

SENSOR STRIP COVER:
Maximizing Network Lifetime on an Interval

by

Benjamin Baumer

A dissertation submitted to the Graduate Faculty in Mathematics in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York.

2012

©2012

Benjamin Baumer

All Rights Reserved

This manuscript has been read and accepted for the Graduate Faculty in Mathematics in satisfaction of the dissertation requirements for the degree of Doctor of Philosophy.

Amotz Bar-Noy

Date

Chair of Examining Committee

Jozef Dodziuk

Date

Executive Officer

Amotz Bar-Noy

Joseph Malkevitch

Christina Zamfirescu

Supervisory Committee

Abstract

SENSOR STRIP COVER:
Maximizing Network Lifetime on an Interval

by

Benjamin Baumer

Advisor: Amotz Bar-Noy

Suppose that n sensors are deployed on a one-dimensional region (a *strip*, or *interval*) that we wish to cover with a wireless sensor network. Each sensor is equipped with a finite battery, and has an adjustable sensing range, which we control. If each sensor's battery drains in inverse linear proportion to its sensing radius, which schedule will maximize the lifetime of the resulting network? We study this SENSOR STRIP COVER problem and several related variants. For the general SENSOR STRIP COVER problem, we analyze performance in both the worst-case and average-case for several algorithms, and show that the simplest algorithm, in which the sensors take turns covering the entire line, has a tight $\frac{3}{2}$ -approximation ratio. Moreover, we demonstrate a more sophisticated algorithm that achieves an expected lifetime of within

12% of the theoretical maximum against uniform random deployment of the sensors. We show that if the sensing radii can be set only once, then the resulting SET ONCE STRIP COVER problem is NP-hard. However, if all sensors *must* be activated immediately, then we provide a polynomial time algorithm for the resulting SET RADIUS STRIP COVER problem. Finally, we consider the imposition of a *duty cycling* restriction, which forces disjoint subsets of the sensors (called *shifts*) to act in concert to cover the entire interval. We provide a polynomial-time solution for the case in which each shift contains at most two sensors. For shifts of size k , we provide worst-case and average-case analysis for the performance of several algorithms.

Acknowledgements

This dissertation has come to fruition due to the tireless, yet considerate efforts of Amotz Bar-Noy, to whom I am forever indebted. Although part of what brought us together may have been bad luck, his patience, guidance, work ethic, and astounding capacity for multi-tasking have made this work possible. I cannot imagine having a more supportive mentor.

No person completes a doctorate without help, but my case suggests that you may need a lot of it! My co-authors, particularly Dror Rawitz and Prithwish Basu, have sharpened my research with their insightfulness and professionalism. My professors, particularly Joe Malkevitch, the late Gary Bloom, and the incomparable János Pach, have both shown me the depths of what I have to learn, and helped me dive deeper. My colleagues, first and foremost Peter Terlecky, but also including Brent Cody, Chris Seaman, Valia Mitsou, and the irrepressible Subir Dhamoon, have helped make this graduate experience both successful and enjoyable. I would be remiss without

recognizing Adriano Garsia, David Reynolds, Brian Miceli, Jared Clorfeine, Michael Donohue, Jason Lee, and Cameron Parker of the University of California at San Diego for the same. My co-workers with the New York Mets, particularly Omar Minaya, John Ricco, Adam Fisher, and Jon Miller, have offered me unusual flexibility and support.

Finally, the personal qualities that have brought me to this achievement have been the joint work of a slew of long-term contributors. My friends, particularly Frank Philbrick, Roger Taylor, Jeff Zimbalist, Thomas Smith, JP Craford, and Joe Randazzo, have suffered through shared intellectual ruminations far longer and less interesting than this one, yet equally rewarding, and certainly dearer to my heart. Their sacrifices have not been overlooked. My family, particularly Donald Baumer, Polly Baumer, and Maggie Baumer, have given me more than you would think would be possible, and asked nothing in return – not even to know, “when are you going to graduate?” And of course, the largest helping of gratitude is reserved (with good cause) for my spectacular wife, Cory Mescon, as it always will be.

Thank you all.

Contents

1	Introduction	1
1.1	Motivation	2
1.2	Problems	3
1.3	Related work	4
1.3.1	Setting activation times	5
1.3.2	Setting radii	6
1.3.3	Duty cycling	7
1.3.4	Adjustable ranges	9
1.4	Contribution	10
2	Preliminaries	13
2.1	Definitions	13
2.2	Problems and solutions	14
2.3	Basic algorithms	19
2.4	Preliminary results	21

<i>CONTENTS</i>	ix
3 Set Radius Strip Cover	26
3.1 Polynomial Time Algorithm	26
4 Set Once Strip Cover	30
4.1 Relationship to Other Problems	30
4.1.1 Computational Complexity	31
4.1.2 Performance gap with SENSOR STRIP COVER	33
4.1.3 Duty cycle algorithms	34
4.2 Worst-case analysis of ROUNDROBIN	36
4.2.1 Cutting the Schedule into Strips	37
4.2.2 Reduction to RADSC with Uniform Batteries	40
4.2.3 Analysis of Round Robin for Unit Batteries	42
5 Sensor Strip Cover	49
5.1 Worst-case analysis	49
5.1.1 ROUNDROBIN	50
5.1.2 GREEDY	52
5.2 Average-case analysis	54
5.2.1 ROUNDROBIN	54
5.2.2 k -ROUNDROBIN	55
5.2.3 log-ROUNDROBIN	58

<i>CONTENTS</i>	x
5.2.4 Optimizations of log-ROUNDROBIN	64
6 <i>k</i>-Duty Cycle Strip Cover	67
6.1 2-DUTYSC	67
6.2 Worst-case analysis	69
6.2.1 ROUNDROBIN	69
6.2.2 Round Robin vs. All	69
6.2.3 MATCH	77
6.3 Average-case analysis	81
6.3.1 SWEEP and NEST	81
6.3.2 Fault Tolerance	85
7 Conclusion	88
7.1 Summary	88
7.2 Open Problems	89
7.2.1 Non-uniform batteries	89
7.2.2 Uniform batteries	90
7.2.3 Variations	91
Bibliography	93

List of Tables

1.1	Summary of results for SENSOR STRIP COVER with unit batteries	11
1.2	Summary of results for k -DUTYSC	12
5.1	Numerical approximations for optimized log-ROUNDROBIN	66

List of Figures

1.1	Relationships between problem variants	4
2.1	Graphical depictions of valid schedules for a randomly generated instance of 5 unit-battery sensors	16
2.2	Graphical depiction of RADSC schedule for a randomly generated instance of 5 unit-battery sensors	17
2.3	Graphical depictions of duty cycle and non-duty cycle schedules.	19
2.4	Illustration of the gaps in ALL	20
2.5	Illustration of the gaps in X_6^*	24
2.6	Graphical comparison of algorithms on X_8^*	25
4.1	Example illustrating that ONCESC is NP-hard	32
4.2	Performance gap between ONCESC and SENSOR STRIP COVER	35
4.3	Best ONCESC schedule vs. best duty cycle schedule	36
4.4	Cutting an optimal schedule into strips	38

4.5	Reduction of non-uniform battery instance to uniform battery instance	41
4.6	Transformation of instance to stretched instance	45
5.1	Depiction of $\frac{3}{2}$ lower bound for ROUNDROBIN	51
5.2	Depiction of $\frac{6}{5}$ lower bound for GREEDY	53
5.3	k -ROUNDROBIN for $k = 2, 3$ against a random instance.	57
5.4	log-ROUNDROBIN with depth 2 and 3 against a random instance.	60
5.5	Normalized sensor network lifetime for $k = 1, 2, 3, 4$ using the log-ROUNDROBIN algorithm.	61
6.1	Tabular and graphical representation of small values of γ_k^*	77
6.2	Depiction of $\frac{15}{11}$ lower bound for MATCH	79
6.3	Performance of ROUNDROBIN, MATCH, SWEEP, and NEST compared to ALL on perfect deployments	82
6.4	Comparison of density estimates of the per sensor network lifetime for various algorithms in 4-DUTYSC	84

Chapter 1

Introduction

Suppose that we wish to cover a one-dimensional region (or interval) with a wireless sensor network. Given the locations and initial battery charges of n sensors deployed on that interval, we have the ability to set the sensing radius of each sensor at any point in time to an arbitrary length. However, the battery of each sensor drains in inverse linear proportion to its sensing radius, so that the larger we make the radius, the faster the battery drains. Our goal is to maximize the *lifetime* of the network, which is the length of time that the entire interval is covered until all of the batteries run out. What schedule will achieve this? In this dissertation, we address this question by analyzing several algorithms applicable to this problem and a few variations of it.

In the most general version of the problem, which we call `SENSOR STRIP`

COVER¹, the initial battery charges are not necessarily the same, the scheduler is free to change the radius of any sensor finitely many times, and there are no restrictions on how many sensors may work together at one time or over the lifetime of the network. However, we also consider variations where all initial battery charges are the same, where the scheduler may set the radius of each sensor only once, and where a *duty cycling* restriction is imposed, wherein the sensors are grouped into disjoint *shifts* which take turns covering the interval. Each of these related variations of the SENSOR STRIP COVER problem are of independent interest.

1.1 Motivation

This type of scheduling problem arises in many applications, often in relation to problems of *barrier coverage* (see [13, 35] for surveys). 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 the line by, say, dropping them from an airplane flying overhead (e.g. [12, 31, 33]). Once deployed, the sensors send

¹We use the terms *strip* and *interval* interchangeably to refer to the one-dimensional coverage region.

us their locations via GPS, which we then analyze and respond with a list of coverage assignments. Replacing the battery in any sensor is infeasible. How do we construct a set of assignments that will keep this vital supply line completely monitored for as long as possible?

1.2 Problems

SENSOR STRIP COVER is the most general version of the problem, where the only significant restriction is that the radius of each sensor can be set only finitely many times. A solution to the problem, which we call a *schedule*, consists of a finite list, each element of which specifies the sensing radius, activation time, and deactivation time of a particular sensor. Note that such a schedule may address a particular sensor finitely many times, turning it on or off.

If, on the other hand, each radius can be set only once, then we call the resulting problem SET ONCE STRIP COVER (ONCESC). In this case a solution consists of a single radius and activation time pair for each sensor. Two subproblems of ONCESC are also of interest. If every sensor must be activated immediately, then we call the resulting problem of determining the optimal radii SET RADIUS STRIP COVER (RADSC). Conversely, if the radii are fixed (and given), then we call the problem of finding the optimal

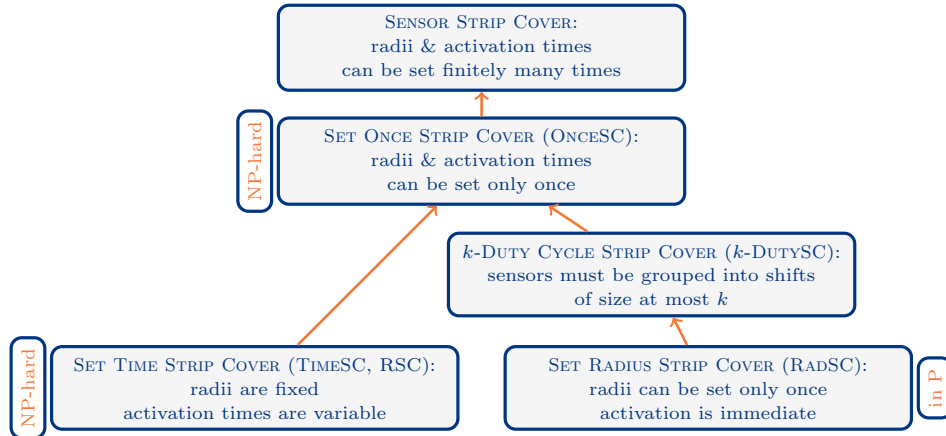


Figure 1.1: Relationships between problem variants. In TIMESC, the radii are given and the problem is to schedule the activation times, while in RADSC, the activation times are immediate, and the problem is to assign the radii. In ONCESC, *both* the radii and the activation times must be set. RADSC can be seen as a “one-shift” variant of n -DUTYSC. Note that k -DUTYSC can be solved in polynomial time for $k \leq 2$.

activation times SET TIME STRIP COVER (TIMESC).

Finally, if the sensors must be grouped into shifts of size at most k that take turns covering the interval, then we call the resulting problem k -DUTY CYCLE STRIP COVER (k -DUTYSC). Figure 1.1 summarizes the important differences between these problems and their relationships with one another.

1.3 Related work

The literature on wireless sensor networks is vast, and we cannot hope to present an exhaustive list of references here. Cardei and Wu [13] and Wang and Xiao [35] have written broader surveys relevant to this material.

Our discussion of work related to `SENSOR STRIP COVER` is organized into four sections: 1) problems that focus on setting sensor activation times; 2) problems that focus on setting sensor radii; and 3) papers that pursue a duty cycling approach for a sensor cover problem; 4) problems in which adjustable range sensors are employed.

1.3.1 Setting activation times

Buchsbaum, et al. [11] initiated this line of research by defining `TIMESC` as a special case of a more general sensor cover problem. In `TIMESC` (which they called `RESTRICTED STRIP COVER (RSC)`), the locations, sensing radii, and battery charges of n sensors placed on an interval are given, and the problem is to compute an optimal set of activation times, so as to maximize the network lifetime. By taking advantage of the packing-covering duality, and using a reduction from `DYNAMIC STORAGE ALLOCATION`, they showed that this problem is NP-hard. They presented an $O(\log \log n)$ -approximation algorithm, which was later improved to a constant factor approximation algorithm by Gibson and Varadarajan [21]. For the special case of `TIMESC` in which the battery charges are all the same, Buchsbaum, et al. presented a simple greedy algorithm that yields an optimal solution in polynomial time. As in `ONCESC`, they considered only the non-preemptive case, in which a

sensor cannot be reactivated once it is turned on.

1.3.2 Setting radii

The problem of finding the optimal set of radial assignments for sensors deployed on an interval, rather than the activation times, is more tractable. Whereas RADSC requires *area* coverage (i.e. a requirement to cover *all* points in an interval), Lev-Tov and Peleg [25] studied a similar problem with *target* coverage (i.e. a requirement to cover only a finite number of specific points in an interval). In MINIMUM SUM OF RADII COVER (MSRC), the input is a set of n sensors and a finite set of m points on the interval that are to be covered, and the goal is to find the radial assignments with the minimum sum of radii, such that all m points are covered. They used dynamic programming to devise a polynomial time algorithm for the one-dimensional case, and a polynomial time approximation scheme for the two-dimensional case. The running time of their algorithm was later improved from $O((n + m)^3)$ to the optimal $O(n + m)$, most recently by Bar-Noy, et al. [8]. Note that the emphasis here is on energy conservation (through minimizing the sum of radii), and not maximizing lifetime explicitly, since they seek only a single coverage assignment. This is akin to a “one-shift” solution.

Bilò et al. [10] generalized MSRC to cases where the battery draining rate α is not linear with respect to the radius, and showed NP-hardness for the case where $\alpha \geq 2$. Alt et al [3] found constant factor approximation algorithms and extended the hardness result to $\alpha > 1$. Gibson, et al. found a polynomial time solution to a related clustering problem using pinned disks [20].

The 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. [16]. They showed that the best one circle solution to CRA yields a $\frac{3}{2}$ -approximation guarantee, and in fact, the instance that produces their lower bound is simply a translation of the instance we use below in Obs. 5.1 to their problem. Moreover, they presented approximation bounds for solutions using a fixed number of circles, an idea which is similar to limiting shift sizes, as we do in k -DUTYSC. In the case where the radii are bounded, and the tree connecting the points in the plane is unknown, CRA is NP-hard.

1.3.3 Duty cycling

Interest in *duty cycling* developed in part from the introduction of the SET k -COVER problem by Slijepcevic and Potkonjak [32]. This problem, which

they showed to be NP-hard, seeks to find at least k disjoint covers among a set of subsets of a base set. In this context, maximizing the number of covers k serves as a proxy for maximizing the actual network lifetime. Perillo and Heinzelman [29] considered a variation in which each sensor has multiple modes. They translated the problem into a generalized maximum flow graph problem, and employed linear programming to find an optimal solution. Abrams, et al. [1] provided approximation algorithms for a modification of the problem in which the objective was to maximize the total area covered by the sensors. Cardei et al. [12, 14, 15] considered adjustable range sensors, but also sought to maximize the number of non-disjoint set covers over a set of target coverage points.

This notion of *duty cycling* as a means to maximize network lifetime has also appeared in the literature of discrete geometry. Pach [27] began the study of decomposability of multiple coverings by showing that for any centrally symmetric convex polygon P , and any positive integer r , there exists a constant k such that a k -fold covering of the plane with translates of P can be decomposed into r disjoint covers. Pach and Tóth [28] subsequently showed that $r = \Omega(\sqrt{k})$. This result was later improved to the optimal $\Omega(k)$ covers by Aloupis et al. [2], while Gibson and Varadarajan [21] showed the same result without the centrally-symmetric restriction. In each of the

above cases, the concept of finding many disjoint set covers, which can be seen as shifts, is used as a proxy for maximizing network lifetime, but the true lifetime is not addressed directly.

1.3.4 Adjustable ranges

Working the plane, Wu and Yang [37] introduced the notion of networks that use sensors with adjustable ranges, and studied energy consumption under a random deployment model. Cardei and Du [12] proposed the DISJOINT SET COVERS problem, in which n sensors monitor m target points, and the goal is find the maximum number of disjoint covers. They showed that this problem is NP-complete, and that any polynomial-time algorithm is at best a 2 approximation. Heuristics are presented for solving a version of the problem that has been converted into a maximum-flow problem. In subsequent work, Cardei et al. [14, 15] extended this problem to the ADJUSTABLE RANGE SET COVER problem by lifting the restriction that the covers be disjoint. An important difference between our work and theirs is that they assumed that the sensing ranges came from a finite discrete set, rather than a continuous range, as we allow.

Unlike the duty cycling approach taken by Cardei et al., Berman et al. [9, 18] sought to maximize the true lifetime in the SENSOR NETWORK LIFETIME

PROBLEM (SNLP), which seeks to cover m target points with n adjustable range sensors. They propose a Linear Program that is exponential in n , but achieve an $O(\log n)$ -approximation by using the Garg-Könemann algorithm for solving LPs. Sensing ranges are allowed to vary continuously up to a maximum cutoff distance. SENSOR STRIP COVER differs from SNLP in that it requires area coverage on a one-dimensional region.

1.4 Contribution

We introduce the SENSOR STRIP COVER problem, which is the first to consider the true lifetime for area coverage on an interval with adjustable sensing ranges. In Chapter 5, we provide several linear-time approximation algorithms for this problem, ranging in complexity from the most simple to more sophisticated. For each, we provide worst-case and average-case analysis under the assumption that the sensors are distributed uniformly at random about the interval. Since, as we will show, a constant factor approximation is trivial, most of our efforts are focused on reducing the approximation ratio in the *average* case, which is likely to be of greater value in an application scenario. Our main result is a demonstration of a linear-time algorithm whose expected lifetime is within 12% of the theoretical maximum. A summary of our results for SENSOR STRIP COVER is shown in Table 1.1.

Algorithm	$\mathbb{E}[T]$	$Var[T]$	AC	WC_{lb}	WC_{ub}
ROUNDROBIN	1.386	0.078	1.443	1.5	1.5
k -ROUNDROBIN	1.386	0.078	1.443	1.5	1.5
log-ROUNDROBIN	1.738	0.022	1.151	1.5	
Optimized log-ROUNDROBIN	1.791		1.116	1.5	
GREEDY				1.2	

Table 1.1: Summary of results for SENSOR STRIP COVER with unit batteries. $\mathbb{E}[T]$ is the expected lifetime per sensor, if the sensors are deployed uniformly at random. AC is an upper bound on the approximation ratio in the average case, while WC_{lb} and WC_{ub} are lower and upper bounds for the worst-case approximation ratios, respectively.

In Chapter 4, we introduce the ONCESC problem, in which the scheduler does not have the ability to vary a sensor’s radius once it has been activated. We show that ONCESC is NP-hard, and that ROUNDROBIN is a $\frac{3}{2}$ -approximation algorithm for both ONCESC and SENSOR STRIP COVER. Moreover, we show that if the batteries are not uniform, then the larger class of duty cycle algorithms cannot improve on this $\frac{3}{2}$ guarantee. Our analysis of ROUNDROBIN is based on comparing its performance to the RADSC optimum of certain instances with unit batteries. We provide an $O(n^2 \log n)$ -time algorithm for RADSC, in which all sensors must be activated immediately, in Chapter 3.

In Chapter 6, we consider the class of k -DUTYSC problems. While ROUNDROBIN trivially solves 1-DUTYSC, we present a polynomial-time algorithm, which we call MATCH, that solves 2-DUTYSC exactly. We then

Algorithm	$\mathbb{E}[T]$	$Var[T]$	AC	WC_{lb}	WC_{ub}
ROUNDROBIN	1.386	0.078	1.443	1.5	1.5
NEST	1.556		1.285	1.364	1.458
SWEEP	1.575		1.270	1.364	1.458
MATCH	<i>1.632</i>		<i>1.225</i>	1.364	1.458

Table 1.2: Summary of results for k -DUTYSC, for $k \geq 4$. Columns are as in Table 1.1. Numbers in italics are obtained experimentally.

show that MATCH is a $\frac{35}{24}$ -approximation algorithm for k -DUTYSC. We also give two lower bounds: $\frac{15}{11}$, for $k \geq 4$, and $\frac{6}{5}$, for $k = 3$, and provide experimental evidence suggesting that these lower bounds are tight. A critical component of this analysis is an understanding of the performance of ROUNDROBIN to an algorithm that uses only a single shift. We prove that when the sensors are equi-spaced on the coverage interval, ROUNDROBIN performs most poorly in comparison to the one shift algorithm, and we study the performance of ROUNDROBIN against these “perfect” deployments. These results are used to analyze MATCH in k -DUTYSC, but are of independent interest, since perfect deployments are in many ways the most natural. We also find upper bounds on the approximation ratio of MATCH in the average-case, and consider a fault tolerance model, under which MATCH becomes optimal if the failure rate of each sensor is sufficiently high. A summary of our results for k -DUTYSC is shown in Table 1.2.

Chapter 2

Preliminaries

In this chapter we codify our notation, define each problem precisely, and give some preliminary results.

2.1 Definitions

Let $U = [0, 1]$ be the interval that we wish to cover. We assume that we are given as input a vector $X = (x_1, \dots, x_n) \in U^n$ of n sensor locations, and a corresponding vector $B = (b_1, \dots, b_n) \in \mathbb{Q}_+^n$ of battery charges, with $b_i \geq 0$ for all i . For convenience, we assume that $x_i \leq x_{i+1}$ for every $i \in \{1, \dots, n-1\}$. Occasionally, we will abuse notation by treating X or B as a set. Let $B^* \triangleq \sum_i b_i$ denote the total battery charge of the system.

2.2 Problems and solutions

An instance of SENSOR STRIP COVER thus consists of a pair $I = (X, B)$, and a solution is an assignment of radii, activation times, and deactivation times to sensors. More specifically a solution (or *schedule*) is a finite list S of coverage assignments (i, ρ, τ, σ) , each of which tells the i^{th} sensor to activate at time τ with a *radius* of ρ , and maintain that coverage until time σ . For a list of length m , let S_j indicate the j^{th} coverage assignment for $1 \leq j \leq m$.

Any schedule can be visualized by a space-time diagram in which each coverage assignment is 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_j} and assigned the radius ρ_j beginning at time τ_j and ending at time σ_j is depicted by a rectangle with lower-left corner $(x_{i_j} - \rho_j, \tau_j)$ and upper-right corner $(x_{i_j} + \rho_j, \sigma_j)$. Let the set of all points contained in this rectangle be denoted as $Rect(S_j)$. A point $(u, t) \in U \times [0, \infty)$ in space-time is *covered* by a schedule S if $(u, t) \in \bigcup_j Rect(S_j)$.

In addition to the *coverage* constraint described above, a solution to SENSOR STRIP COVER must also satisfy a *battery* constraint. That is, no sensor i can consume more than b_i battery units. During the j^{th} coverage assignment,

sensor i_j consumes $\rho_j \cdot (\sigma_j - \tau_j)$ battery units. Thus, a valid schedule must have $\sum_j \rho_j \cdot (\sigma_j - \tau_j) \leq b_i$ for all $1 \leq i \leq n$.

The *lifetime* of the network in a solution S is the maximum value T such that every point $(u, t) \in U \times [0, T]$ is covered, and no sensor i has consumed more than b_i battery units.

Example. For example, consider the randomly generated instance for which $X = (0.178, 0.275, 0.417, 0.532, 0.737)$ and $B = (1, 1, 1, 1, 1)$. Graphical depictions of two valid schedules are shown in Figure 2.1. The horizontal axis shows the coverage interval U extending from 0 to 1. The vertical axis illustrates time, beginning at 0 and extending until $2B^* = 10$ time units have elapsed¹. The location of each sensor is indicated by a red dot on the horizontal axis, and coverage assignments are depicted by blue rectangles, with a vertical dashed arrow indicating the sensor providing that coverage.

We denote the optimal solution by $\text{OPT}(X, B)$, although we sometimes use OPT , when the instance is clear from the context. If for any instance (X, B) , the lifetime returned by the algorithm ALG is at least $\frac{1}{r} \cdot \text{OPT}(X, B)$, for some $r \geq 1$, then r is an upper bound on the approximation ratio of ALG , and we say that ALG is an r -approximation algorithm. A lower bound for

¹We will show in Lemma 2.1 that $2B^*$ is the maximum possible lifetime.

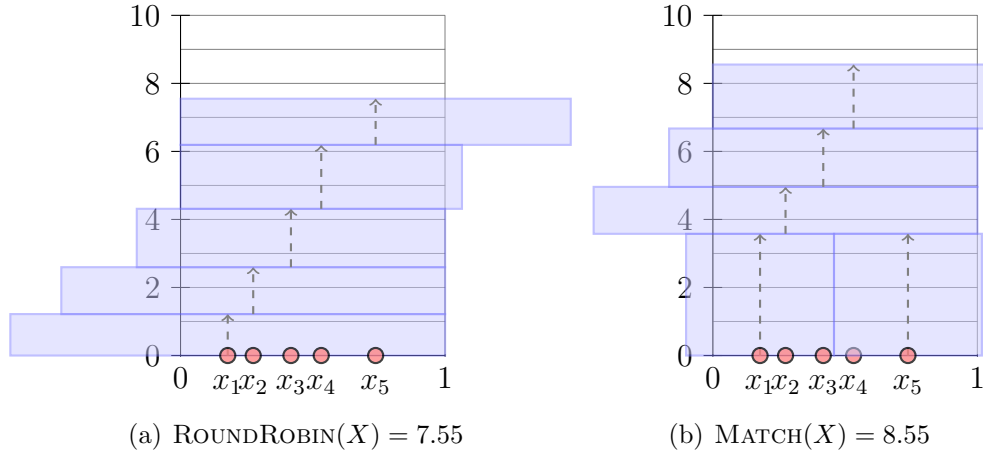


Figure 2.1: Graphical depictions of valid schedules for a randomly generated instance of 5 unit-battery sensors. Here, $X = (0.178, 0.275, 0.417, 0.532, 0.737)$. At left, the `ROUNDROBIN` algorithm produces a lifetime of 7.55 time units, while on the right, the `MATCH` algorithm produces a lifetime of 8.55 time units. Neither achieves the maximum theoretical lifetime of 10 time units, although in this case `MATCH` is optimal.

r can be found by identifying one such instance for which $r \cdot \text{ALG}(X, B) \leq \text{OPT}(X, B)$.

Variants. In `SET ONCE STRIP COVER (ONCESC)`, the radius of each sensor can be set only once. Consequently, while the input remains a pair $I = (X, B)$, a solution can be written as $S = (\rho, \tau)$, where ρ and τ are vectors of length n that indicate the radius and activation time of each of the n sensors. Since there is nothing to be gained by saving battery life for a subsequent coverage assignment, it is implied that each sensor remains active until its battery is fully depleted. Thus, sensor i becomes active at time τ_i ,

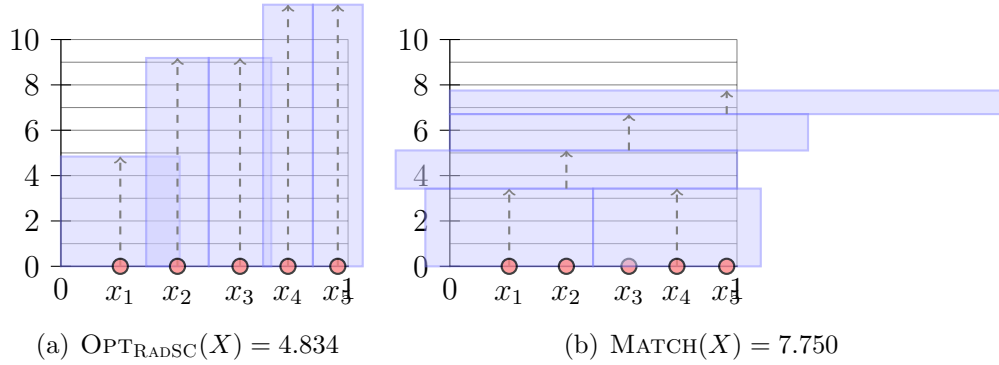


Figure 2.2: Graphical depiction of RADSC schedule for a randomly generated instance of 5 unit-battery sensors. Note that the lifetime achieved in RADSC at left is only 4.84 due to the large radius required on the left-most sensor. Conversely, by allowing multiple coverage groups MATCH achieves a lifetime of 7.75.

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. The goal remains to find the set of n pairs $S = (\rho, \tau)$ that maximizes the lifetime T .

The SET RADIUS STRIP COVER (RADSC) problem is a variant of ON-CESSC in which $\tau = 0^n$. Hence, a solution is simply a radial assignment ρ . A depiction of an optimal RADSC schedule is shown in Figure 2.2. 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. Thus, in graphical treatments of TIMESC, the coverage rectangles are given, and the problem is to position them vertically so as to maximum network lifetime.

Duty cycling. In the *duty cycling* model, we first assume that all sensors have unit capacity batteries. We will justify this assumption later.

A pair (C, t) , where $C \subseteq X$ is a subset of k sensor locations and $t \geq 0$, is called a *k-duty cycle* (or simply a *duty cycle*, or a *shift*). The sensors in C are activated at the same time and are deactivated together after t time units. A duty cycle (C, t) is feasible if the sensors in C can cover the interval U for the duration of t time units. More specifically, a sensor i such that $x_i \in C$ is assigned a radius $1/t$ and covers the range $[x_i - 1/t, x_i + 1/t]$, and the duty cycle is feasible if $U \subseteq \bigcup_{i \in C} [x_i - 1/t, x_i + 1/t]$.

We will show below that given a subset C of k sensors, the optimal t can be computed in linear time by the ALL algorithm. Thus, a solution to the *k-DUTY CYCLE STRIP COVER* (*k-DUTYSC*) consists of a partition of X into m non-empty pairwise disjoint subsets $C_1, \dots, C_m \subseteq X$ such that $|C_j| \leq k$, for every j . The goal is to find a solution that maximizes $\sum_j \text{ALL}(C_j)$.

The optimal lifetime for *k-DUTYSC* is denoted by OPT_k . An illustration of the differences between duty cycle solutions and non-duty cycles solutions is shown in Figure 2.3.

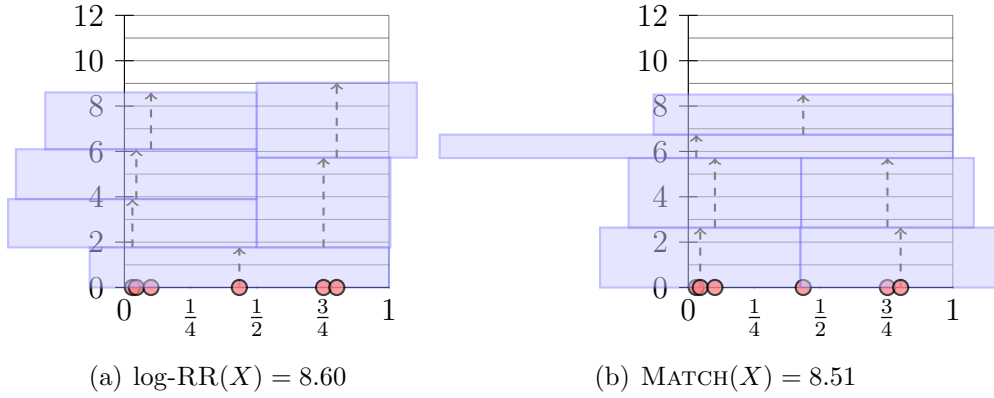


Figure 2.3: Graphical depictions of duty cycle and non-duty cycle schedules. Even for unit-battery instances, the duty cycling restriction (at right), can impose a penalty relative to the general model (at left).

2.3 Basic algorithms

Much of our analysis is predicated on two particularly simple algorithms:

ROUNDROBIN and ALL.

The ROUNDROBIN algorithm forces the sensors to take turns covering U .

Namely it assigns, for every i ,

$$\rho_i = r_i \triangleq \max\{x_i, 1 - x_i\}, \quad \tau_i = \sum_{j=1}^{i-1} b_j / \rho_j . \quad (2.1)$$

The lifetime of ROUNDROBIN is thus

$$\text{RR}(X, B) \triangleq \sum_{i=1}^n \frac{b_i}{r_i} .$$

Note that ROUNDROBIN gives a valid solution for *every* variant of the SENSOR STRIP COVER problem. In particular, ROUNDROBIN is by definition an

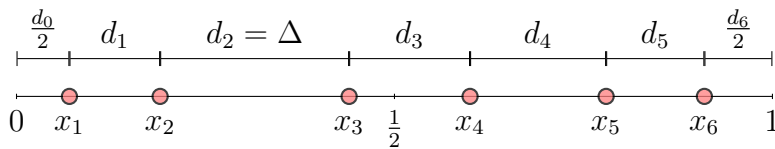


Figure 2.4: Illustration of the gaps in ALL. Note that the gaps neighboring the endpoints are half as large as the corresponding value of d (i.e. d_0 and d_n). Δ is defined as the largest among the d_i 's.

optimal solution to 1-DUTYSC, since each sensor must work alone, therefore there is only one possible solution: $C_i = \{i\}$, for every i .

Conversely, suppose that a subset $C \subseteq X$ of the sensors *must* be activated, and that all batteries are uniform. Then we let $\text{ALL}(C)$ denote the maximum t for which (C, t) is a feasible shift. $\text{ALL}(C)$ is called the *lifetime* of C . Given a duty cycle $C = \{x_{i_1}, \dots, x_{i_k}\}$ we define

$$d_j \triangleq \begin{cases} 2x_{i_1} & j = 0, \\ 2(1 - x_{i_k}) & j = k, \\ x_{i_{j+1}} - x_{i_j} & \text{otherwise,} \end{cases}$$

and $\Delta \triangleq \max_j \{d_j\}$. An illustration of this definition is shown in Figure 2.4.

In addition to ROUNDROBIN and ALL, we develop and discuss several algorithms listed in Tables 1.1 and 1.2. k -ROUNDROBIN and log-ROUNDROBIN are extensions of ROUNDROBIN we develop for SENSOR STRIP COVER and discuss in Chapter 5, while MATCH, SWEEP and NEST are duty cycle algorithms we discuss in Chapter 6. Graphical depictions of schedules for ROUNDROBIN, ALL, log-ROUNDROBIN, and MATCH are show below in Fig-

ure 2.6.

2.4 Preliminary results

Maximum lifetime. The best possible lifetime of any instance is constrained by the total battery charge in the system, as well as the location of the sensors.

Lemma 2.1. *The maximum lifetime of an instance (X, B) is at most $2B^*$.*

Proof. Consider an optimal solution S for (X, B) with lifetime T . Due to the battery constraint, the total area of space-time covered by sensor i is at most $2b_i$, for all $1 \leq i \leq n$, since the width of each rectangle is twice the assigned radius. Due to the coverage constraint, the lifetime T of the network is at most the total area of space-time covered by the sensors, which is at most $2 \sum_i b_i = 2B^*$. \square

Corollary 2.2. *The maximum lifetime of a unit battery instance X is at most $2n$.*

Remark 2.3. *Note that the result of Lemma 2.1 applies even to the case where there may be infinitely many coverage assignments.*

We restrict our attention to algorithms that do not produce superfluous coverage. That is, we only consider algorithms that do not use a larger

radius when a smaller one will suffice. In light of this, a constant factor approximation bound is immediate.

Observation 2.4. Any algorithm for the SENSOR STRIP COVER problem achieves at least a 2-approximation.

Proof. Since the length of U is 1, there is never a reason to assign a radius of more than 1 to any sensor. But the algorithm that assigns a radius of exactly 1 to every sensor and forces each to cover all of U them in succession achieves a lifetime of B^* . By Obs. 2.1, this is at least half of $\text{OPT}(X, B)$. \square

Given an instance (X, B) , let $\bar{r} = \sum_i \frac{b_i}{B^*} \cdot r_i$ be the average of the r_i 's, weighted by their respective battery charge. We define the following lower bound on $\text{RR}(X, B)$:

$$\text{RR}'(X, B) \triangleq \frac{B^*}{\bar{r}} .$$

Lemma 2.5. $\text{RR}'(X, B) \leq \text{RR}(X, B)$, for every instance (X, B) .

Proof. We have that

$$\begin{aligned} \text{RR}(X, B) &= \sum_{i=1}^n \frac{b_i}{r_i} = \sum_{i=1}^n \frac{b_i^2}{b_i r_i} \\ &\geq \frac{(\sum_{i=1}^n b_i)^2}{\sum_{i=1}^n b_i r_i} = \frac{B^*}{\frac{1}{B^*} \sum_{i=1}^n b_i r_i} = \frac{B^*}{\bar{r}} \\ &= \text{RR}'(x, b) , \end{aligned}$$

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

□

Observation 2.6. $\text{ALL}(C) = \frac{2}{\Delta}$.

Proof. Since Δ is by definition the size of the largest gap between any neighboring sensors, or between the left-most sensor and 0 or the right-most sensor and 1, the radial assignment $r_i = \Delta/2$ ensures coverage of $[0, 1]$. This assignment results in a lifetime of $2/\Delta$, and thus, the maximum lifetime of C is at least $2/\Delta$. So suppose that $\text{ALL}(C) > 2/\Delta$. Then $r_i < \Delta/2$, for every i . But by construction, there exists a gap that is exactly Δ , and thus, if all r_i 's are less than $\Delta/2$, this gap is not covered, a contradiction. □

In light of Observation 2.6, it suffices to refer to any subset $C \subseteq X$ as a shift, with a corresponding lifetime that is inferred from $\text{ALL}(C)$. Note that if all sensors have batteries with capacity β , then for any shift C , $\text{ALL}(C)$, and hence the maximum lifetime T , are multiplied by a factor of β . Hence, we assume that all sensors in k -DUTYSC have unit capacity batteries.

Perfect deployment. We define

$$X_n^* = \left\{ \frac{2i-1}{2n} : i \in \{1, \dots, n\} \right\} = \left\{ \frac{1}{2n}, \frac{3}{2n}, \dots, \frac{2n-1}{2n} \right\}.$$

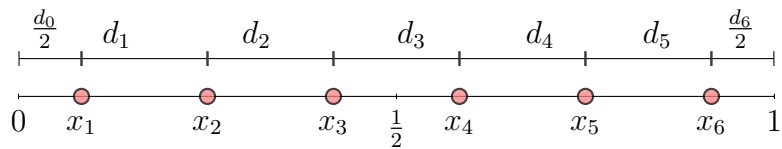


Figure 2.5: Illustration of the gaps in X_6^* . Note that *all* gaps correspond to Δ .

We refer to X_n^* as the *perfect deployment* since the lifetime of X_n^* is $2n$ in any of the unit battery problems we consider. In particular $\text{ALL}(X_n^*) = 2n$. In Figure 2.5, we illustrate the size of the gaps in X_6^* , and in Figure 2.6, we show graphical depictions of schedules for four algorithms against X_8^* .

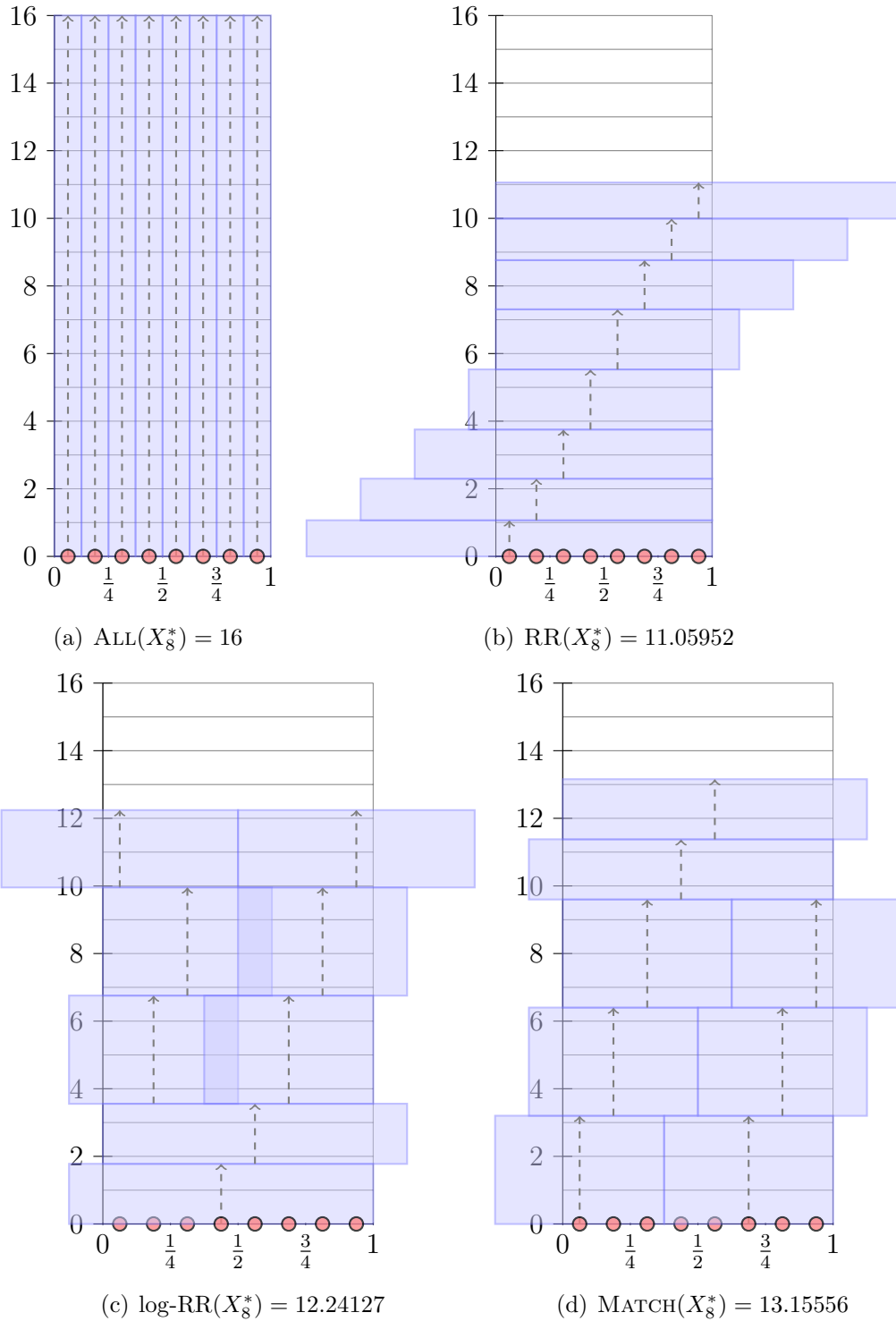


Figure 2.6: Graphical comparison of algorithms on X_g^* . Note that while ALL is optimal against perfect deployment, ROUNDROBIN performs worst.

Chapter 3

Set Radius Strip Cover

In this section we present an optimal $O(n^2 \log n)$ -time algorithm for the SET RADIUS STRIP COVER (RADSC) problem.

3.1 Polynomial Time Algorithm

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 3.1. *There exists a proper optimal assignment for every RADSC instance $I = (X, B)$.*

Proof. 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 with lifetime T , and is thus optimal. \square

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 3.2. *Let T be the optimal lifetime of a given RADSC instance $I = (X, B)$. There exist two sensors $i, k \in \{0, \dots, n+1\}$, 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 3.1. 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 . \square

Lemma 3.2 implies that there are $O(n^2)$ possible lifetimes. This leads to an algorithm for solving RADSC.

Theorem 3.3. *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 $i, k \in \{0, \dots, n + 1\}$ 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 $\rho_\ell^{ik} \leftarrow b_\ell / T_{ik}$, 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 $O(n^2 \log n)$. \square

Chapter 4

Set Once Strip Cover

In this chapter we consider the ONCESC problem, a restriction of the SENSOR STRIP COVER problem in which the radius and activation time of each sensor can be set only once. The major results of this chapter are to prove that ONCESC is NP-hard, and to prove that $\frac{3}{2}$ is an upper bound for the approximation ratio of ROUNDROBIN in ONCESC. In Chapter 5, we will extend this result to SENSOR STRIP COVER.

4.1 Relationship to Other Problems

While no complexity result for SENSOR STRIP COVER is known, we show in this section that ONCESC is NP-hard, using a reduction from PARTITION. Furthermore, we show that the set-once restriction induces a performance penalty, even for an optimal algorithm.

4.1.1 Computational Complexity

The PARTITION problem is well-known to be NP-complete [19].

Definition 4.1 (PARTITION). *Given a multiset Y of integers, do there exist two non-empty disjoint subsets $Y_0, Y_1 \subseteq Y$, such that $Y_0 \cup Y_1 = Y$, and $\sum_{y \in Y_0} y = \sum_{y \in Y_1} y$?*

We prove that the decision version of ONCESC is NP-complete via a reduction from PARTITION. That is, given an instance (X, B) , does there exist a ONCESC schedule that achieves the maximum possible lifetime of $2B^*$?

Theorem 4.2. *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 by placing n sensors with battery y_i at $\frac{1}{2}$, and two additional sensors equipped with battery β at $\frac{1}{6}$ and $\frac{5}{6}$, respectively. 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 $Y \in \text{PARTITION}$ if and only if the maximum possible lifetime of $8\beta = 2B^*$ is achievable for the ONCESC instance.

First, suppose $Y \in \text{PARTITION}$, hence there exist two non-empty disjoint subsets $Y_0, Y_1 \subseteq Y$, such that $Y_0 \cup Y_1 = Y$, and $\sum_{y \in Y_0} y = \beta = \sum_{y \in Y_1} y$.

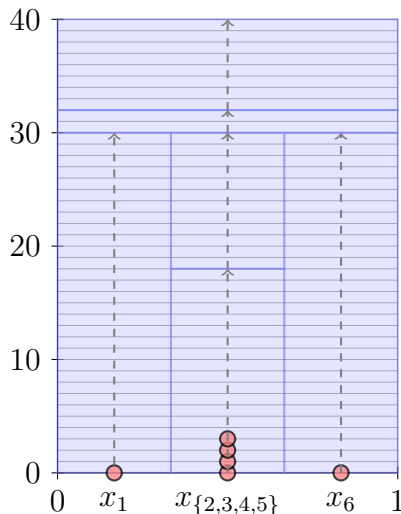


Figure 4.1: Example illustrating that ONCESC is NP-hard. $Y = \{1, 2, 3, 4\}$ is a given instance of PARTITION, and $(X, B) = ((\frac{1}{6}, \frac{1}{2}, \dots, \frac{1}{2}, \frac{5}{6}), (5, 1, 2, 3, 4, 5))$ is the translated ONCESC instance. A solution to PARTITION corresponds to a ONCESC schedule that achieves the maximum possible lifetime of $2B^* = 40$.

Schedule the sensors in Y_0 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 $i \in Y_0$. 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 in Y_0 can thus cover $[0, 1]$ for 6β time units (see Figure 4.1 for an example). Next, the sensors in Y_1 can cover $[0, 1]$ for an additional 2β time units, since they all require a radius of $\rho_{i+1} = \frac{1}{2}$, for every $i \in Y_1$. 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 batteries of these two partitions form a solution to PARTITION.

This completes the proof that the decision version of ONCESC is NP-complete. It follows that the optimization version of ONCESC is NP-hard.

□

4.1.2 Performance gap with Sensor Strip Cover

Next, we show that there is a gap between the lifetime of the optimal solutions of ONCESC and SENSOR STRIP COVER. In a sense, we lose flexibility by not allowing a sensor to receive multiple coverage assignments.

Lemma 4.3. *There exists an instance (X, B) for which the ratio between the optimal value for SENSOR STRIP COVER and the optimal value of ONCESC is $\frac{7}{6}$.*

Proof. Consider the instance with three sensors where $X = (\frac{1}{6}, \frac{1}{2}, \frac{5}{6})$ and $B = (1, 1.5, 1)$.

In SENSOR STRIP COVER all can work together with radii $\frac{1}{6}$ for 6 time units, and then the second sensor can survive for 1 additional time unit by adjusting its radius to $\frac{1}{2}$. Hence we get a lifetime of $7 = 2B^*$, which by Lemma 2.1 is the best possible.

On the other hand, in ONCESC the middle sensor cannot readjust its radius. Thus, if the sensors all work together with radii of $\frac{1}{6}$, then the lifetime achieved is 6 and we are done. On the other had, the middle sensor can work alone with a radius of $\frac{1}{2}$ for 3 time units. Then the other two sensors can work together with radii of $\frac{1}{3}$ for an additional 3 units. Again, the combined lifetime is 6 time units. We depict these schedules graphically in Figure 4.2. □

4.1.3 Duty cycle algorithms

In this section we show that, in the worst case, no duty cycle *algorithm* performs better than ROUNDROBIN in ONCESC. More specifically, we show

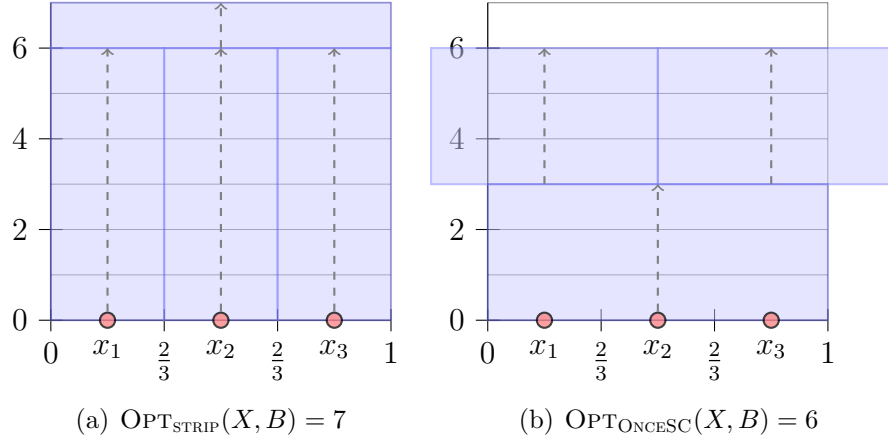


Figure 4.2: Performance gap between ONCESC and SENSOR STRIP COVER. For the instance $X = (\frac{1}{6}, \frac{1}{2}, \frac{5}{6})$ and $B = (1, 1.5, 1)$, an optimal ONCESC solution is no better than a $\frac{7}{6}$ approximation of an optimal SENSOR STRIP COVER solution.

that the approximation ratio of any duty cycle algorithm is at least $\frac{3}{2}$.

Lemma 4.4. *The approximation ratio of any duty cycle algorithm is at least $\frac{3}{2}$.*

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 $8 = 2 \sum_i b_i = 2B^*$.

On the other hand, the best duty cycle algorithm is ROUNDROBIN, which

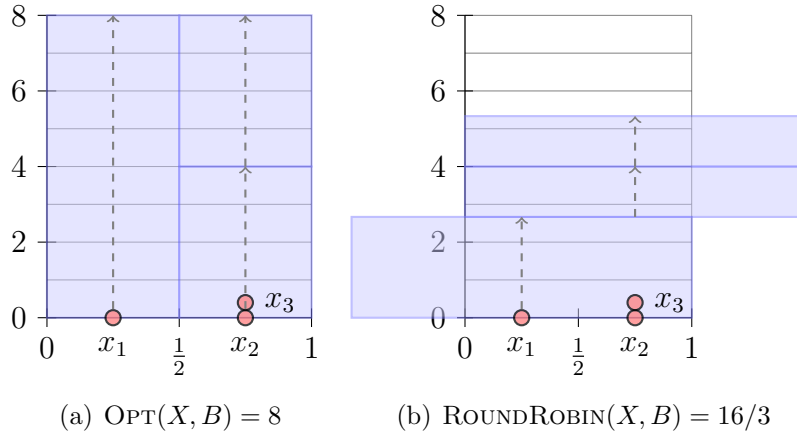


Figure 4.3: Best ONCESC schedule vs. best duty cycle schedule. Here $X = (\frac{1}{4}, \frac{3}{4}, \frac{3}{4})$ and $B = (2, 1, 1)$.

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 4.3. □

4.2 Worst-case analysis of RoundRobin

Having shown in Section 4.1.1 that ONCESC is NP-hard, we now turn our attention to approximation algorithms. In particular, we find the precise value of the approximation ratio of ROUNDROBIN. While ROUNDROBIN is probably the simplest possible algorithm, Obs. 2.4 and Obs. 5.1 imply only that its approximation ratio is between $\frac{3}{2}$ and 2. In this section, we show that the approximation ratio of ROUNDROBIN in ONCESC is exactly $\frac{3}{2}$.

The structure of the proof is as follows. We start with an optimal schedule

S , and cut it into disjoint time intervals, or strips, such that the same set of sensors is active within each time interval. Each strip induces a RADSC instance I_j and solution S_j . For each such strip $I_j = (X^j, B^j)$ we compare the performance of ROUNDROBIN to the lifetime T_j of the strip solution S_j . More specifically, we prove that $\text{RR}(X^j, B^j) \geq \frac{3}{2}T_j$. We do this using a reduction to a special case of RADSC with unit batteries.

4.2.1 Cutting the Schedule into Strips

Given an instance $I = (X, B)$, and a solution $S = (\rho, \tau)$ with lifetime T , 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 $\Omega = \{\omega_1, \dots, \omega_\ell\}$, where $\omega_j < \omega_{j+1}$, for every j . Notice that $0, T \in \Omega$. Next, we partition the time interval $[0, T]$ into the sub-intervals $[\omega_j, \omega_{j+1}]$, for every $j \in \{1, \dots, \ell - 1\}$.

We define a new instance for every sub-interval. Let $X^j \subseteq X$, for every $j \in \{1, \dots, \ell - 1\}$ be the set of sensors that participate in covering $[0, 1]$ during the j th sub-interval. That is,

$$X^j = \{x_i : [\omega_j, \omega_{j+1}] \subseteq [\tau_i, \tau_i + b_i/\rho_i]\} .$$

Also, let $T_j = \omega_{j+1} - \omega_j$, and let B_i^j be the energy that was consumed by sensor

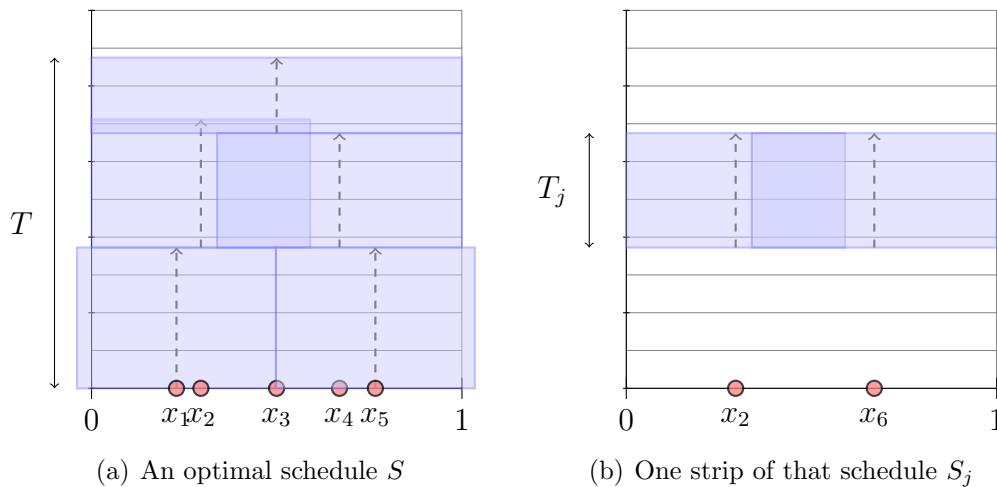


Figure 4.4: Cutting an optimal schedule into strips. Note that coverage overlaps may occur in both the horizontal and vertical directions in the optimal schedule, but only horizontally in a strip.

i during the j th sub-interval, i.e., $B_i^j = \rho_i \cdot T_j$. Observe that $I_j = (X^j, B^j)$ is a valid instance of RADSC, for which ρ^j , where $\rho_i^j = \rho_i$ for every sensor i such that $x_i \in X^j$, is a solution that achieves a lifetime of exactly T_j . Note that even if S is an optimal solution to I , it is not necessarily the case that ρ_j is an optimal solution to I_j .

We further modify the instance $I_j = (X^j, B^j)$ and the solution ρ^j as follows:

- Starting with $i = 1$, we remove sensor i from the instance if the interval $[0, 1]$ is covered during $[\omega_j, \omega_{j+1}]$ without i .
- We decrease the battery and the radius of the left-most sensor as much

as possible, and we also decrease the battery and the radius of the right-most sensor as much as possible.

Figure 4.4 illustrates the procedure of cutting an optimal schedule into strips.

Observation 4.5. *Let sensors 1 and m be the left-most and right most sensors in X^j . Then, either $\rho_1^j = x_1$ or the point $x_1^j + \rho_1^j$ is only covered by sensor 1. Also, either $\rho_m^j = 1 - x_m$ or the point $x_m^j - \rho_m^j$ is only covered by sensor m .*

Observe that $\text{RR}(X^j, B^j) = \sum_{x_i \in x^j} \frac{b_i^j}{r_i}$ is the ROUNDROBIN lifetime of the j^{th} strip. In the sequel we show that

Lemma 4.6. $\text{RR}'(X^j, B^j) \geq \frac{2}{3}T_j$, for every j .

It follows that

Theorem 4.7. ROUNDROBIN is a $\frac{3}{2}$ -approximation algorithm for ONCESC.

Proof. First, observe that

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

By Lemmas 2.5 and 4.6 we have that

$$\begin{aligned}
\text{RR}(X, B) &\geq \sum_j \text{RR}(X^j, B^j) \geq \sum_j \text{RR}'(X^j, B^j) \\
&\geq \sum_j \frac{2}{3} T_j = \frac{2}{3} T \\
&= \frac{2}{3} \text{OPT}(X, B) .
\end{aligned}$$

□

4.2.2 Reduction to RadSC with Uniform Batteries

Given the RADSC instance $I_j = (X^j, B^j)$ and a solution ρ^j , we construct an instance $I'_j = (Y^j, \bar{1})$ with unit size batteries and a RADSC solution σ^j , such that the lifetime of σ_j is T_j .

Let OPT_0 denote the optimal RADSC lifetime. We assume that $b_i^j \in \mathbb{N}$ and $b_i^j \geq 3$ for every i , since (i) $b_i^j \in \mathbb{Q}$ for every i , (ii) $\text{OPT}_0(X, \beta B) = \beta \cdot \text{OPT}_0(X, B)$, and (iii) $\text{RR}(X, \beta B) = \beta \cdot \text{RR}(X, B)$.

The instance I'_j is constructed as follows. We replace each sensor i such that $x_i^j \in X^j$ with b_i^j unit battery sensors whose average location is x_i . These unit battery sensors are called the *children* of i . To do this, we divide the interval $[x_i^j - \rho_i^j, x_i^j + \rho_i^j]$ into b_i^j equal sub-intervals, and place a unit battery sensor in the middle of each sub-interval. Observe that child sensors may be placed outside $[0, 1]$, namely to the left of 0 or to the right of 1. The

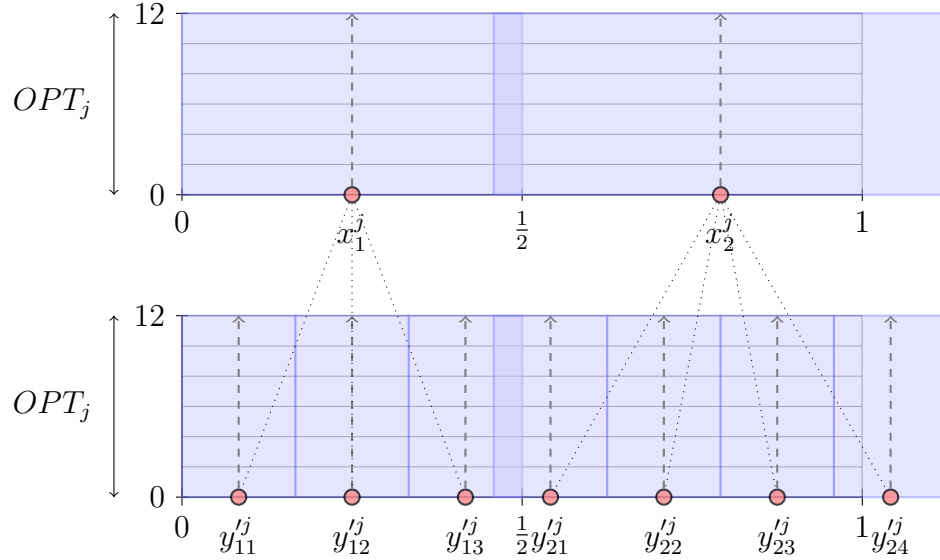


Figure 4.5: Reduction of non-uniform battery instance I_j to uniform battery instance I'_j . At the top, $I_j = ((\frac{1}{4}, \frac{19}{24}), (3, 4))$, while at the bottom, $I'_j = ((\frac{1}{12}, \frac{3}{12}, \frac{5}{12}, \frac{13}{24}, \frac{17}{24}, \frac{21}{24}, \frac{25}{24}), \bar{1})$.

solution σ_j is defined as follows. For any child k of a sensor i in I_j , we set $\sigma_k^j = \rho_i^j / b_i^j$. The example shown in Figure 4.5 illustrates the reduction from a non-uniform battery instance in a particular strip, to a unit battery instance.

We prove that the lifetime of σ^j is T_j .

Lemma 4.8. *The lifetime of σ_j is T_j .*

Proof. First, the b_i^j children of a sensor i in I_j cover the interval $[x_i^j - \rho_i^j, x_i^j + \rho_i^j]$. Also, a child k of i survives $1/\sigma_k^j = b_i^j/\rho_i^j = T_j$ time units. \square

Next, we prove that the lower bound on the performance of ROUNDROBIN did not change.

Lemma 4.9. $\text{RR}'(Y^j, \bar{1}) = \text{RR}'(X^j, B^j)$.

Proof. Let p^j be the ROUNDROBIN radii of Y^j . Hence,

$$\text{RR}'(Y^j, \bar{1}) = \frac{\sum_i b_i^j}{p^j} = \frac{B^{j*}}{\frac{1}{B^{j*}} \sum_k p_k^j} = \frac{B^{j*}}{\frac{1}{B^{j*}} \sum_i b_i^j r_i^j} = \frac{B^{j*}}{r^j} = \text{RR}'(X^j, B^j),$$

where we have used the fact that a parent's location is equal to the average location of her children. \square

Lemma 4.6 now follows from Lemmas 4.8, 4.9, and 4.15. The latter states that $\text{RR}'(Y^j, \bar{1}) \geq \frac{2}{3} \text{OPT}_0(Y^j, \bar{1})$, and we devote the next section to proving it.

4.2.3 Analysis of Round Robin for Unit Batteries

For the remainder of this section, we assume that we are given a unit battery instance X that corresponds to the j th 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 $[0, 1]$. We show that $\text{RR}'(X) \geq \frac{2}{3} \text{OPT}_0(X)$.

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

Lemma 4.10. $\max_{i \in \{i_0, \dots, i_1-1\}} \{x_{i+1} - x_i\} = \max_{i \in \{1, \dots, n-1\}} \{x_{i+1} - x_i\}$.

Proof. By Observation 4.5 either $\rho_1^j = x_1$ and hence none of its children are located to the left of 0, or the point $x_1^j + \rho_1^j$ is 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 $b_i^j \geq 3$, for all i .) The same argument can be used for the right-most sensor. \square

We define

$$d_0 \triangleq \begin{cases} x_{i_0} - x_{i_0-1} & \text{if } i_0 > 1 \text{ and } -x_{i_0-1} < x_{i_0}, \\ 2x_{i_0} & \text{otherwise,} \end{cases}$$

and

$$d_1 \triangleq \begin{cases} x_{i_1+1} - x_{i_1} & \text{if } i_0 < n \text{ and } x_{i_0+1} - 1 < 1 - x_{i_1}, \\ 2(1 - x_{i_1}) & \text{otherwise,} \end{cases}$$

and

$$\Delta \triangleq \max \left\{ d_0, d_1, \max_{i \in \{i_0, \dots, i_1-1\}} \{x_{i+1} - x_i\} \right\} .$$

We describe the optimal RADSC lifetime in terms of Δ .

Lemma 4.11. *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 d_0, d_n , 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 $\Delta = d_0$, then if $d_0 = 2x_{i_0}$, 0 is uncovered, and otherwise, there is a point in $[0, x_{i_0}]$ that is uncovered. A similar argument holds if $\Delta = d_n$. \square

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.

Definition 4.12. *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:*

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

Figure 4.6 is an illustration of how unit battery instances are stretched.

Observation 4.13. *Let X' be a stretched instance of X . Then $|\{i : x'_i \leq \frac{1}{2}\}| = \lceil n/2 \rceil$ and $|\{i : x'_i > \frac{1}{2}\}| = \lfloor n/2 \rfloor$.*

Lemma 4.14. *Let X' be the stretched instance of X . Then, $\text{OPT}_0(X') = \text{OPT}_0(X)$ and $\text{RR}'(X') \leq \text{RR}'(X)$.*

Proof. First, by construction, the internal gaps in X' are of length Δ and $d'_0, d'_1 \leq \Delta$. Thus, by Lemma 4.11, $\text{OPT}_0(X') = \text{OPT}_0(X)$. By Lemma 4.10

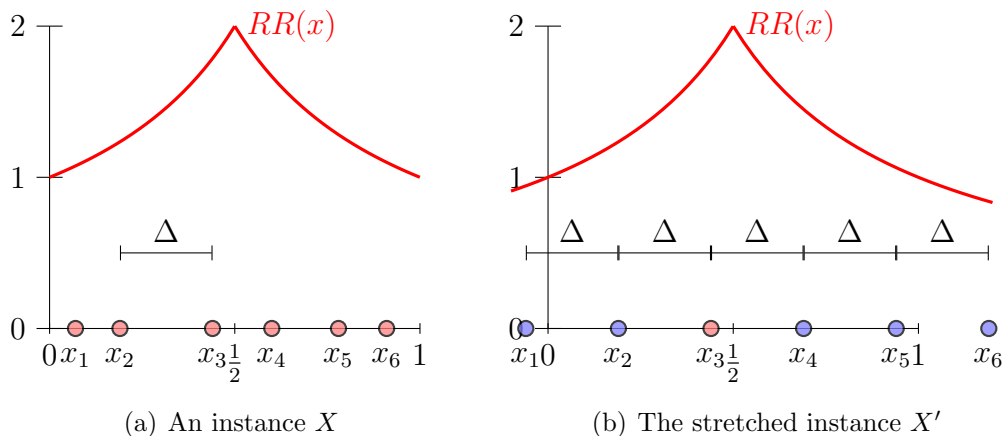


Figure 4.6: Transformation of instance X to stretched instance X' . The sensor closest to $\frac{1}{2}$ (x_3) remains in place, while the other sensors are placed at increasing intervals of Δ away from x_3 . The curve traces the ROUNDROBIN lifetime of a sensor as a function of its location.

we know that the sensors moved away from $\frac{1}{2}$, hence $\sum_i r'_i \geq \sum_i r_i$ and $RR'(X') \leq RR'(X)$. \square

Now we are ready to bound $RR(X)$.

Lemma 4.15. $RR'(X) \geq \frac{2}{3} \text{OPT}_0(X)$, for every instance $I = (X, \bar{1})$ of RADSC, where sensors may be located outside $[0, 1]$.

Proof. First by Lemma 4.14 we may assume that the instance is stretched.

Suppose that n is even. Then 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,

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

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

$$\frac{\text{RR}'(X)}{\text{OPT}_0(X)} = \frac{n/\bar{r}}{2/\Delta} \geq \frac{n\Delta}{1 + \frac{n\Delta}{2}} = \frac{1}{\frac{1}{n\Delta} + \frac{1}{2}} \geq \frac{2}{3}.$$

If n is odd, then without loss of generality we can assume that there are

exactly $\frac{n+1}{2}$ sensors to the left of $1/2$, and exactly $\frac{n-1}{2}$ to the right. Then

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

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

$$\frac{\text{RR}'(X)}{\text{OPT}_0(X)} = \frac{n\bar{r}}{2/\Delta} = \frac{n\Delta}{1 + \frac{(n^2+1)\Delta}{2n}} = \frac{1}{\frac{1}{n\Delta} + \frac{1}{2} + \frac{1}{2n^2}} \geq \frac{2n^2}{3n^2 - n + 1} > \frac{2}{3} .$$

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

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

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 stretched 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

$$\begin{aligned} r_1 - r_2 &= \left(r_{\frac{n+1}{2}} + \frac{\Delta(n-1)}{2} \right) - \left(r_{\frac{n+3}{2}} + \frac{\Delta(n-3)}{2} \right) \\ &= r_{\frac{n+1}{2}} - r_{\frac{n+3}{2}} + \Delta \\ &= 2\epsilon . \end{aligned}$$

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

$$\begin{aligned} \bar{r} &\geq \frac{r_{\frac{n+1}{2}} + r_{\frac{n+3}{2}}}{2} + \frac{r_{\frac{n+1}{2}} - r_{\frac{n+3}{2}}}{2n} + \frac{\Delta}{n} \frac{(n-1)^2}{4} \\ &\geq \frac{1 + \Delta}{2} - \frac{\Delta}{4n} + \frac{\Delta}{n} \frac{(n-1)^2}{4} \\ &= \frac{1}{2} + \frac{n\Delta}{4} . \end{aligned}$$

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

Chapter 5

Sensor Strip Cover

In the previous chapters, we found a polynomial-time solution for RADSC, and showed that ONCESC is NP-hard. In this chapter, we consider the more general SENSOR STRIP COVER problem. In the first section, we analyze the worst-case performance of several algorithms for the case where the battery charges of the sensors were allowed to be different. In the second section, we analyze the average-case performance of these same algorithms under the assumption that all of the initial battery charges are the same. Without loss of generality, we assume that all batteries are unit (e.g. $B = (1, \dots, 1)$).

5.1 Worst-case analysis

By Obs. 2.4, 2 is an upper bound on the approximation ratio of any algorithm. In what follows, we find lower bounds for the approximation ratios of several natural algorithms, and a tight upper bound for the approximation

ratio of ROUNDROBIN.

5.1.1 RoundRobin

The simplest algorithm, ROUNDROBIN, forces each sensor to successively cover all of U for as long as possible. That is, each sensor i is assigned a radius of $r_i \triangleq \max\{x_i, 1 - x_i\}$, and is pushed onto a single queue¹.

Observation 5.1. *The approximation ratio of ROUNDROBIN in SENSOR STRIP COVER is at least $\frac{3}{2}$.*

Proof. Consider the instance with $X = (\frac{1}{4}, \frac{3}{4})$ and $B = (1, 1)$. If the sensors work together to cover U , then each employs a radius of $\frac{1}{4}$, and a lifetime of 4 (which is the best possible) is achieved. On the other hand, in ROUNDROBIN the sensors work alone, and each employs a radius of $\frac{3}{4}$. Thus, ROUNDROBIN achieves a lifetime of only $\frac{8}{3}$, and the ratio between $\text{OPT}(X) = 4$ and $\text{ROUNDROBIN}(X) = \frac{8}{3}$ is $\frac{3}{2}$. We depict both schedules graphically in Figure 5.1. □

Note that this result applies to ONCESC and k -DUTYSC, for $k > 1$, as well. In particular, it is interesting that the performance of ROUNDROBIN suffers this much against even the weakest duty cycling model, 2-DUTYSC.

¹A queue is an abstract data structure in which the first element pushed into the queue is the first element to come out (FIFO). Note that this is consistent with the definition of ROUNDROBIN shown in Equation 2.1.

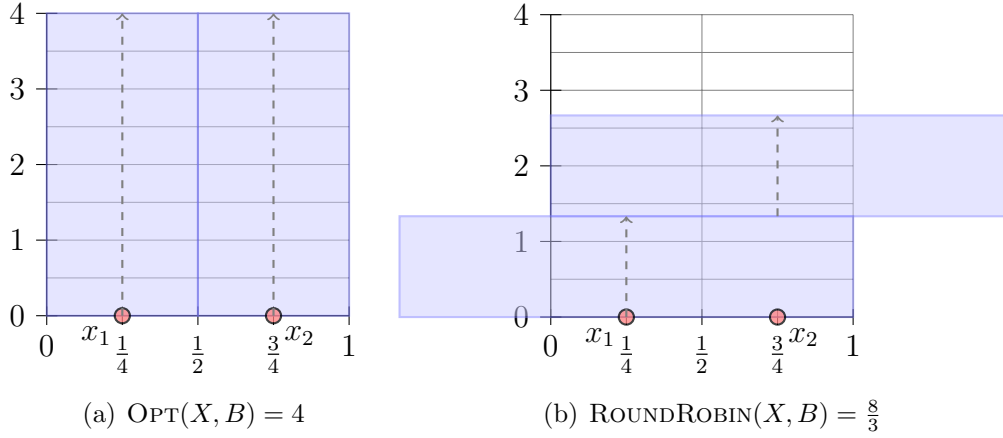


Figure 5.1: Depiction of $\frac{3}{2}$ lower bound for ROUNDROBIN. For the instance $X = (\frac{1}{4}, \frac{3}{4})$ and $B = (1, 1)$, OPT produces a lifetime of 4, but ROUNDROBIN achieves a lifetime of only $\frac{8}{3}$.

The $\frac{3}{2}$ lower bound for ROUNDROBIN in SENSOR STRIP COVER proves that ROUNDROBIN cannot guarantee that it will produce a lifetime greater than $\frac{2}{3}$ of OPT. But from Lemma 2.1, the best guarantee that ROUNDROBIN can make is a lifetime of at least $\frac{1}{2}$ of OPT. In fact, we will show that ROUNDROBIN can guarantee at least $\frac{2}{3}$ of OPT in all instances.

Theorem 5.2. ROUNDROBIN is a $\frac{3}{2}$ -approximation algorithm for SENSOR STRIP COVER.

Proof. Let S be an optimal schedule for instance (X, B) of SENSOR STRIP COVER. In a manner similar to the one we employed in Section 4.2.1, we can time-divide this schedule into disjoint time intervals in which no sensor changes its coverage assignment. The set of active sensors and the associ-

ated battery life that they consume in each of these time intervals constitute a valid ONCESC instance. By Theorem 4.7, ROUNDROBIN achieves a lifetime of at least $2/3$ of OPT on any ONCESC instance. Thus, the lifetime of ROUNDROBIN on *every* time interval is at least $2/3$ of OPT, and the cumulative performance of ROUNDROBIN(X, B) is at least $2/3$ of OPT(X, B). \square

5.1.2 Greedy

Consider next a GREEDY algorithm, which iteratively schedules the least-wasteful assignment of radii until a sensor runs out of battery life. In this case, the least wasteful assignment is the one that minimizes overlapping or unnecessary coverage (e.g. outside of U).

Observation 5.3. *The approximation ratio of GREEDY is at least $\frac{6}{5}$.*

Proof. Consider the instance with $X(\epsilon) = \{\frac{1}{6} - \epsilon, \frac{1}{2}, \frac{5}{6}\}$, for some $\epsilon > 0$ and $B = (1, 1, 1)$. By working alone (i.e. in ROUNDROBIN fashion) the middle sensor can cover U perfectly with a radius of $\frac{1}{2}$. In fact, no other assignment can cover U without producing some wasteful coverage. Thus, GREEDY chooses to activate the middle sensor by itself on U first, and then use the remaining sensors in tandem afterwards. This produces a lifetime approaching 5 as $\epsilon \rightarrow 0$. On the other hand, by sacrificing some wasted coverage, the optimal algorithm can run all three sensors together, and achieve

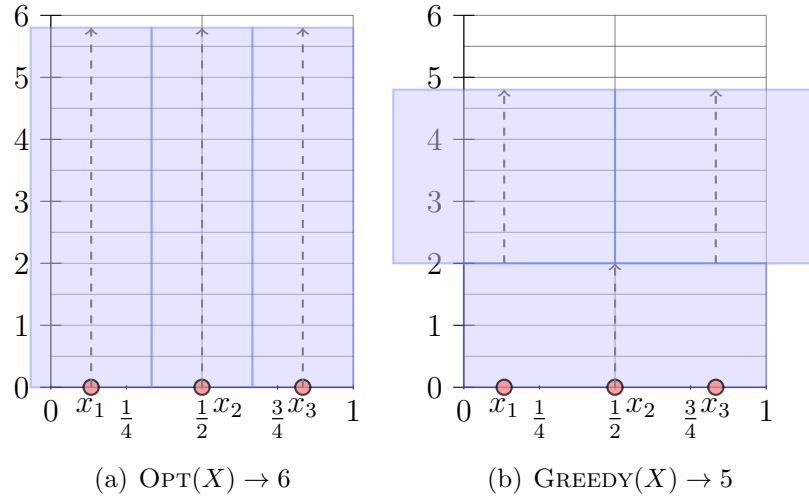


Figure 5.2: Depiction of $\frac{6}{5}$ lower bound for GREEDY. For the instance $X = \{\frac{1}{6} - \epsilon, \frac{1}{2}, \frac{5}{6}\}$, OPT approaches a lifetime of 6 as $\epsilon \rightarrow 0$, while GREEDY approaches a lifetime of only 5.

$\lim_{\epsilon \rightarrow 0} \text{OPT}(X, B) = 6$. We depict both schedules in Figure 5.2. □

Remark 5.4. *In Section 5.2.3, we define a more sophisticated algorithm called log-ROUNDROBIN, and analyze its average-case performance. Here, it suffices to remark that the perfect deployment instance X_{2k}^* provides a $\frac{3}{2}$ lower bound on the approximation ratio of log-ROUNDROBIN with depth parameter k . Subsequent optimizations that we make to this algorithm do not affect the lower bound.*

5.2 Average-case analysis

For any solution, let $\bar{T} = T/n \in [1, 2]$ 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 the average lifetime of a particular coverage group.

5.2.1 RoundRobin

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

$$\mu_T \triangleq \mathbb{E}[T_{0,1}(X)] = \int_0^1 \frac{dx}{\max(x, 1-x)} = 2 \int_{\frac{1}{2}}^1 \frac{dx}{x} = 2 \ln x \Big|_{\frac{1}{2}}^1 = 2 \ln 2, \quad (5.1)$$

and variance

$$\sigma_T^2 \triangleq \mathbb{E}[T_{0,1}^2(X)] - \mu_T^2 = \int_0^1 \frac{dx}{(\max(x, 1-x))^2} - \mu_T^2 = 2 - 4 \ln^2 2. \quad (5.2)$$

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 $n \rightarrow \infty$. For this reason we report the variance but focus most of our attention on the expected average lifetime of each algorithm.

Theorem 5.5. *If the sensor locations are distributed uniformly at random over $[0, 1]$, then ROUNDROBIN achieves an approximation ratio of $1/\ln 2 \approx 1.443$ in the average case.*

Proof. Since $\text{OPT}(X) \leq 2n$ by Lemma 2.1, the result follows from equation 5.1. □

5.2.2 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 $i = 1, \dots, k$ 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 5.6. *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 [26], 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

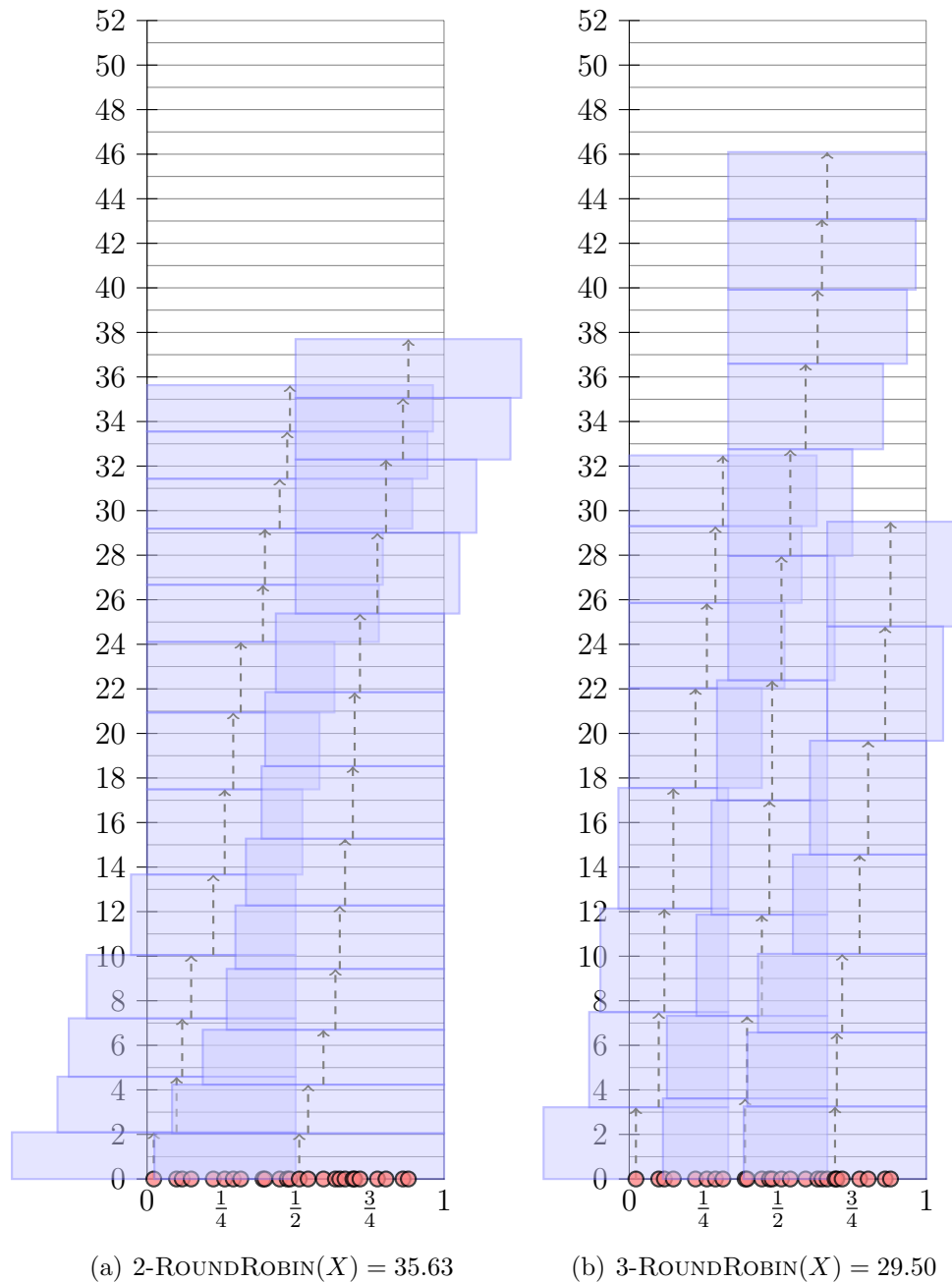


Figure 5.3: $k\text{-ROUNDROBIN}$ for $k = 2, 3$ against a random instance with 26 sensors. Even with proper load balancing, there are many inefficient sensors. The resulting average lifetimes are 1.370 (left) and 1.135 (right).

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 $n \rightarrow \infty$ 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 -ROUNDRROBIN on perfectly balanced stacks, and then throws the n_2 leftover sensors away. Thus, the actual expected average lifetime of the algorithm is

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

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

Observation 5.7. *k -ROUNDRROBIN provides the same worst-case and average-case performance as ROUNDRROBIN.*

5.2.3 log-RoundRobin

Nevertheless, clever applications of ROUNDRROBIN can yield efficient algorithms. While the expected lifetime of a sensor in ROUNDRROBIN is in-

dependent 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

$$\mathbb{E}[T_{a,b}(X; c)] = \frac{1}{2c} \int_{\frac{b+a}{2}-c}^{\frac{b+a}{2}+c} \frac{dx}{\max(x-a, b-x)} = \frac{1}{c} \ln \left(1 + \frac{2c}{b-a} \right). \quad (5.3)$$

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:

$$Var(\hat{T}_{a,b}(X; c)) = 4 \left[1 - \frac{1}{1 + \frac{b-a}{2c}} - \left(\frac{b-a}{2c} \cdot \ln \left(1 + \frac{2c}{b-a} \right) \right)^2 \right]. \quad (5.4)$$

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-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 $i = 0, 1, \dots, 2^k$.² If sensor

²Note 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} .

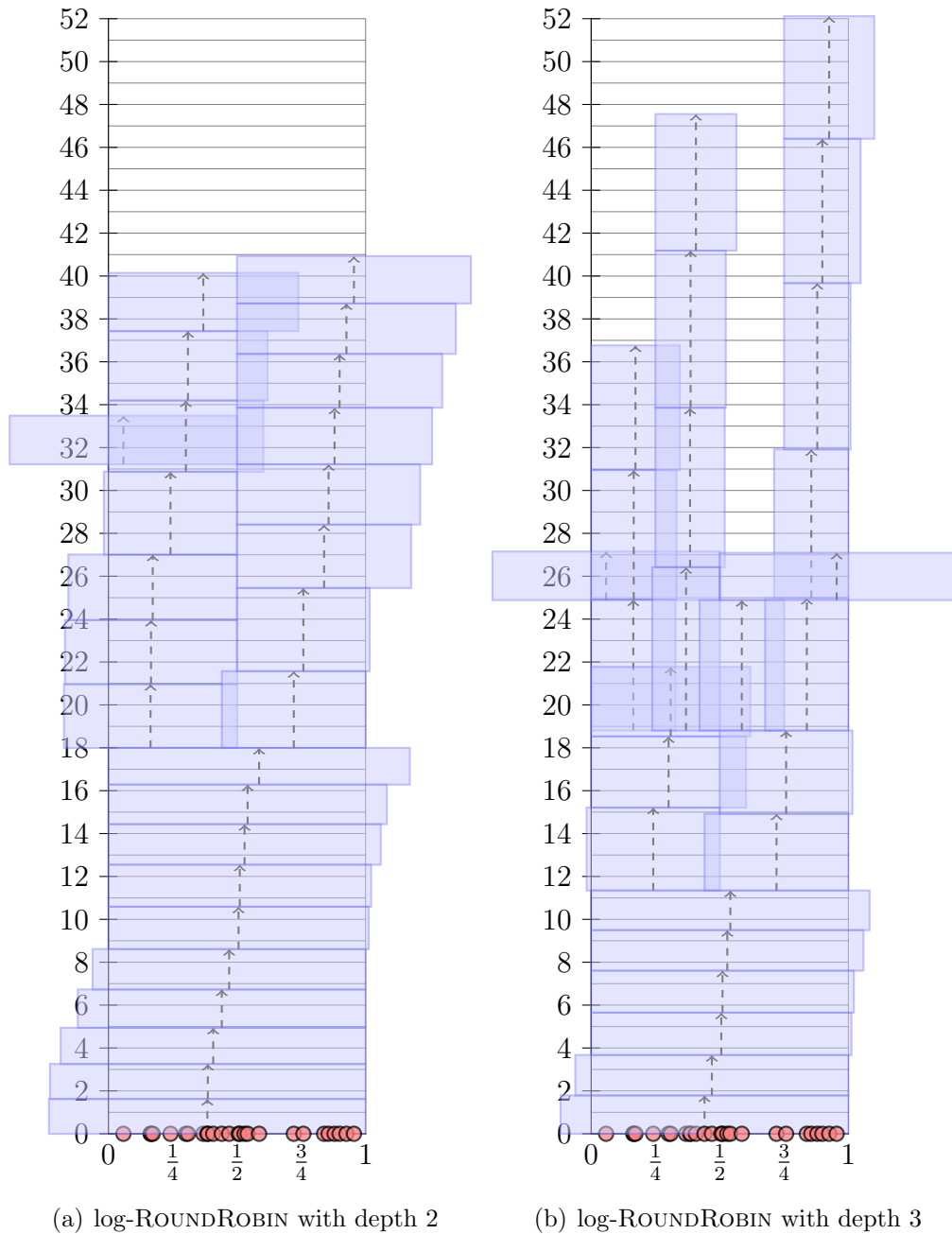


Figure 5.4: log-ROUNDROBIN with depth 2 and 3 against a random instance with 26 sensors. The depth parameter of 2 depicted at left offers better load balancing than the parameter of 3 at right for 26 sensors.

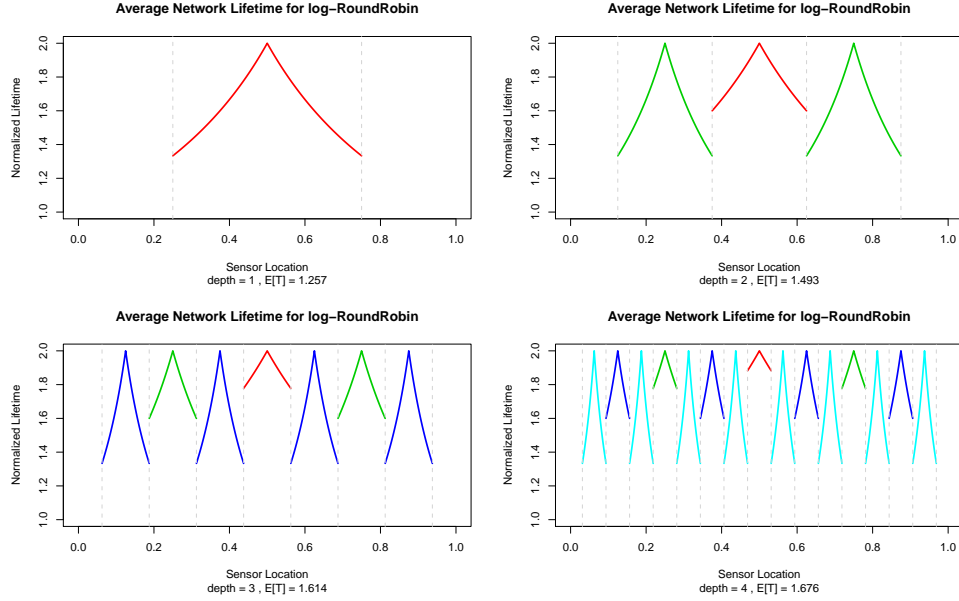


Figure 5.5: Normalized sensor network lifetime for $k = 1, 2, 3, 4$ using the log-ROUNDROBIN algorithm. Each color represents the lifetime of the sensors in $\Gamma_k(j)$. Note that while the actual lifetime of a sensor in $\Gamma_k(j)$ may reach 2^j , it must run in parallel with 2^{j-1} partners, so the normalized lifetime of the group is at most 2. The expected average lifetime of the network approaches 1.737752 as $k \rightarrow \infty$.

$x \in U_k(i)$, 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 is shown in Figure 5.5.

For $j = 1, \dots, k$, we define $\Gamma_k(j)$ to 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 for $\Gamma_k(j)$ using Equation 5.3

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

and the variance using Equation 5.4:

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

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

$$\mathbb{E}[\hat{T}_k] = \sum_{j=1}^k \frac{\mathbb{E}[\hat{T}_k(j)]}{2^{k-j+1}} = 2 \ln \prod_{j=1}^k (1 + 2^{j-k-1}) = 2 \ln \prod_{\ell=1}^k (1 + 2^{-\ell}). \quad (5.5)$$

The analogous infinite product is a q -series [36], denoted here by $(-1; \frac{1}{2})_{\infty}$, for which we can compute an approximate limiting value. This leads directly to the expected average lifetime:

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

³We 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 $k \rightarrow \infty$, so we omit it from our calculations.

The mean normalized variance satisfies

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

which has the approximate limit of 0.02202547 as $k \rightarrow \infty$. Computation of the total variance is omitted, since it requires extensive calculation that adds little elucidation, but it will converge to the above as $k \rightarrow \infty$.

Furthermore, it is clear from Figure 5.5 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-ROUNDROBIN, each set $\Gamma_k(j)$ for $j = 1, \dots, k$ maintains 2^{j-1} parallel queues. Proper functioning of our algorithm requires balanced loads across these queues, but the hierarchical structure of log-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, suppose 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 $n_1 = 2^k \cdot \min_{1 \leq i \leq 2^k-1} N_i$ yields

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

Theorem 5.8. *The log-ROUNDROBIN algorithm is at least a $\frac{3}{2}$ -approximation of OPT, but for sufficiently large n , achieves an approximation ratio of 1.151 in the average case with high probability.*

5.2.4 Optimizations of log-RoundRobin

Still, it is clear from Figure 5.5 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 $\frac{1}{2 \ln \frac{3}{2}} \approx 1.233$. On the other hand, it is easy to see that the relative efficiency of $\Gamma_k(1)$ approaches 1 as $k \rightarrow \infty$. Therefore, we can improve the efficiency of log-ROUNDROBIN by shrinking the intervals over which $\Gamma_k(k)$ is active. Note that since every $\Gamma_k(j)$ for $j = 1, \dots, k-1$ 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 5.3, the expected normalized

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

lifetime becomes

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

for $j = 1, \dots, k-1$, 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 5.5 that can be expressed as another q -series:

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

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

$$T'_k(\epsilon) = 0 \Rightarrow \frac{1}{3-\epsilon} = \sum_{j=1}^{k-1} \frac{1}{2^{j+1} + 1 + \epsilon}. \quad (5.6)$$

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 5.1, alongside the expected network lifetime of the optimized algorithm. These optimizations improve the expected average network lifetime by more than 3% above that of log-ROUNDROBIN.

k	ϵ	$T_k(0)$	$T_k(\epsilon)$	Gain %	$ U_k(k; \epsilon) \%$
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 5.1: Numerical approximations for optimized log-ROUNDROBIN. 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 5.9. *For sufficiently large n , the optimized log-ROUNDROBIN algorithm achieves an approximation ratio of 1.117 in the average case 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 $k \rightarrow \infty$ 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.

Chapter 6

k -Duty Cycle Strip Cover

In this chapter we explore the k -DUTY CYCLE STRIP COVER (k -DUTYSC) problem, in which the sensors must be grouped into disjoint *shifts* of size at most k , which take turns covering the interval. This restriction of SENSOR STRIP COVER is of independent interest, and as we saw in the Introduction, is a frequently used device for attacking sensor cover problems. Note that once again we assume uniform batteries throughout this chapter.

6.1 2-DutySC

While 1-DUTYSC can be solved trivially by ROUNDROBIN, we present a polynomial-time algorithm for solving 2-DUTYSC. The algorithm is based on a reduction to the MAXIMUM WEIGHT MATCHING problem in bipartite graphs, which can be solved in $O(n^2 \log n + nm)$ time in graphs with n vertices and m edges (see, e.g., [17]).

Theorem 6.1. *2-DUTYSC can be solved in polynomial time.*

Proof. Given a 2-DUTYSC instance X with n sensors, we construct a bipartite graph $G = (L, R, E)$ as follows:

$$L = \{v_i : i \in \{1, \dots, n\}\}$$

$$R = \{v'_i : i \in \{1, \dots, n\}\}$$

$$E = \{(v_i, v'_j) : i \leq j\} .$$

The weight of an edge $e = (v_i, v'_j)$ is defined as follows:

$$w(e) = \begin{cases} \text{RR}(x_i) & i = j, \\ \text{ALL}(\{x_i, x_j\}) & i < j. \end{cases}$$

Observe that a 2-DUTYSC solution C_1, \dots, C_m for X induces a (perfect) matching whose weight is the lifetime of the solution. A matching $M \subseteq E$ also induces a 2-DUTYSC solution whose lifetime is the weight of the matching. Hence, the weight of a maximum weight matching in G is the optimal 2-DUTYSC lifetime of X . \square

The algorithm that is described in the theorem is henceforth referred to as MATCH.

6.2 Worst-case analysis

6.2.1 RoundRobin

We have shown in Theorem 5.2 that ROUNDROBIN is a $\frac{3}{2}$ -approximation algorithm for SENSOR STRIP COVER. Since ROUNDROBIN schedules are duty cycle schedules and any k -DUTYSC schedule is also a SENSOR STRIP COVER schedule, it follows that

Theorem 6.2. *ROUNDROBIN is a $\frac{3}{2}$ -approximation algorithm for k -DUTYSC, for every $k \geq 2$.*

In what follows, we develop detailed analysis of the performance of ROUNDROBIN vs. ALL, which we then use in the next section to prove upper bounds on the approximation ratio of MATCH in k -DUTYSC, for $k \geq 3$. This analysis is also of independent interest, in that it proves that the perfect deployment instance are the worst for ROUNDROBIN.

6.2.2 Round Robin vs. All

Assume we are given a set X of k sensors. We define

$$\gamma(X) \triangleq \frac{\text{RR}(X)}{\text{ALL}(X)}.$$

In this section we look for a lower bound on $\min_{X:|X|=k} \gamma(X)$.

Due to Theorem 6.2 it follows that $\gamma(X) \geq \frac{2}{3}$, for every k and any set X of k sensors. We prove the stronger result that the placement that minimizes the ratio is the perfect deployment, namely X_k^* . Notice that this is true for $k = 2$, since $\gamma(X_2^*) = \frac{2}{3}$.

Stretching the Instance

Our first step is to transform X into an instance X' for which $\gamma(X') \leq \gamma(X)$. This is done by pushing sensors away from $\frac{1}{2}$ so that all internal gaps are of size Δ . (See Section 2 for the definition of Δ .) If a sensor needs to be moved to the left of 0, it is placed at 0, and if it needs to move to the right of 1, it is placed at 1.

Definition 6.3. *For a given instance X , let j be the sensor whose location is closest to $\frac{1}{2}$. Then we define the stretched instance X' of X as follows:*

$$x'_i = \begin{cases} \max\{0, x_j - (j - i)\Delta\} & i < j, \\ x_j & i = j, \\ \min\{1, x_j + (i - j)\Delta\} & i > j. \end{cases}$$

Lemma 6.4. *Let X' be the stretched instance of X . Then, $\gamma(X') \leq \gamma(X)$.*

Proof. Sensors only get pushed away from $\frac{1}{2}$, and thus their ROUNDROBIN lifetime only decreases. Thus, $\text{RR}(X') \leq \text{RR}(X)$. By definition, Δ must equal either d_0 , d_n or the length of the largest internal gap in X . However neither d_0 nor d_n can be larger in X' than it was in X , since no sensors move

closer to $\frac{1}{2}$. Moreover, by construction the length of the largest internal gap in X' is Δ . Hence $\Delta' \leq \Delta$, and $\text{ALL}(X') \geq \text{ALL}(X)$. \square

Perfect Deployment is the Worst

By Lemma 6.4, it suffices to consider only stretched instances. The next step is to show that the worst stretched instance is in fact the perfect deployment.

Given a stretched instance X' with k sensors, let k_{out} be the number of sensors located on either 0 or 1, and let $k_{in} \triangleq k - k_{out}$ be the number of sensors in $(0, 1)$. Notice that $k_{in} \geq 1$. Also notice that if $k = 1$, then $\text{RR}(X') = \text{ALL}(X')$, and recall that $\gamma(X) \geq \frac{2}{3} = \gamma(X_2^*)$. Therefore we may assume that $k = k_{in} + k_{out} \geq 3$.

Let a and b be the gaps between 0 and the leftmost sensor not at 0, and 1 and the rightmost sensor not at 1, respectively. For reasons of symmetry we assume, w.l.o.g., that $a \leq b$. Hence, $\lfloor k_{in}/2 \rfloor$ sensors are located in $(0, \frac{1}{2}]$ and $\lfloor k_{in}/2 \rfloor$ sensors are located in $(\frac{1}{2}, 1)$.

The stretched deployment X' can be described as follows:

$$X' = \{0^{\lfloor k_{out}/2 \rfloor}, a, a + \Delta', \dots, a + (k_{in} - 1)\Delta' = 1 - b, 1^{\lfloor k_{out}/2 \rfloor}\}$$

Note that $\Delta' = \frac{1-a-b}{k_{in}-1}$, if $k_{in} \geq 2$. Otherwise, if $k_{in} = 1$, then $k_{out} > 1$ and

$\Delta' = b$. The ROUNDROBIN lifetime of X' is:

$$\text{RR}(X') = k_{out} + \sum_{i=0}^{\lceil k_{in}/2 \rceil - 1} \frac{1}{1 - (a + i\Delta')} + \sum_{i=0}^{\lfloor k_{in}/2 \rfloor - 1} \frac{1}{1 - (b + i\Delta')}. \quad (6.1)$$

We distinguish three cases:

1. $X' = \{a, a + \Delta', \dots, a + (k_{in} - 1)\Delta' = 1 - b\}$, where $a \in [0, \Delta'/2]$ and $b \in (0, \Delta'/2]$.

Let Ω_0 be the set of all such instances. Note that $k_{out} = 0$, if $a > 0$, and that $k_{out} = 1$, if $a = 0$. However, notice that (6.1) holds if we use $k_{out} = 0$, even for the case where $a = 0$.

2. $X' = \{a, a + \Delta', \dots, a + (k_{in} - 2)\Delta' = 1 - b, 1\}$, where $a \in [0, \Delta'/2]$, $b \in [\Delta'/2, \Delta']$.

Let Ω_1 be the set of all such instances. Note that $k_{out} = 1$, if $a > 0$, and that $k_{out} = 2$, if $a = 0$. However, notice that (6.1) holds if we use $k_{out} = 1$, even for the case where $a = 0$.

3. $X' = \{0^{\lfloor k_{out}/2 \rfloor}, a, a + \Delta', \dots, a + (k_{in} - 1)\Delta' = 1 - b, 1^{\lceil k_{out}/2 \rceil}\}$, where $a, b \in [0, \Delta']$ and $k_{out} \geq 2$.

Let $\Omega_{k_{out}}$ be the set of all such instances that correspond to k_{out} . Note that if $a = 0$, (6.1) holds if we use $k_{out} + 1$ in place of k_{out} and $a = \Delta'$.

Lemma 6.5. $\gamma(X')$ has no local minima in $\Omega_{k_{out}}$, for any k_{out} .

Proof. First assume that $k_{in} \geq 2$. Due to (6.1) and Observation 2.6 we have that

$$\begin{aligned} \gamma(X') &= \frac{\Delta'}{2} \left[k_{out} + \sum_{i=0}^{\lceil k_{in}/2 \rceil - 1} \frac{1}{1 - (a + i\Delta')} + \sum_{i=0}^{\lfloor k_{in}/2 \rfloor - 1} \frac{1}{1 - (b + i\Delta')} \right] \\ &= \frac{k_{out}}{2} \cdot \Delta' + \sum_{i=0}^{\lceil k_{in}/2 \rceil - 1} f_{k_{in}}^{(i)}(a, b) + \sum_{i=0}^{\lfloor k_{in}/2 \rfloor - 1} f_{k_{in}}^{(i)}(b, a), \end{aligned}$$

where $f_{k_{in}}^{(i)}(a, b) = \frac{\Delta'}{2 - 2(a + i\Delta')}$. Since

$$\frac{\partial \Delta'}{\partial a} = -\frac{1}{k_{in} - 1},$$

it follows that

$$\begin{aligned} \frac{\partial f_{k_{in}}^{(i)}(a, b)}{\partial a} &= \frac{(2 - 2(a + i\Delta'))(\frac{\partial \Delta'}{\partial a}) - \Delta'(-2 - 2i\frac{\partial \Delta'}{\partial a})}{(2 - 2(a + i\Delta'))^2} \\ &= \frac{2(\frac{\partial \Delta'}{\partial a}(1 - a) + \Delta')}{4(1 - (a + i\Delta'))^2} \\ &= \frac{1}{2(k_{in} - 1)} \cdot \frac{-b}{(g_{k_{in}}^{(i)}(a, b))^2}, \end{aligned}$$

where $g_{k_{in}}^{(i)}(a, b) = 1 - (a + i\Delta')$. Thus,

$$\begin{aligned} \frac{\partial \gamma(X')}{\partial a} &= \frac{-k_{out}}{2(k_{in} - 1)} \\ &+ \frac{1}{2(k_{in} - 1)} \left[\sum_{i=0}^{\lceil k_{in}/2 \rceil - 1} \frac{-b}{(g_{k_{in}}^{(i)}(a, b))^2} + \sum_{i=0}^{\lfloor k_{in}/2 \rfloor - 1} \frac{-a}{(g_{k_{in}}^{(i)}(b, a))^2} \right]. \end{aligned}$$

Since $a = b = k_{out} = 0$ is not possible for any domain $\Omega_{k_{out}}$, we have that

$\frac{\partial \gamma(X')}{\partial a} < 0$. Hence $\gamma(X')$ decreases as a increases. An analogous calculation shows that the same is true for $\frac{\partial \gamma(X')}{\partial b}$. Thus, since neither $\frac{\partial \gamma(X')}{\partial a}$ nor $\frac{\partial \gamma(X')}{\partial b}$

can be zero at any point in the interior of the domain $\Omega_{k_{out}}$, $\gamma(X')$ has no local minima. Finally, any minima must occur when both a and b are as large as possible within the domain $\Omega_{k_{out}}$.

It remains to consider the case where $k_{in} = 1$. Since $k_{out} > 1$, we have that $\Delta' = b$. Hence,

$$\gamma(X') = \frac{b}{2} \left(k_{out} + \frac{1}{b} \right) = \frac{1}{2}(bk_{out} + 1),$$

which means that $\frac{\partial \gamma(X')}{\partial b} > 0$. Hence, $\gamma(X')$ decreases as b decreases. It follows that the minima occurs when $a = b = \frac{1}{2}$. \square

Let $\gamma_k^* = \gamma(X_k^*)$. We show that, for any fixed k , $\gamma(X)$ reaches its minimum at $X = X_k^*$.

Theorem 6.6. $\min_{X:|X|=k} \gamma(X) = \gamma_k^*$.

Proof. We prove that $\min_{X \in \Omega_{k_{out}}} \gamma(X) \geq \gamma_k^*$ by induction on k_{out} .

For the base case, if $X' \in \Omega_0$, then by Lemma 6.5, $\gamma(X')$ achieves its minimum on the boundary of Ω_0 , when a and b are as large as possible, namely for $a = b = \Delta/2$. In this case, $X' = X_k^*$. Thus, for all $X' \in \Omega_0$, $\gamma(X') > \gamma_k^*$ if $X \neq X_k^*$.

For the inductive step, let $X \in \Omega_{k_{out}}$, for $k_{out} \geq 1$, and assume that $\min_{X \in \Omega_{k_{out}-1}} \gamma(X) \geq \gamma_k^*$. By Lemma 6.5 it follows that $\gamma(X')$ achieves its

minimum on the boundary of $\Omega_{k_{out}}$. If $k_{out} = 1$ (and $k_{in} \geq 2$), then the minimum is when $a = \Delta/2$ and $b = \Delta$, namely for $X' = \{\frac{\Delta}{2}, \frac{3\Delta}{2}, \dots, 1 - \Delta, 1\}$. By symmetry, this instance has the same ratio as the instance $X'' = \{1 - x : x \in X'\}$, which is in Ω_0 with parameters $a = 0$ and $b = \frac{\Delta}{2}$. If $k_{out} > 1$, then the minimum is when $a = \Delta$ and $b = \Delta$. In this case $X' \in \Omega_{k_{out}-1}$ with parameters $a = 0$ and $b = \Delta$. Hence by the induction hypothesis we have that $\gamma(X') \geq \gamma_k^*$. \square

Properties of γ_k^*

In this section we explore γ_k^* as a function of k . Observe that for even k we have that

$$\gamma_k^* = \frac{1}{2k} \cdot 2 \sum_{i=1}^{k/2} \frac{2k}{2k+1-2i} = 2 \sum_{i=1}^{k/2} \frac{1}{2k+1-2i} = 2 \sum_{i=k/2+1}^k \frac{1}{2i-1},$$

and for odd k we have that

$$\gamma_k^* = \frac{1}{2k} \left[2 + 2 \sum_{i=1}^{(k-1)/2} \frac{2k}{2k+1-2i} \right] = \frac{1}{k} + 2 \sum_{i=(k+1)/2}^{k-1} \frac{1}{2i+1}.$$

Lemma 6.7. γ_k^* satisfies the following:

- (i) $\gamma_k^* \leq \gamma_{k+2}^*$, for every even k .
- (ii) $\gamma_k^* \geq \gamma_{k+2}^*$, for every odd k .
- (iii) $\gamma_k^* \geq \gamma_{k+1}^*$, for every odd k .

Proof. Due to the convexity of the function $f(z) = \frac{1}{z}$, we have that for even k ,

$$\begin{aligned}\gamma_{k+2}^* - \gamma_k^* &= 2 \sum_{i=k/2+2}^{k+2} \frac{1}{2i-1} - 2 \sum_{i=k/2+1}^k \frac{1}{2i-1} \\ &= \frac{2}{2k+3} + \frac{2}{2k+1} - \frac{2}{k+1} \\ &> 0.\end{aligned}$$

and by the same rationale, for odd k ,

$$\begin{aligned}\gamma_k^* - \gamma_{k+2}^* &= \left(\frac{1}{k} + 2 \sum_{i=(k+1)/2}^{k-1} \frac{1}{2i+1} \right) - \left(\frac{1}{k+2} + 2 \sum_{i=(k+1)/2+1}^{k+1} \frac{1}{2i+1} \right) \\ &= \frac{1}{k} + \frac{1}{k+2} - \frac{2}{2k+1} - \frac{2}{2k+3} \\ &> 0.\end{aligned}$$

Finally, for odd k ,

$$\gamma_k^* - \gamma_{k+1}^* = \frac{1}{k} + 2 \sum_{i=(k+1)/2}^{k-1} \frac{1}{2i+1} - 2 \sum_{i=(k+1)/2+1}^{k+1} \frac{1}{2i-1} = \frac{1}{k} - \frac{2}{2k+1} > 0.$$

□

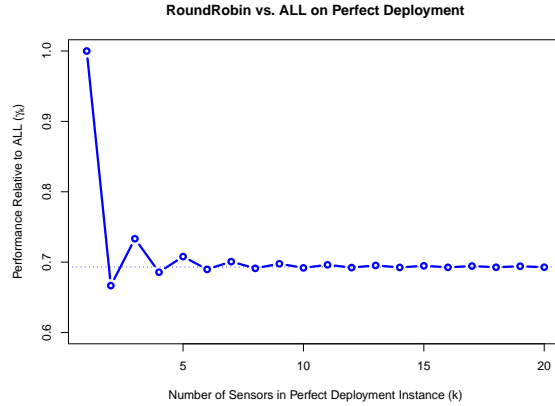
Lemma 6.8. $\lim_{k \rightarrow \infty} \gamma_k^* = \ln 2$.

Proof. Observe that for both even and odd k 's we have that

$$\gamma_k^* \geq \sum_{i=k+1}^{2k} \frac{1}{i} = H_{2k} - H_k \quad \text{and} \quad \gamma_k^* \leq \sum_{i=k}^{2k-1} \frac{1}{i} = H_{2k-1} - H_{k-1},$$

k	γ_k^*	Approx
1	1	1
2	$\frac{2}{3}$	0.6667
3	$\frac{11}{15}$	0.7333
4	$\frac{24}{35}$	0.6857
5	$\frac{223}{315}$	0.7079
6	$\frac{478}{693}$	0.6898
7	$\frac{6313}{9009}$	0.7007
8	$\frac{4448}{6435}$	0.6912
\vdots	\vdots	\vdots
∞	$\ln 2$	0.6931

(a) Exact and approximate values of γ_k^* .



(b) γ_k^* is an alternating sequence that converges to $\ln 2$

Figure 6.1: Tabular and graphical representation of small values of γ_k^* . Note that $\gamma^* < \ln 2$ for k even, and $\gamma^* > \ln 2$ for k odd.

where H_k the k th Harmonic number. It follows that

$$\lim_{k \rightarrow \infty} \gamma_k^* = \lim_{k \rightarrow \infty} (H_{2k} - H_k) = \ln 2 .$$

□

The table in Figure 6.1(a) contains several values of γ_k^* , whose convergence is also depicted graphically in Figure 6.1(b).

6.2.3 Match

By Theorem 6.2, ROUNDROBIN is a $\frac{3}{2}$ -approximation algorithm for k -DUTYSC for every $k \geq 2$. In this section we analyze the performance of MATCH in k -DUTYSC for $k \geq 3$. Recall that MATCH finds the best solution among

those using shifts of size at most 2. Since MATCH is more powerful than ROUNDROBIN, its approximation ratio is at most $\frac{3}{2} = 1.5$. We show that the approximation ratio of MATCH in k -DUTYSC is at most $\frac{35}{24} \approx 1.458$. We also provide lower bounds: $\frac{15}{11} \approx 1.364$ and for $k \geq 4$, and $\frac{6}{5} = 1.2$, for $k = 3$. At the end of the section we discuss ways to improve the analysis of MATCH.

Upper Bound

We use our analysis of ROUNDROBIN vs. ALL to obtain an upper bound on the performance of MATCH for k -DUTYSC.

Theorem 6.9. *MATCH is a $\frac{35}{24}$ -approximation algorithm for k -DUTYSC, for every $k \geq 3$.*

Proof. Let X be a k -DUTYSC instance and let C_1, \dots, C_m be an optimal solution for X . Since for every C_j with $|C_j| \leq 2$, $\text{MATCH}(C_j) = \text{OPT}_k(C_j)$, we construct an alternative solution by splitting to singletons every subset C_j such that $|C_j| > 2$. For any such C_j , by Theorem 6.6 we know that $\text{RR}(C_j) \geq \gamma_{C_j}^* \text{ALL}(C_j)$. Moreover, by Lemma 6.7 we know that $\gamma_{C_j}^* \geq \min_{k \geq 3} \gamma_k^* = \gamma_4^* = \frac{24}{35}$. The lemma follows. \square

We can generalize this approach to find upper bounds on the performance of OPT_k in n -DUTYSC, for $k \leq n$.

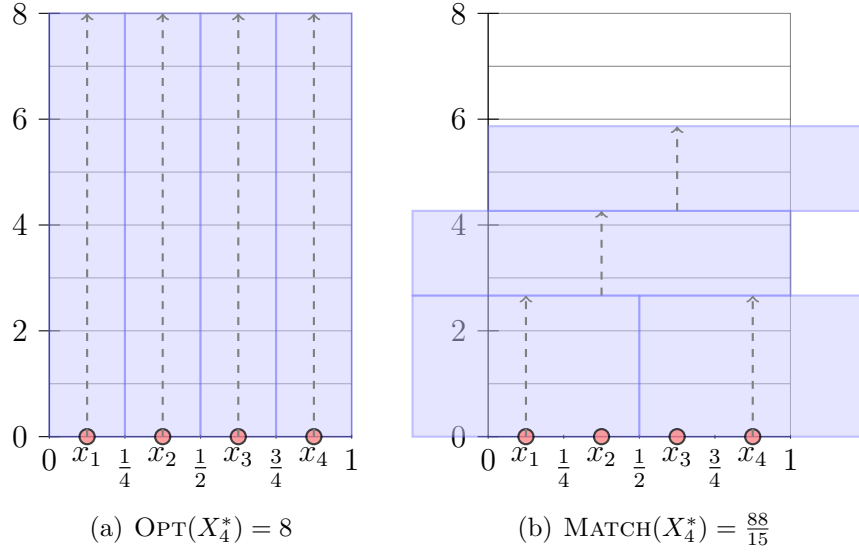


Figure 6.2: Depiction of $\frac{15}{11}$ lower bound for MATCH, for $k \geq 4$. For the instance $X_4^* = \{\frac{1}{8}, \frac{3}{8}, \frac{5}{8}, \frac{7}{8}\}$, OPT produces a lifetime of 8, while MATCH achieves only $\frac{88}{15}$.

Lemma 6.10. $\text{OPT}_k(X) \geq \gamma_\ell^* \cdot \text{OPT}_n(X)$, where ℓ is the smallest even integer larger than k .

Proof. Since $\ell > k$, OPT_k is not optimal on instances with ℓ sensors. The result then follows from the fact that γ_ℓ^* increases monotonically for even ℓ (see Figure 6.1). □

Hence, the approximation ratio of an algorithm that solves k -DUTYSC is at most $1/\gamma_\ell^*$. Lemma 6.8 implies that the least upper bound achievable via this technique is $1/\ln 2 \approx 1.4427$.

Lower Bound

We show that the approximation ratio of `MATCH` is no better than $\frac{15}{11}$, for $k \geq 4$, and $\frac{6}{5}$ for $k \geq 3$.

Lemma 6.11. $\text{MATCH}(X_4^*) = \frac{11}{15} \text{OPT}_k(X_4^*)$, for every $k \geq 4$.

Proof. Consider the instance $X_4^* = \{\frac{1}{8}, \frac{3}{8}, \frac{5}{8}, \frac{7}{8}\}$, and observe that the `ROUNDROBIN` lifetime of the sensors on the outside is $\text{RR}(\frac{1}{8}) = \text{RR}(\frac{7}{8}) = \frac{8}{7}$, while the lifetime is $\text{RR}(\frac{3}{8}) = \text{RR}(\frac{5}{8}) = \frac{8}{5}$ for the sensors in the middle. Perhaps surprisingly, any sensible pairing of the sensors achieves a lifetime of $\frac{8}{3}$. Thus, one `MATCH` solution is to pair the outside sensors for a lifetime of $\frac{8}{3}$, and then run `ROUNDROBIN` on the middle sensors successively, for an additional lifetime of $2 \cdot \frac{8}{5}$, as depicted in Figure 6.2. Thus, the total lifetime is

$$\text{MATCH}(X_4^*) = \frac{8}{3} + 2 \cdot \frac{8}{5} = \frac{88}{15} .$$

The lemma follows, since $\text{OPT}_k(X_4^*) = \text{ALL}(X_4^*) = 8$, for every $k \geq 4$. \square

We have a weaker lower bound for $k = 3$.

Lemma 6.12. $\text{MATCH}(X_3^*) = \frac{5}{6} \text{OPT}_k(X_3^*)$, for every $k \geq 3$.

Proof. `MATCH` has 2 duty cycles: $\{\frac{1}{6}, \frac{5}{6}\}, \{\frac{1}{2}\}$, which cover U for 3 and 2 time units, respectively. Hence, $\text{MATCH}(X_3^*) = 5$, while $\text{ALL}(X_3^*) = 6$. \square

Conjecture 6.13. $\text{MATCH}(X) \geq \frac{5}{6}\text{OPT}_3(X)$. Moreover, for every $k \geq 4$, $\text{MATCH}(X) \geq \frac{11}{15}\text{OPT}_k(X)$.

For some positive integer d , let $D_d = \{\frac{i}{d} : i \in \{0, 1, \dots, d\}\}$ be a discretization of $[0, 1]$. Clearly, as $d \rightarrow \infty$, D becomes a close approximation of $[0, 1]$. Using brute force, we checked all 680 possible instances in D_{16}^3 and all 2380 possible instances in D_{16}^4 , and found no instance X for which $\text{MATCH}(X) < \frac{5}{6}\text{OPT}_3(X)$ in the first case, nor any for which $\text{MATCH}(X) < \frac{11}{15}\text{OPT}_4(X)$ in the second case.

Again, due to Lemma 6.7, we can generalize Lemma 6.11 to find lower bounds on the performance of OPT_k in n -DUTYSC.

Lemma 6.14. For every $\ell \leq n$, $\text{OPT}_k(X) \geq \frac{\text{OPT}_k(X_\ell^*)}{2^\ell} \cdot \text{OPT}_n(X)$.

Note that for $k = 1, 2$ we recover the $2/3$ and $11/15$ bounds demonstrated previously.

6.3 Average-case analysis

6.3.1 Sweep and Nest

One way to improve the analysis of Algorithm MATCH would be to first prove that perfect deployments are worst with respect to MATCH (as they are with respect to ROUNDROBIN), and then to analyze $\gamma_k^2 = \text{MATCH}(X_k^*)/\text{ALL}(X_k^*)$.

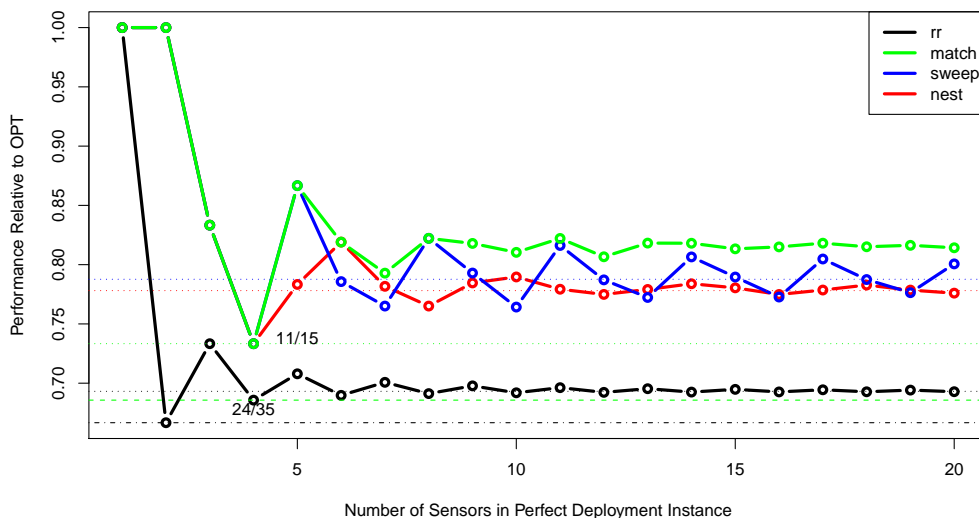


Figure 6.3: Performance of ROUNDROBIN, MATCH, SWEEP, and NEST compared to ALL on perfect deployments. The performance of MATCH approaches an experimentally determined value of approximately 0.816.

Our experiments show that γ_k^2 seem to converge to approximately 0.816, which is significantly higher than $\lim_{k \rightarrow \infty} \gamma_k^* = \ln 2 \approx 0.693$. See Figure 6.3.

We would like to evaluate $\lim_{k \rightarrow \infty} \gamma_2^*$. However, since MATCH explicitly evaluates each pair of sensors, it is not a simple algorithm. Nevertheless, we obtain lower bounds on the limit by analyzing simple heuristics. For a given instance X , let $L = X \cap [0, \frac{1}{3}]$, $M = X \cap (\frac{1}{3}, \frac{2}{3})$, and $R = X \cap [\frac{2}{3}, 1]$. Note that $|L| = |R|$ for the perfect deployment instance X_k^* of any size. Consider the following heuristics:

- SWEEP: Sensors in L and R are paired to create shifts of size two,

starting with the leftmost sensor in L and the *leftmost* sensor in R .

Any remaining sensors are put in size one shifts.

- NEST: Sensors in L and R are paired to create shifts of size two, starting with the leftmost sensor in L and the *rightmost* sensor in R .

Any remaining sensors are put in size one shifts.

Note that the two preceding heuristics return valid (albeit suboptimal) 2-DUTYSC solutions.

Using similar arguments as in Section 6.2.2 we can show that:

$$\frac{\text{SWEEP}(X_n^*)}{\text{ALL}(X_n^*)} \approx 1/2 + H_{4n/3} - H_n$$

and

$$\frac{\text{NEST}(X_n^*)}{\text{ALL}(X_n^*)} \approx (H_n - H_{n/2})/2 + (H_{2k/3} - H_{k/2})/2 + (H_{4n/3} - H_n) ,$$

which means that

$$\lim_{n \rightarrow \infty} \frac{\text{SWEEP}(X_n^*)}{\text{ALL}(X_n^*)} = \frac{1}{2} + \ln(4/3) \approx 0.788$$

and

$$\lim_{n \rightarrow \infty} \frac{\text{NEST}(X_n^*)}{\text{ALL}(X_n^*)} = \frac{1}{2} \ln 2 + \frac{1}{2} \ln(4/3) + \ln(4/3) \approx 0.778 .$$

A comparison of the performance of both SWEEP and NEST on perfect deployments to the performance of ALL is given in Figure 6.3.

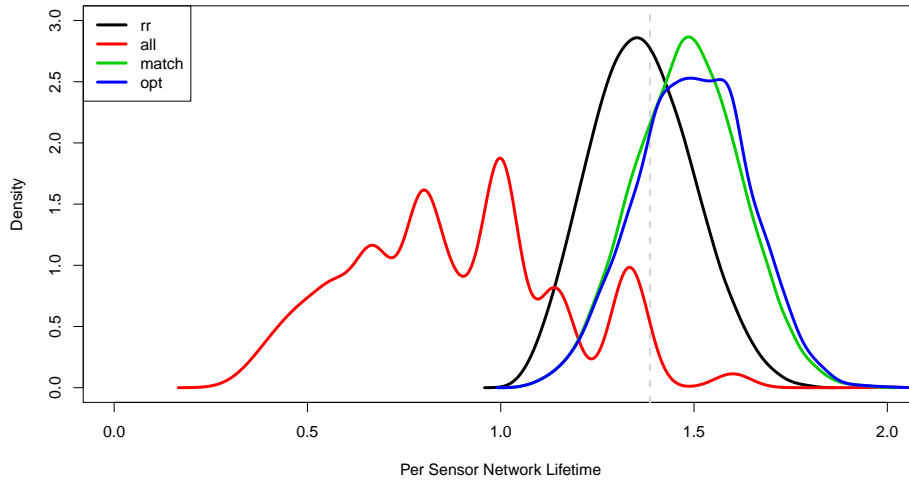


Figure 6.4: Comparison of density estimates of the per sensor network lifetime for various algorithms in 4-DUTYSC. The lifetime of each algorithm was computed for all 2380 possible instances of 4 sensors on a grid of size 16.

In Figure 6.4 below, we show the density of the per sensor network lifetime for ROUNDROBIN, ALL, MATCH, and OPT, for all 2380 possible instances with 4 sensors over a grid of size 16. X_4^* was the only instance X found for which $\text{MATCH}(X) \leq \frac{11}{15}\text{OPT}_4(X)$. Moreover, for about 82% of the instances, $\text{MATCH}(X) = \text{OPT}(X)$, and the mean approximation ratio between MATCH and OPT was 1.0078. Meanwhile, the average lifetime per sensor for MATCH was 1.483, which surpasses the corresponding figure for ROUNDROBIN of $2 \ln 2 = 1.386$.

6.3.2 Fault Tolerance

In this section we extend our analysis to incorporate a fault tolerance model, in which each sensor may fail to activate with probability $p \in [0, 1]$. We assume that failures occur randomly and independently. If any sensor in a shift fails to activate, then the entire coverage lifetime of that shift is lost. Under these assumptions, we can make the following observation about the expected lifetime of an algorithm.

Observation 6.15. *For any shift $C \subseteq X$, the expected lifetime of the shift is $(1 - p)^{|C|} \cdot \text{ALL}(C)$.*

The expected lifetime of a solution $C_1, \dots, C_m \subseteq X$ is thus $\sum_i (1 - p)^{|C_i|} \cdot \text{ALL}(C_i)$. In the fault tolerant version of k -DUTYSC our goal is to find a solution $C_1, \dots, C_m \subseteq X$ such that $|C_i| \leq k$ with maximum expected lifetime. Let $\text{OPT}_k^p(X)$ denote the expected lifetime of an optimal k -DUTYSC solution of X .

Theorem 6.16. *Fault tolerant 2-DUTYSC can be solved in polynomial time.*

Proof. The proof is similar to the proof of Theorem 6.1. The only difference is that the weight of an edge $e = (v_i, v'_j)$ is defined as follows:

$$w(e) = \begin{cases} (1 - p) \cdot \max\{x_i, 1 - x_i\} & i = j, \\ (1 - p)^2 \cdot \text{ALL}(\{x_i, x_j\}) & i < j. \end{cases}$$

□

We show that if the probability of failure is high enough, MATCH, or even ROUNDROBIN, compute optimal solutions.

Lemma 6.17. *If $p \geq \frac{1}{3}$, then $\mathbb{E}[\text{RR}(X)] = \text{OPT}_k^p(X)$, for every X .*

Proof. Let C_1, \dots, C_m be an optimal schedule. Consider any shift C_j for which $|C_j| \geq 2$. By Observation 6.15, the expected lifetime of that shift is at most $(1-p)^2 \cdot \text{ALL}(C_j)$. Since $\gamma(C_j) \geq \frac{2}{3}$ for every C_j (due to Theorem 4.7), we have that $\text{RR}(C_j) \geq \frac{2}{3} \text{ALL}(C_j)$. It follows that

$$(1-p)^2 \cdot \text{ALL}(C_j) \leq \frac{4}{9} \text{ALL}(C_j) \leq \frac{2}{3} \text{RR}(C_j) .$$

□

Lemma 6.18. *If $p \geq 1 - \sqrt{\gamma_3^*} \approx 0.144$, then $\mathbb{E}[\text{MATCH}(X)] = \text{OPT}_k^p(X)$, for every X .*

Proof. Let C_1, \dots, C_m be an optimal schedule. Consider any shift C_j for which $|C_j| \geq 3$. By Observation 6.15, the expected lifetime of that shift is $(1-p)^{|C_j|} \cdot \text{ALL}(C_j)$. Due to Theorem 6.6 we have that

$$(1-p)^{|C_j|} \text{ALL}(C_j) \leq \frac{(1-p)^{|C_j|-1}}{\gamma_k^*} \cdot (1-p) \text{RR}(C_j) .$$

If $|C_j| = 3$, we have that $(1-p)^{|C_j|-1} / \gamma_k^* \leq 1$, since $p \geq 1 - \sqrt{\gamma_3^*} \approx 0.144$.

Also, for $|C_j| > 3$ we have that $(1-p)^{|C_j|-1} / \gamma_k^* \leq (1-p)^3 / \gamma_4^* < 1$, since

$p > 1 - \sqrt[3]{\gamma_4^*} \approx 0.118$. It follows that if $p \geq 1 - \sqrt{\gamma_3^*}$, then there exists an optimal schedule that does not use shifts of size larger than 2. Hence, MATCH computes an optimal solution. \square

Chapter 7

Conclusion

The problems considered in this dissertation are simple to state, yet often quickly lead to non-trivial technicalities. Our emphasis on the *lifetime* of a network addresses application considerations more directly than previous attempts to maximize the number of covers, or conserve energy in a single cover. Moreover, our study of the expected performance of algorithms in the average case is likely to be more relevant in many realistic applications.

7.1 Summary

In this dissertation we have studied the SENSOR STRIP COVER problem and several related variants, and found bounds for the worst-case and average-case approximation ratios of several algorithms. In particular, we have shown that ROUNDROBIN, which is perhaps the simplest possible algorithm, has a tight approximation ratio of $\frac{3}{2}$ in SENSOR STRIP COVER, with the perfect

deployment instances being the worst-case inputs. Along the way, we have demonstrated a polynomial-time algorithm for RADSC, and proven that ONCESC is NP-hard. For the case where all of the sensors have the same initial battery charge, we have developed an algorithm that will achieve a lifetime that is within 12% of the theoretical maximum against uniform random sensor deployments. For the k -DUTYSC problem, we have found an optimal polynomial-time algorithm for $k \leq 2$, and found approximation bounds for its performance for $k > 2$.

7.2 Open Problems

There remain many unanswered questions about SENSOR STRIP COVER and its variants.

7.2.1 Non-uniform batteries

When the batteries do not all have the same initial charge, we showed that ONCESC is NP-hard, but it remains to be seen whether the same is true for SENSOR STRIP COVER. Moreover, the fact that both ONCESC and TIMESCS are NP-hard, while RADSC can be solved in polynomial time, suggests that hardness comes from setting the activation times.

Future work may include finding algorithms with better approximation

ratios for either problem. However, note that we have eliminated duty cycle algorithms as candidates.

7.2.2 Uniform batteries

Requiring that all initial battery charges be the same seems to make the problem easier. For example, `TIMESC` is NP-hard in general, but admits a polynomial-time solution in the uniform battery case [11]. In Section 4.1, we proved that `ONCESC` is NP-hard and that for a specific instance a gap exists between the optimal lifetime in `SENSOR STRIP COVER` and the optimal lifetime in `ONCESC`. Both of these proofs required non-uniform batteries, and thus it is not clear whether `SENSOR STRIP COVER` with uniform batteries is in fact NP-hard, or even if a gap exists between it and `ONCESC`.

In Chapter 5, we found the expected performance of several algorithms under the assumptions that the sensor locations were distributed uniformly at random, and the initial battery charges were the same. It is not clear how those results would change under different assumptions about the distribution of sensor locations *or* battery charges. This type of analysis could be performed for any pair of probability distributions with finite support.

In this average-case analysis, we measured the quality of approximation against the theoretical maximum determined by the total battery charge of

the system $(2n)$. This is only an upper bound on the performance of OPT , since it is clear that there are instances in which even OPT cannot achieve a lifetime of $2n$ (e.g. $X = (\frac{1}{3})$). Thus, the expected performance of OPT against uniform random deployment is unknown. It may be the case that the optimized log-ROUNDROBIN algorithm is in fact very close to being optimal.

While 1-DUTYSC can be solved trivially by ROUNDROBIN , and we have shown that 2-DUTYSC can be solved in polynomial time using MATCH , it remains open whether $k\text{-DUTYSC}$ is NP-hard, for $k \geq 3$. It would also be interesting to close the gap between the upper and lower bounds on the approximation ratio of MATCH , for $k \geq 3$. We offered one possible direction to improving the upper bound in Section 6.3.

7.2.3 Variations

This problem setting is rich, in that there are many variations in the setup that can alter the resulting analysis dramatically. In this paper 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. $\tau = b/\rho^2$). This may correspond more closely to a realistic dissipation of power.

We have assumed that once deployed, the sensors cannot be moved. Phelan, et al. [30] have considered a variation of SENSOR STRIP COVER in which the sensors are allowed to be moved, at a certain cost. This line of research could have interesting connections with our findings, particularly the discovery that perfect deployment results in the worst-case approximation ratio for ROUNDROBIN.

Finally, while we have restricted our attention to a one-dimensional coverage region, one could consider a variety of similar problems in higher dimensions. For example, one might keep the sensor locations restricted to the line, but consider a two-dimensional coverage region (i.e. *beach* coverage). Conversely, the sensors could be located in the plane, while the coverage region remains one-dimensional (i.e. *road* coverage). Of course, from an application point-of-view the most important variation would allow both the sensor locations and the coverage region to be two-dimensional.

Bibliography

- [1] Zoë Abrams, Ashish Goel, and Serge A. Plotkin. Set k -cover algorithms for energy efficient monitoring in wireless sensor networks. In *IPSN*, pages 424–432, 2004.
- [2] Greg Aloupis, Jean Cardinal, Sébastien Collette, Stefan Langerman, David Orden, and Pedro Ramos. Decomposition of multiple coverings into more parts. *Discrete & Computational Geometry*, 44(3):706–723, 2010.
- [3] Helmut Alt, Esther M. Arkin, Hervé Brönnimann, Jeff Erickson, Sándor P. Fekete, Christian Knauer, Jonathan Lenchner, Joseph S. B. Mitchell, and Kim Whittlesey. Minimum-cost coverage of point sets by disks. In *SoCG*, pages 449–458, 2006.
- [4] Aung Aung. Distributed algorithms for improving wireless sensor network lifetime with adjustable sensing range. Master’s thesis, Georgia State University, 2007. Paper 42.
- [5] Amotz Bar-Noy and Ben Baumer. Maximizing network lifetime on the line with adjustable sensing ranges. In *ALGOSENSORS*, pages 28–41, 2011.
- [6] Amotz Bar-Noy, Ben Baumer, and Dror Rawitz. Changing of the guards: Strip cover with duty cycling. In *SIROCCO*, 2012. To appear.
- [7] Amotz Bar-Noy, Ben Baumer, and Dror Rawitz. Set it and forget it: Approximating the set once strip cover problem. *Arxiv preprint arXiv:1204.1082v1*, 2012.

- [8] Amotz Bar-Noy, Theodore Brown, Matthew P. Johnson, and Ou Liu. Cheap or flexible sensor coverage. In *DCOSS*, pages 245–258, 2009.
- [9] Piotr Berman, Gruia Calinescu, C. Shah, and Alexander Zelikovsky. Efficient energy management in sensor networks. *Ad Hoc and Sensor Networks, Wireless Networks and Mobile Computing*, 2:71–90, 2005.
- [10] Vittorio Bilò, Ioannis Caragiannis, Christos Kaklamanis, and Panagiotis Kanellopoulos. Geometric clustering to minimize the sum of cluster sizes. In *ESA*, pages 460–471, 2005.
- [11] Adam L. Buchsbaum, Alon Efrat, Shaili Jain, Suresh Venkatasubramanian, and Ke Yi. Restricted strip covering and the sensor cover problem. In *SODA*, pages 1056–1063, 2007.
- [12] Mihaela Cardei and Ding-Zhu Du. Improving wireless sensor network lifetime through power aware organization. *Wireless Networks*, 11(3):333–340, 2005.
- [13] Mihaela Cardei and Jie Wu. Coverage in wireless sensor networks. In *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, chapter 19, pages 432–446. CRC Press, 2004.
- [14] Mihaela Cardei, Jie Wu, and Mingming Lu. Improving network lifetime using sensors with adjustable sensing ranges. *IJSNet*, 1(1/2):41–49, 2006.
- [15] Mihaela Cardei, Jie Wu, Mingming Lu, and Mohammad O. Pervaiz. Maximum network lifetime in wireless sensor networks with adjustable sensing ranges. In *WiMob (3)*, pages 438–445, 2005.
- [16] Erin W. Chambers, Sándor P. Fekete, Hella-Franziska Hoffmann, Dimitri Marinakis, Joseph S. B. Mitchell, Venkatesh Srinivasan, Ulrike Stege, and Sue Whitesides. Connecting a set of circles with minimum sum of radii. pages 183–194, 2011.
- [17] William J. Cook, William H. Cunningham, and Alexander Schrijver William R. Pulleyblank. *Combinatorial Optimization*. John Wiley & Sons, 1998.

- [18] Akshaye Dhawan, Chinh T. Vu, Alexander Zelikovsky, Yingshu Li, and Sushil K. Prasad. Maximum lifetime of sensor networks with adjustable sensing range. In *SNPD*, pages 285–289, 2006.
- [19] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [20] Matt Gibson, Gaurav Kanade, Erik Krohn, Imran A. Pirwani, and Kasturi R. Varadarajan. On clustering to minimize the sum of radii. In *SODA*, pages 819–825, 2008.
- [21] Matt Gibson and Kasturi R. Varadarajan. Decomposing coverings and the planar sensor cover problem. In *FOCS*, pages 159–168, 2009.
- [22] Gaurav S. Kasbekar, Yigal Bejerano, and Saswati Sarkar. Lifetime and coverage guarantees through distributed coordinate-free sensor activation. In *MOBICOM*, pages 169–180, 2009.
- [23] Evangelos Kranakis, Danny Krizanc, Lata Narayanan, and Ladislav Stacho. Maximum interference of random sensors on a line. In *SIROCCO*, pages 197–210, 2010.
- [24] Santosh Kumar, Ten-Hwang Lai, and József Balogh. On k -coverage in a mostly sleeping sensor network. *Wireless Networks*, 14(3):277–294, 2008.
- [25] Nissan Lev-Tov and David Peleg. Polynomial time approximation schemes for base station coverage with minimum total radii. *Computer Networks*, 47(4):489–501, 2005.
- [26] Michael Mitzenmacher and Eli Upfal. *Probability and computing - randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.
- [27] János Pach. Covering the plane with convex polygons. *Discrete & Computational Geometry*, 1:73–81, 1986.
- [28] János Pach and Géza Tóth. Decomposition of multiple coverings into many parts. *Comput. Geom.*, 42(2):127–133, 2009.

- [29] Mark A. Perillo and Wendi Beth Heinzelman. Optimal sensor management under energy and reliability constraints. In *Wireless Communications and Networking*, volume 3, pages 1621–1626. IEEE, 2003.
- [30] Brian Phelan, Peter Terlecky, Amotz Bar-Noy, Theodore Brown, and Dror Rawitz. Should I stay or should I go? maximizing lifetime with relays. In *DCOSS*, 2012. To appear.
- [31] Anwar Saipulla, Cédric Westphal, Benyuan Liu, and Jie Wang. Barrier coverage of line-based deployed wireless sensor networks. In *INFOCOM*, pages 127–135, 2009.
- [32] Sasha Slijepcevic and Miodrag Potkonjak. Power efficient organization of wireless sensor networks. In *ICC '01: IEEE International Conference on Communications*, volume 2, pages 472–476. IEEE, 2001.
- [33] Yoshiaki Taniguchi, Tomoya Kitani, and Kenji Leibnitz. A uniform air-drop deployment method for large-scale wireless sensor networks. *IJS-Net*, 9(3/4):182–191, 2011.
- [34] Jiong Wang and Sirisha Medidi. Energy efficient coverage with variable sensing radii in wireless sensor networks. page 61, 2007.
- [35] Lan Wang and Yang Xiao. A survey of energy-efficient scheduling mechanisms in sensor networks. *MONET*, 11(5):723–740, 2006.
- [36] Eric W. Weisstein. q -series. From MathWorld—A Wolfram Web Resource. Visited on 04/23/2012.
- [37] Jie Wu and Shuhui Yang. Coverage issue in sensor networks with adjustable ranges. In *ICPP Workshops*, pages 61–68, 2004.