\ldots,i_1-1\}}\{x_{i+1} - x_i\} = \max_{i \in \{1,\ldots,n-1\}}\{x_{i+1} - x_i\}$.

Proof. By Observation 4 either $\rho_1 = x_1$ and hence none of its children are located to the left of 0, or the points to the left of $x_1 + \rho_1$ are only covered by sensor 1, which means that the gaps between 1's children to the left of zero also appears between its children within $[0,1]$. (Recall that $b_i^j/\beta^j \geq 3$, for all i.) The same argument can be used for the right-most sensor. □

As illustrated in Figure 6, we define

$$\Delta_0 \triangleq \begin{cases} x_{i_0} - x_{i_0-1} & i_0 > 1, -x_{i_0-1} < x_{i_0}, \\ 2x_{i_0} & \text{otherwise,} \end{cases}$$

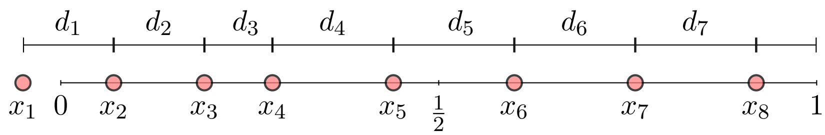


Figure 6: Illustration of the gaps in a unit battery instance x. Note that $i_0 = 2$ and $i_1 = 8$. $\Delta_0 = d_1$, since sensor 1 is closer to 0 than sensor 2. Also, $\Delta_1 = 2(1-x_8)$. Hence, $\Delta = \max\{d_4, d_1, 2(1-x_8)\}$.

$$\Delta_1 \triangleq \begin{cases} x_{i_1+1} - x_{i_1} & i_1 < n, x_{i_1+1} - 1 < 1 - x_{i_1}, \\ 2(1 - x_{i_1}) & \text{otherwise}, \end{cases}$$

and

$$\Delta \triangleq \max\left\{\Delta_0, \Delta_1, \max_{i \in \{i_0,\ldots,i_1-1\}} \{x_{i+1} - x_i\}\right\} .$$

We describe the optimal RadSC lifetime in terms of Δ.

Lemma 8. *The optimum lifetime of x is $\frac{2}{\Delta}$.*

Proof. To verify that $2/\Delta$ can be achieved, consider the solution in which $\rho_i = \Delta/2$ for all i. Clearly, $[0, 1]$ is covered, and all sensors die after $2/\Delta$ time units. Now suppose that a solution ρ exists with lifetime strictly greater than $2/\Delta$. Hence $\max_i\{\rho_i\} < \Delta/2$. By definition, Δ must equal Δ_0, Δ_1, or the maximum internal gap. If the latter, then there exists a point $u \in [0, 1]$ between the two sensors forming the maximum internal gap that is uncovered. On the other hand, if $\Delta = \Delta_0$, then if $\Delta_0 = 2x_{i_0}$, 0 is uncovered, and otherwise, there is a point in $[0, x_{i_0}]$ that is uncovered. A similar argument holds if $\Delta = \Delta_1$. □

In the next definition we transform x into an instance x' by pushing sensors away from $\frac{1}{2}$, so that each internal gap between sensors is of equal width. See Figure 7 for an illustration.

Definition 1. *For a given instance x, let k be a sensor whose location is closest to $1/2$. Then we define the* stretched *instance x' of x as follows:*

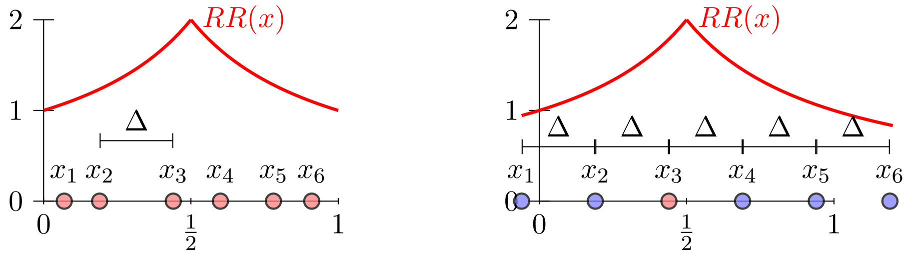
$$x'_i = \begin{cases} (1 - r_k) - (\lceil n/2 \rceil - i)\Delta & i \le \lceil n/2 \rceil, \\ (1 - r_k) + (i - \lceil n/2 \rceil)\Delta & i > \lceil n/2 \rceil . \end{cases}$$

Observation 9. *Let x' be a stretched instance of x. Then $|\left\{i : x'_i \le \frac{1}{2}\right\}| = \left\lceil \frac{n}{2} \right\rceil$ and $|\left\{i : x'_i > \frac{1}{2}\right\}| = \left\lfloor \frac{n}{2} \right\rfloor$.*

Lemma 10. *Let x' be the stretched instance of x. Then, $\textsc{Opt}_0(x') = \textsc{Opt}_0(x)$ and $\mathrm{RR}'(x') \le \mathrm{RR}'(x)$.*

Proof. First, by construction, the internal gaps in x' are of length Δ and $\Delta'_0, \Delta'_1 \le \Delta$. Thus, by Lemma 8, $\textsc{Opt}_0(x') = \textsc{Opt}_0(x)$. By Lemma 7 we know that the sensors moved away from $\frac{1}{2}$, hence $\sum_i r'_i \ge \sum_i r_i$ and $\mathrm{RR}'(x') \le \mathrm{RR}'(x)$. □

Now we are ready to bound $\mathrm{RR}(x)$.



(a) An instance x

(b) The stretched instance x'

Figure 7: Transformation of instance x to stretched instance x'. The sensor closest to $\frac{1}{2}$ (x_3) remains in place, while the other sensors are placed at increasing intervals of Δ away from x_3. The RoundRobin lifetime of a sensor is shown as a continuous function of its location x.

Lemma 11. $\mathrm{RR}'(x) \geq \frac{2}{3}\textsc{Opt}_0(x)$, *for every instance* $I = (x, \mathbf{1})$ *of* RadSC, *where sensors may be located outside* $[0, 1]$.

Proof. By Lemma 10 we may assume that the instance is stretched. First, suppose that n is even. Since x is a stretched instance, it must be the case that exactly half of the sensors lie to the left of $1/2$, and exactly half lie to the right. Hence,

$$\begin{aligned}
\overline{r} \triangleq \frac{1}{n}\sum_{i=1}^{n} r_i &= \frac{1}{n}\left[\sum_{j=0}^{n/2-1}(r_{n/2} + j\Delta) + \sum_{j=0}^{n/2-1}(r_{n/2+1} + j\Delta)\right] \\
&= \frac{1}{n}\left[\frac{n}{2}\cdot r_{n/2} + \Delta\binom{n/2}{2} + \frac{n}{2}\cdot r_{n/2+1} + \Delta\binom{n/2}{2}\right] \\
&= \frac{r_{n/2} + r_{n/2+1}}{2} + \frac{2\Delta}{n}\binom{n/2}{2} \\
&= \frac{1+\Delta}{2} + \frac{\Delta(n-2)}{4} \\
&= \frac{1}{2} + \frac{n\Delta}{4}\ ,
\end{aligned}$$

where we have used the fact that since the sequence is stretched $r_{n/2} + r_{n/2+1} = 1 + \Delta$. Furthermore, since $n\Delta \geq 1$, it now follows that

$$\frac{\mathrm{RR}'(x)}{\textsc{Opt}_0(x)} = \frac{n/\overline{r}}{2/\Delta} = \frac{n\Delta}{1 + n\Delta/2} = \frac{1}{\frac{1}{n\Delta} + \frac{1}{2}} \geq \frac{2}{3}\ .$$

If n is odd, then w.l.o.g. there are $\frac{n+1}{2}$ sensors to the left of 1/2, and $\frac{n-1}{2}$ to the right. Then

$$\begin{aligned}
\overline{r} &= \frac{1}{n}\left[\sum_{j=0}^{(n-1)/2}(r_{(n+1)/2}+j\Delta)+\sum_{j=0}^{(n-3)/2}(r_{(n+3)/2}+j\Delta)\right] \\
&= \frac{1}{n}\left[\frac{n+1}{2}\cdot r_{(n+1)/2}+\Delta\binom{(n+1)/2}{2}+\frac{n-1}{2}\cdot r_{(n+3)/2}+\Delta\binom{(n-1)/2}{2}\right] \\
&= \frac{r_{(n+1)/2}+r_{(n+3)/2}}{2}+\frac{r_{(n+1)/2}-r_{(n+3)/2}}{2n}+\frac{\Delta}{n}\cdot\frac{(n-1)^2}{4} \\
&\leq \frac{1+\Delta}{2}+\frac{\Delta}{n}\cdot\frac{(n-1)^2}{4} \\
&= \frac{1}{2}+\Delta\frac{n^2+1}{4n}\ .
\end{aligned}$$

We have two cases. If $r_1 \geq 1$, then there are $n-1$ gaps of size Δ, as well as one gap of size at most $\Delta/2$. Since the gaps cover the entire interval, we have that $(n-1)\Delta+\frac{\Delta}{2}\geq 1$. It follows that $n\Delta \geq \frac{2n}{2n-1}$. Thus, we can demonstrate the same bound, since

$$\frac{\mathrm{RR}'(x)}{\textsc{Opt}_0(x)}=\frac{n/\overline{r}}{2/\Delta}\geq\frac{n\Delta}{1+\frac{(n^2+1)\Delta}{2n}}=\frac{1}{\frac{1}{n\Delta}+\frac{1}{2}+\frac{1}{2n^2}}\geq\frac{2n^2}{3n^2-n+1}>\frac{2}{3}\ .$$

Finally, we consider the case where $r_1 < 1$. For some $\epsilon \in (0, \Delta/2]$, we can set $r_{(n+1)/2} = \frac{1}{2}+\epsilon$. Since sensors $(n+1)/2$ and $(n+3)/2$ are of distance Δ from one another, it follows that

$$r_{\frac{n+3}{2}}-r_{\frac{n+1}{2}}=(1/2+\Delta-\epsilon)-(1/2+\epsilon)=\Delta-2\epsilon\ .$$

Moreover, we will show that $\epsilon \leq \Delta/4$, and thus $r_{(n+3)/2}-r_{(n+1)/2}\geq \Delta/2$. To see this, note first that it follows from the definition of a stretch sequence and the assumption that $r_1 < 1$ that $r_1 = r_{(n+1)/2}+\Delta(n-1)/2$ and $r_2 = r_{(n+3)/2}-\Delta(n-3)/2$. Hence their difference is

$$r_1-r_n=(r_{\frac{n+1}{2}}+\tfrac{1}{2}\Delta(n-1))-(r_{\frac{n+3}{2}}+\tfrac{1}{2}\Delta(n-3))=r_{\frac{n+1}{2}}-r_{\frac{n+3}{2}}+\Delta=2\epsilon\ .$$

However since $1-\Delta/2 \leq r_n \leq r_1 < 1$, it must be the case that $r_1 - r_n \leq \Delta/2$, and this implies that $\epsilon \leq \Delta/4$.

Finally, a computation similar to the one above reveals that

$$\overline{r}\leq\frac{r_{\frac{n+1}{2}}+r_{\frac{n+3}{2}}}{2}+\frac{r_{\frac{n+1}{2}}-r_{\frac{n+3}{2}}}{2n}+\frac{\Delta}{n}\frac{(n-1)^2}{4}\leq\frac{1+\Delta}{2}-\frac{\Delta}{4n}+\frac{\Delta}{n}\frac{(n-1)^2}{4}=\frac{1}{2}+\frac{n\Delta}{4}\ .$$

As this is the same bound that we obtained in the even case, we similarly achieve the same 2/3 bound. □

4.4 Putting It All Together

It remains only to connect the pieces we have accumulated in the previous three sections.

Lemma 12. $\mathrm{RR}'(x^j, b^j) \geq \frac{2}{3}T_j$, *for every strip* j.

Proof. The result follows immediately from Lemmas 5, 6, and 11. □

Our main result now follows from our construction.

Theorem 3. RoundRobin *is a* $\frac{3}{2}$*-approximation algorithm for* OnceSC*.*

Proof. First, observe that

$$\sum_j \mathrm{RR}(x^j, b^j) = \sum_j \sum_{x_i \in x^j} \frac{b_i^j}{r_i} = \sum_i \frac{1}{r_i} \sum_{j: x_i \in x^j} b_i^j \leq \sum_i \frac{1}{r_i} b_i = \mathrm{RR}(x, b) \ .$$

By Lemmas 3 and 12 we have that

$$\mathrm{RR}(x, b) \geq \sum_j \mathrm{RR}(x^j, b^j) \geq \sum_j \mathrm{RR}'(x^j, b^j) \geq \sum_j \frac{2}{3} T_j = \frac{2}{3} T = \frac{2}{3} \mathrm{Opt}(x, b) \ ,$$

and we are done. □

Theorem 3 readily extends to the Strip Cover problem, assuming all radii are rational and all radii changes are done at rational times.

Theorem 4. RoundRobin *is a* $\frac{3}{2}$*-approximation algorithm for* Strip Cover*.*

5 Duty Cycle Algorithms

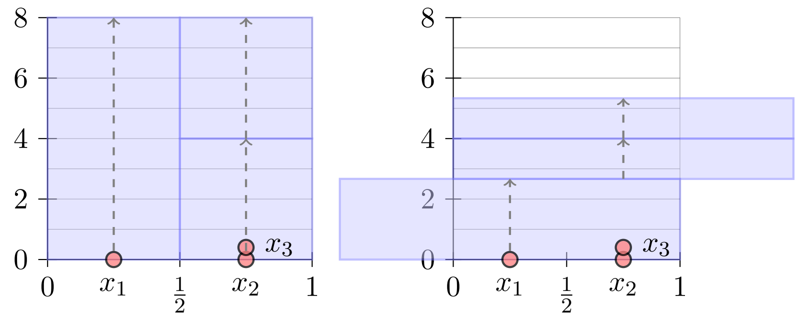
In this paper we analyzed the RoundRobin algorithm in which each sensor works alone. One may consider a more general version of this approach, where a schedule induces a partition of the sensors into sets, or *shifts*, and each shift works by itself. In RoundRobin each shift consists of one active sensor. We refer to such an algorithm as a *duty cycle* algorithm.

In this section we show that, in the worst case, no duty cycle algorithm outperforms RoundRobin. More specifically, we show that the approximation ratio of any duty cycle algorithm is at least $\frac{3}{2}$.

Lemma 13. *The approximation ratio of any duty cycle algorithm is at least* $\frac{3}{2}$ *for both* OnceSC *and* Strip Cover*.*

Proof. Consider an instance where $x = (\frac{1}{4}, \frac{3}{4}, \frac{3}{4})$ and $b = (2, 1, 1)$. An optimal solution is obtained by assigning $\rho_1 = \rho_2 = \rho_3 = \frac{1}{4}$, $\tau_1 = \tau_2 = 0$ and $\tau_3 = 4$. That is, sensor 1 covers the interval $[0, 0.5]$ for 8 time units, sensors 2 covers $[0.5, 1]$ until time 4, and sensors 3 covers $[0.5, 1]$ from time 4 to 8. This solution is optimal in that it achieves the maximum possible lifetime of $8 = 2 \sum_i b_i$.

On the other hand, the best duty cycle algorithm is RoundRobin, which achieves a lifetime of 16/3 time units. (The shifts $\{1, 2\}$ and $\{3\}$ would also result in a lifetime of 16/3 time units.) Both schedules are shown in Figure 8. □



(a) OPT$(x, b) = 8$ (b) ROUNDROBIN$(x, b) = 16/3$

Figure 8: Best schedule vs. best duty cycle schedule. Here $x = (\frac{1}{4}, \frac{3}{4}, \frac{3}{4})$ and $b = (2, 1, 1)$.

6 Set Radius Strip Cover

In this section we present an optimal $O(n^2 \log n)$-time algorithm for the RADSC problem. Recall that in RADSC we may only set the radii of the sensors since all the activation times must be set to 0. More specifically, we assign non-zero radii to a subset of the sensors which we call *active*, while the rest of the sensors get $\rho_i = 0$ and do not participate in the cover.

Given an instance (x, b), a radial assignment ρ is called *proper* if the following conditions hold:

1. Every sensor is either inactive, or exhausts its battery by time T, where T is the lifetime of ρ. That is, $\rho_i \in \{0, b_i/T\}$,
2. No sensor's coverage is superfluous. That is, for every active sensor i there is a point $u_i \in [0, 1]$ such that $u_i \in [x_i - \rho_i, x_i + \rho_i]$ and $u_i \notin [x_k - \rho_k, x_k + \rho_k]$, for every active $k \neq i$.

Lemma 14. *There is a proper optimal assignment for every* RADSC *instance.*

Proof. Let $I = (x, b)$ be a RADSC instance, and let ρ be an optimal assignment for I with lifetime T. We first define the assignment $\rho' = b/T$ and show that it is feasible. Since ρ has lifetime T, any point $u \in [0, 1]$ is covered by some sensor i throughout the time interval $[0, T]$. It follows that $\rho_i \leq b_i/T = \rho'_i$. Hence, $u \in [x_i - \rho'_i, x_i + \rho'_i]$, and thus ρ' has lifetime T. Next, we construct an assignment ρ''. Initially, $\rho'' = \rho'$. Then starting with $i = 1$, we set $\rho''_i = 0$ as long as ρ'' remains feasible. Clearly, $\rho''_i \in \{0, b_i/T\}$. Furthermore, for every sensor i there must be a point $u_i \in [x_i - \rho''_i, x_i + \rho''_i]$ such that $u_i \notin [x_k - \rho''_k, x_k + \rho''_k]$, for every active $k \neq i$, since otherwise i would have been deactivated. Hence, ρ'' is a proper assignment. Moreover, ρ'' has lifetime T and thus it is optimal. □

Given a proper optimal solution, we add two dummy sensors, denoted 0 and $n + 1$, with zero radii and zero batteries at 0 and at 1, respectively. The dummy sensors are considered active. We show that the optimal lifetime of a given instance is determined by at most two active sensors.

Lemma 15. *Let T be the optimal lifetime of a given* RADSC *instance $I = (x, b)$. There exist two sensors $i, k \in \{0, \ldots, n + 1\}$, where $i < k$, such that $T = \frac{b_k + b_i}{x_k - x_i}$.*

Proof. Let ρ be the proper optimal assignment, whose existence is guaranteed by Lemma 14. We claim that there exist two neighboring active sensors i and k, where $i < k$, such that $\rho_i + \rho_k = x_k - x_i$. The lemma follows, since $\rho_i = b_i/T$ and $\rho_k = b_k/T$.

Observe that if $\rho_i + \rho_k < x_k - x_i$, for two neighboring active sensors i and k, then there is a point in the interval (x_i, x_k) that is covered by neither i and k, but is covered by another sensor. This means that either i or k is redundant, in contradiction to ρ being proper. Hence, $\rho_i + \rho_k \geq x_k - x_i$, for every two neighboring active sensors i and k.

Let $\alpha = \min\left\{\frac{\rho_k+\rho_i}{x_k-x_i} : i,k \text{ are active}\right\}$. If $\alpha = 1$, then we are done. Otherwise, we define the assignment $\rho' = \rho/\alpha$. ρ' is feasible since $\rho'_i + \rho'_k = \frac{1}{\alpha}(\rho_i + \rho_k) \geq x_k - x_i$, for every two neighboring active sensors i and k. Furthermore, the lifetime of ρ' is αT, in contradiction to the optimality of ρ. □

Lemma 15 implies that there are $O(n^2)$ possible lifetimes. This leads to an algorithm for solving RadSC.

Theorem 5. *There exists an $O(n^2 \log n)$-time algorithm for solving* RadSC.

Proof. First if $n = 1$, then $\rho_1 \leftarrow r_1 \triangleq \max(x_1, 1 - x_1)$ and we are done. Otherwise, let $T_{ik} \leftarrow \frac{b_k+b_i}{x_k-x_i}$, for every $i, k \in \{0, \ldots, n+1\}$ such that $i < k$. After sorting the set $\{T_{ik} : i < k\}$, perform a binary search to find the largest potentially feasible lifetime. The feasibility of candidate T_{ik} can be checked using the assignment $\rho^{ik}_\ell \leftarrow b_\ell / T_{ik}$, for every sensor ℓ.

There are $O(n^2)$ candidates, each takes $O(1)$ to compute, and sorting takes $O(n^2 \log n)$ time. Checking the feasibility of a candidate takes $O(n)$ time, and thus the binary search takes $O(n \log n)$. Hence, the overall running time is $O(n^2 \log n)$. □

7 Discussion and Open Problems

We have shown that RoundRobin, which is perhaps the simplest possible algorithm, has a tight approximation ratio of $\frac{3}{2}$ for both OnceSC and Strip Cover. We have also shown that both OnceSC and Strip Cover are NP-hard, but it remains to be seen whether the same is true for the special case with unit size batteries. Future work may include finding algorithms with better approximation ratios for either problem. However, we have eliminated duty cycle algorithms as candidates. Observe that both OnceSC and TimeSC are NP-hard, while RadSC can be solved in polynomial time. This suggests that hardness comes from setting the activation times.

We have assumed that the battery charges dissipate in direct inverse proportion to the assigned sensing radius (e.g. $\tau = b/\rho$). It is natural to suppose that an exponent could factor into this relationship, so that, say, the radius drains in quadratic inverse proportion to the sensing radius (e.g. $\tau = b/\rho^2$). One could expand the scope of the problem to higher dimensions. Before moving both the sensor locations and the region being covered to the plane, one might consider moving one but not the other. This yields two different problems: (i) covering the line with sensors located in the plane; and (ii) covering a region of the plane with sensors located on a line.

References

[1] Z. Abrams, A. Goel, and S. A. Plotkin. Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. In *3rd International Symposium on Information Processing in Sensor Networks*, pages 424–432, 2004.

[2] G. Aloupis, J. Cardinal, S. Collette, S. Langerman, D. Orden, and P. Ramos. Decomposition of multiple coverings into more parts. *Discrete & Computational Geometry*, 44(3):706–723, 2010.

[3] A. Bar-Noy and B. Baumer. Average case network lifetime on an interval with adjustable sensing ranges. *Algorithmica*, 72(1):148–166, 2015.

[4] A. Bar-Noy, B. Baumer, and D. Rawitz. Changing of the Guards: Strip Cover with Duty Cycling. In *19th International Colloquium on Structural Information and Communication Complexity*, volume 7355 of *Lecture Notes in Computer Science*, pages 36–47, 2012.

[5] A. Bar-Noy, T. Brown, M. P. Johnson, and O. Liu. Cheap or flexible sensor coverage. In *5th IEEE International Conference on Distributed Computing in Sensor Systems*, volume 5516 of *Lecture Notes in Computer Science*, pages 245–258, 2009.

[6] A. Bar-Noy, D. Rawitz, and P. Terlecky. Maximizing barrier coverage lifetime with mobile sensors. In *21st Annual European Symposium on Algorithms*, volume 8125 of *Lecture Notes in Computer Science*, pages 97–108, 2013.

[7] A. L. Buchsbaum, A. Efrat, S. Jain, S. Venkatasubramanian, and K. Yi. Restricted strip covering and the sensor cover problem. In *18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1056–1063, 2007.

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

[9] M. Cardei and J. Wu. Coverage in wireless sensor networks. In *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, chapter 19, pages 19.1–19.10. CRC press, 2004.

[10] M. Cardei, J. Wu, and M. Lu. Improving network lifetime using sensors with adjustable sensing ranges. *International Journal of Sensor Networks*, 1(1/2):41–49, 2006.

[11] M. Cardei, J. Wu, M. Lu, and M. O. Pervaiz. Maximum network lifetime in wireless sensor networks with adjustable sensing ranges. In *IEEE International Conference on Wireless And Mobile Computing, Networking And Communications*, volume 3, pages 438–445, 2005.

[12] E. W. Chambers, S. P. Fekete, H. Hoffmann, D. Marinakis, J. S. B. Mitchell, V. Srinivasan, U. Stege, and S. Whitesides. Connecting a set of circles with minimum sum of radii. In *12th International Symposium on Algorithms and Data Structures*, volume 6844 of *Lecture Notes in Computer Science*, pages 183–194, 2011.

[13] M. Gibson and K. R. Varadarajan. Optimally decomposing coverings with translates of a convex polygon. *Discrete & Computational Geometry*, 46(2):313–333, 2011.

[14] S. Kumar, T.-H. Lai, and A. Arora. Barrier coverage with wireless sensors. *Wireless Networks*, 13(6):817–834, 2007.

[15] N. Lev-Tov and D. Peleg. Polynomial time approximation schemes for base station coverage with minimum total radii. *Computer Networks*, 47(4):489–501, 2005.

[16] J. Pach. Covering the plane with convex polygons. *Discrete & Computational Geometry*, 1(1):73–81, 1986.

[17] J. Pach and G. Tóth. Decomposition of multiple coverings into many parts. *Computational Geometry: Theory and Applications*, 42(2):127–133, 2009.

[18] M. A. Perillo and W. B. Heinzelman. Optimal sensor management under energy and reliability constraints. In *IEEE Wireless Communications and Networking*, pages 1621–1626, 2003.

[19] A. Saipulla, C. Westphal, B. Liu, and J. Wang. Barrier coverage with line-based deployed mobile sensors. *Ad Hoc Networks*, 11(4):1381–1391, 2013.

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

[21] Y. Taniguchi, T. Kitani, and K. Leibnitz. A uniform airdrop deployment method for large-scale wireless sensor networks. *International Journal of Sensor Networks*, 9(3/4):182–191, 2011.

[22] L. Wang and Y. Xiao. A survey of energy-efficient scheduling mechanisms in sensor networks. *Mobile Networks and Applications*, 11(5):723–740, 2006.