2-2014

Lean, Green, and Lifetime Maximizing Sensor Deployment on a Barrier

Peter Michael Terlecky
Graduate Center, City University of New York

Lean, Green, and Lifetime Maximizing Mobile Sensor Deployment on a Barrier

by

Peter Michael Terlecky

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.

2014

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

Linda Keen

Date — Executive Officer

Amotz Bar-Noy

Joseph Malkevitch

Christina Zamfirescu

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

Lean, Green, and Lifetime Maximizing Mobile Sensor Deployment on a Barrier

by

Peter Michael Terlecky

Advisor: Amotz Bar-Noy

Mobile sensors are located on a barrier represented by a line segment, and each sensor has a single energy source that can be used for both moving and sensing. Sensors may move once to their desired destinations and then coverage/communication is commenced. The sensors are collectively required to cover the barrier or in the communication scenario set up a chain of communication from endpoint to endpoint. A sensor consumes energy in movement in proportion to distance traveled, and it expends energy per time unit for sensing in direct proportion to its radius raised to a constant exponent. The first focus is of energy efficient coverage. A solution is sought which minimizes the sum of energy expended by all sensors while guaranteeing coverage for a predetermined amount of time. The objective of minimizing the maximum energy expended by any one sensor is also considered. The dual model is then studied. Sensors are equipped with batteries and a solution is sought which maximizes the coverage lifetime of the network, i.e. the minimum lifetime of any sensor. In both of these models, the variant where sensors are equipped with predetermined radii is also examined. Lastly, the problem of maximizing the lifetime of a wireless connection between a transmitter and a receiver using mobile relays is considered. These problems are mainly examined from the point of view of approximation algorithms due to the hardness of many of them.

Acknowledgements

I am extremely greatful to my advisor Amotz Bar-Noy for his thoughtful conversations, advice, direction, and support. An exceptionally large thank you goes to Dror Rawitz who has been like a second advisor to me. Your impact and influence has been tremendous. I would like to thank my committee members, Joseph Malkevitch and Christina Zamfirescu for their support and encouragement. A very big thanks goes out to the Mathematics Assistant Program Officer, Robert Landsman, for all of his help, guidance, and kindness throughout the years.

Ben Baumer and Subir Dhamoon have been great friends throughout this journey. Its been a pleasure sharing this experience with you guys, thanks for your friendship. I would be remiss without recognizing Brian Phelan, Yuqing Tang, Valia Mitsou, Yosef Alayev, Ivo Vigan, George Rabanca, Radoslaw Wojciechowski, Ali Assarpour, Matthew Johnson, Kenneth Ezirim, Nooreddin Naghibolhosseini, and Su Xing. I have greatly enjoyed our interactions, thank you for the intellectual and social colloquy.

The largest serving of gratitude is given to my mother, Stephanie Terlecky, for all the sacrifices she has undertaken for her family. I have yet to meet a harder working person. Your inspiration has been the driving force behind this dissertation. I would like to thank my father Peter Terlecky and sisters Tanya Terlecky and Christine Terlecky who have picked me up both literatively and figuratively throughout the years. A very special thank you is reserved for my uncle Stephen Szulhan and aunt Maria Szulhan who have been like second parents to me. Your altruism has been a saving grace. A large thanks to my godfather and uncle Fr. John Terlecky for his prayers and encouragement. Lastly, I would like to give a big thanks to all of my extended family and friends.

Without the help of all of you, this dissertation would not be possible. From the bottom of my heart,

Thank You All!!!

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Battery lifetime is a significant bottleneck on wireless sensor network performance. Consequently, one of the fundamental problems in sensor networks is optimizing battery usage when accomplishing fundamental tasks such as covering, monitoring, tracking, and communicating.

This work studies the problem of optimizing battery usage when covering a boundary or a barrier by mobile sensors. Examples of barrier coverage include covering borders, bridges, coastlines, railroads, supply chains, etc. Often by covering region boundaries, the benefit of protecting the interior is gained as any intrusion from the exterior can be detected.

One example of barrier coverage with the intent of protecting the interior is the United States' and now apparently India's use of unattended ground sensors (UGSs) for border patrol [1]. These sensors are planted partially underground along a border and can monitor an area of about 3 kilometers. They can detect movement by man or machine and are able to take a photo of a possible intruder and relay it back to a central commander. Maintaining the life of these sensors for as long as possible is paramount as the death of any one of these sensors exposes the interior to possible intrusion.

Sensors have also been used in barrier coverage for forest fire detection [26]. In particular, if a region is to be monitored for the spread of fire into the region, sensors can be placed along the region's border in order to detect the spread of fire from the exterior to the interior. Sensors may also be used in barrier coverage with the intent of monitoring the barrier. An example of coverage with just the intent of monitoring the barrier is the deployment of sensors on bridges such as the Golden Gate Bridge in San Fransisco to monitor the material health of the bridge [33].

This work also considers the problem of maintaining a straight-line chain of communication from a transmitter to a receiver for as long as possible using mobile relays (sensors which relay information). Information is transferred from relay to relay until it reaches the receiver. The radius of communication of a sensor is the distance to its rightmost neighbor (the length of transmission from the sensor to the next adjacent sensor). The problem is theoretically equivalent to coverage of a barrier with mobile sensors, where sensors are only capable of one-sided coverage, coverage to the right. Placing relays along a line of communication from transmitter to receiver can significantly lengthen communication lifetime as the transmission distance of the transmitter is drastically shortened from being to the reciever to being to the next adjacent relay. Mobility can be additionally harnessed to further optimize transmission lifetime over strictly stationary solutions.

One application of such a model is creating a chain of communication from a transmitter to a receiver located deep in a narrow mine. Mobile relays can move along a track or path to specified locations to relay the signal which may not otherwise reach the receiver. These relays are battery powered and battery is drained by moving and by relaying the signal. Where should the relays move to, in order to optimize communication lifetime?

Most previous research has implicitly assumed a two battery model, in which there is a battery for movement and a separate battery for sensing/communicating. These works attempt to optimize on only one of these parameters. This work considers a model where energy is consumed by sensing and movement from a single battery source as is most commonly the architecture [3] and hence has most relevance to application. The optimization attempts to balance the cost of sensing/communicating with the cost of movement.

The set up and sense framework is mainly studied. In this framework, sensors first move to their desired locations and then coverage/communication commences. Movement drains energy in proportion to distance traveled, and sensing/communicating drains energy *per time unit* in direct proportion to a sensor's radius raised to a constant exponent. The energy drained in sensing for a predetermined amount of time from a desired position is the sum of the energy drained in movement to that position and the energy drained in sensing/communicating with its coverage radius. Inversly, the lifetime of a sensor is the initial energy less the energy drained in movement divided by the energy per time unit drained in sensing.

1.2 Problems

Chapter 2: "Green" Barrier Coverage explores the most energy efficient way of covering a straight-line barrier for a predetermined amount of time with mobile sensors given some initial arrangement of these sensors on the barrier. Two notions of energy-efficiency are explored; the first is minimizing the total amount of energy expended by all sensors in the network, and the second is minimizing the maximium amount of energy expended by any one sensor.

The input consists of the initial positions of the sensors, a specified coverage time t, a cost of movement of a per unit distance traveled -also called the cost of *friction*, and an exponent of coverage drain, α. A constraint is that the barrier must be covered for t time units. In SumVar, the objective is to find destinations for the sensors and radii which minimize the sum of energy spent. We seek the solution which spends the least total energy while covering for t time units. For MaxVar, destinations and radii must also be found, but the objective is to minimize the maximum energy used. A solution is sought which does not deplete the energy of any one sensor by too much.

For the following two problems, along with the initial positions and a specified coverage time of t being given, an assigned coverage radius for each sensor is also given. If a sensor is used in coverage, it must cover with its assigned radius. In SumFix, the goal is to determine destinations and select a subset of coverage radii (covering sensors) which minimize the sum of the energy consumed. In MaxFix, the objective is to determine destinations and select a subset of coverage radii which minimize the maximum energy consumed by any one sensor. When a specific friction parameter a is considered, a subscript a is added to the problem name. For example, SumVar$_0$ is SumVar with $a = 0$.

Chapter 3: Maximizing Barrier Coverage Lifetime entertains the dual objective to minimizing the maximum energy (battery) consumed for a specified lifetime. Namely, it concerns determining the destinations and radii which maximize the lifetime of coverage given initial battery powers and initial positions for the sensors. Coverage expires with the death of a sensor as a gap in coverage is consequently created. A solution is therefore sought which maximizes the minimum lifetime of any sensor.

The input consists of initial positions of the sensors as well as initial battery powers. Also, included in the input is a cost of movement of a per unit distance traveled, and an exponent of coverage drain, α. In the Barrier Coverage with Variable Radii problem (abbreviated BCVR) the goal is to find a deployment and radii that maximize the coverage lifetime. In the Barrier Coverage with Fixed Radii problem (BCFR) an assigned coverage radius for each sensor is also given. If a sensor is active, it must use its assigned coverage radius. The objective is to find a deployment of the sensors and select a subset of coverage radii (covering sensors) that maximize the coverage lifetime of the network.

Chapter 4: Maximizing Communication Lifetime engages the topic of maximizing the lifetime of communication between a transmitter and receiver using mobile relays on a barrier. The input consists of the initial positions for the relays, the initial battery powers of the relays, a distance D

between transmitter and receiver, a cost of movement of a per unit distance traveled, and an exponent of coverage drain, α. In the MaxFD problem, the objective is to assign destinations to the sensors, with transmission ranges corresponding to distances between adjacent nodes, such that the lifetime until the first death of a sensor is maximized. Here it is assumed that once a relay dies, the communication chain is broken. In the MaxTL problem, it is assumed that relays can be deployed multiple times and consequently readjust their transmission ranges after each redeployment. The objective is to assign time-dependent destinations to the sensors such that the transmission lifetime, the length of time the transmitter can communicate with the receiver, is maximized.

1.3 Related Work

It is noted that this dissertation is a compendium of the published works [40, 10] and the preprint [9]. The discussion of related work is naturally organized into two sections: 1) papers that focus on coverage, and 2) papers that focus on communication.

1.3.1 Coverage

Most previous research has implicitly assumed a two battery model, in which there is a separate battery for movement and a separate battery for sensing. These works attempt to optimize on only one of the parameters.

Movement Optimization. When only moving is optimized (covering energy is ignored), the problem is equivalent to having an infinite covering battery. In our model such problems can be described by setting $t = 0$. Czyzowicz et al. [18] addressed the problem of deploying sensors on a line barrier while minimizing the maximum distance traveled by any sensor, where radii are uniform. This is $\textsc{MaxFix}_1$ with uniform radii and $t = 0$ in our model. (In this case we may assume without loss of generality that $a = 1$.) They provided a polynomial time algorithm for this problem. It follows that there is a polynomial time algorithm for $\textsc{MaxFix}$ with $t = 0$ and uniform radii, for any $a \in (0, \infty)$. They also gave an NP-hardness result for a variant of this problem with non-uniform radii in which one sensor is assigned a predetermined position. Chen et al. [17] gave a polynomial time algorithm for the more general case in which the sensing radii are non-uniform, namely for $\textsc{MaxFix}_1$ with $t = 0$ and improved upon the running time for $\textsc{MaxFix}_1$ with uniform radii and $t = 0$.

Czyzowicz et al. [19] studied the problem of covering a barrier with mobile sensors with the goal of minimizing the sum of distances traveled by all sensors. This problem is a special case of $\textsc{SumFix}_1$ in which $t = 0$ (without loss of generality $a = 1$). They present a polynomial time algorithm for $\textsc{SumFix}_1$ with uniform radii and $t = 0$, and they also showed that the non-uniform problem cannot be approximated within a factor of c, for any constant c.

There are other problem in which movement is optimized. We list several examples. Mehrandish et al. [37] considered the same model with the objective of minimizing the number of sensors which must move to cover the barrier. Drobev et al. [21] considers the problem of covering a set of barriers attempting to optimize movement costs. Tan and Wu [42] presented improved algorithms for minimizing the max distance traveled and minimizing the sum of distances traveled when sensors must be positioned on a circle in regular n-gon position. The problems were initially considered by Bhattacharya et al. [12].

Coverage Optimization. In many papers it is assumed that sensors are static, and the goal is to minimize sensing energy. Li et al. [36] presented a polynomial time algorithm for $\textsc{SumFix}_\infty$ ($\textsc{SumFix}$ with $a = \infty$ implying static sensors) and an FPTAS for SumVar$_\infty$ with $\alpha = 1$. They can also show that SumVar$_\infty$ with $\alpha = 1$ is NP-hard [35]. Agnetis et al. [2] considered an extension of SumVar$_\infty$ with $\alpha = 2$. They gave a closed form solution for this problem if the coverage set is given, and developed a branch-and-bound algorithm and heuristics. Some papers explored discrete coverage of points on the barrier by static sensors, see, e.g., [34, 8].

Chambers et. al [15] looked at the problem of given points in the plane, finding an assignment of radii which forms a connected set and for which the sum of the radii to a given power is minimized. The problem of maximizing the lifetime of a network of static sensors has also been considered. Buchsbaum et al. [13] and Gibson and Varadarajan [24] considered the Restricted Strip Cover problem in which sensors are static and radii are fixed, and sensors may start covering at any time. Bar-Noy et al. [5, 6, 7] studied problems involving stationary sensors with variable radii that may start covering at any time.

1.3.2 Communication.

Several papers address the problem of optimal placement of relays between a source and a destination to optimize other objective functions besides lifetime. Appuswamy *et al.* [4] optimize the capacity of the implied logical channel between the source and the destination under some interference model. This paper allows only grid placements of relays with the same, non-adjustable transmission range and does not consider movement cost.

Goldenberg et al. [25] prove that, for equal battery levels, the optimal locations for the relays on the line are equidistant apart and present an algorithm for moving nodes to their optimal locations using information from 1-hop neighbors only. They prove that their algorithm preserves the connections between all the relays.

Jiang *et al.* [28] develop a number of algorithms to speed up the rate of convergence and, hence, minimize the number of iterations required to move each of the nodes to their optimal locations. We extend the above works by taking into consideration the remaining battery life in each node and the effect that different mobility costs will have on the optimal locations of the nodes.

El-Moukaddem *et al.* [22] consider using mobile relays to enhance the lifetimes of existing network routes between static nodes in a 2D plane. Each relay can assist a single link between two nodes. They include realistic costs for movement and transmission and take into consideration the battery power remaining in each of the nodes.

The numerous uses of mobility in WSNs are discussed in Francesco *et* *al.* [20] along with the challenges that arise when mobile nodes are introduced to a network such as maintaining connectivity.

Along with mobile relays, other uses of mobility include mobile sinks and data MULEs. Controlled sink mobility is considered by Basagni *et al.* [11], while predictable sink mobility is considered by Chakrabarti *et al.* [14], Chandra *et al.* [16] and Mhatre *et al.* [38]. Another approach considered in the literature is allowing the sink to move randomly as in Juang *et al.* [29] and Kim *et al.* [32]. Data MULEs were introduced by Shah *et al.* [41] and are further explored by Jain *et al.* [27].

1.4 Contributions

"Green" Barrier Coverage. Our results for SumVar are provided in Section 2.2. We present an $O(n)$ time algorithm for SumVar_0 (SumVar with $a = 0$) and an FPTAS[1] for SumVar, for any a. Our FPTAS is based on the FPTAS for SumVar_∞ (SumVar with $a = \infty$) with $\alpha = 1$ by Li et al. [36]. However, we introduce several new ideas in order to cope with sensor mobility and with $\alpha > 0$. In particular we show that there exist a non-swapping optimal solution and we use the fact that the optimal value for

[1]An FPTAS is a fully polynomial time approximation scheme where for any $\epsilon > 0$, there exists a solution which is within a factor of $1 + \epsilon$ of the optimal with running time polynomial in the size of the input and $1/\epsilon$.

$a = 0$ serves as a lower bound for case where $a > 0$. Section 2.3 deals with MaxVar. We give $O(n)$ time algorithms for MaxVar_0 and for MaxVar_∞. We also present an FPTAS for MaxVar when $a \in (0, \infty)$.

In Section 2.4 we study MaxFix. We provide an $O(n \log n)$ time algorithm for MaxFix_0. We show that MaxFix is strongly NP-hard for every $a \in (0, \infty)$ and $\alpha \geq 1$. We also show that MaxFix is NP-hard, for every $a \in (0, \infty)$ and $\alpha \geq 1$, even if $x \in (\frac{1}{2})^n$, and this result implies that it is NP-hard to find an optimal ordering. We study SumFix in Section 2.5. We show that SumFix_0 is NP-hard for $\alpha = 1$, and provide an FPTAS for SumFix_0 for any α. We show that SumFix cannot be approximated within a factor of $O(n^c)$, for any constant c, unless P=NP. We also prove that SumFix with uniform radii can be approximated to within an additive approximation $\varepsilon > 0$, for any constant $\varepsilon > 0$.

Maximizing Barrier Coverage Lifetime. In Section 3.2 we show that the static ($a = \infty$) and fully dynamic ($a = 0$) cases are solvable in polynomial time for both BCFR and BCVR. On the negative side, we show in Section 3.6 that it is NP-hard to approximate BCFR (i) within any multiplicative approximation factor, or (ii) within an additive factor ε, for some $\varepsilon > 0$, in polynomial time unless P=NP, for any $a \in (0, \infty)$ and $\alpha \geq 1$, even if $x = p^n$, where $p \in (0, 1)$. We also show that BCVR is strongly NP-hard for any $a \in (0, \infty)$ and $\alpha \geq 1$.

In Section 3.3 we consider constrained versions of BCFR and BCVR in which the input contains a total order on the sensors that the solution is required to satisfy. We design a polynomial-time algorithm for the decision problem of BCFR in which the goal is to determine whether a given lifetime t is achievable and to compute a solution with lifetime t, if t is achievable. We design a similar algorithm for BCVR that, given t, determines whether t is achievable assuming precise calculations. Using these decision algorithms we present parametric search algorithms for constrained BCFR and BCVR. We consider the case where the sensors are initially located on the edges of the barrier (i.e., $x \in \{0, 1\}^n$) in Section 3.4. For both BCFR and BCVR, we show that, for every candidate lifetime t, we may assume a final ordering of the sensors. (The ordering depends only on the battery powers in the BCVR case, and it can be computed in polynomial time in the BCFR case.) Using our decision algorithms, we obtain parametric search algorithms for this special case. In Section 3.5 we show that the uniform instances of both BCFR (uniform batteries and radii) and BCVR (uniform batteries) have non-swapping optimal solutions and thus can be solved with the parametric search algorithms.

Maximizing Communication Lifetime. In Section 4.2 we give optimal solutions to MaxFD and MaxTL where there is no movement cost. In Section 4.3 we consider MaxTL without swapping. In particular, it is shown that there is no justification for movement unless a relay dies, implying that a single deployment suffices when maximizing lifetime of first death. We also show that in any optimal solution the transmitter must be the last node to die. In Section 4.4 we consider MaxFD restricted to grid points. We provide two algorithms which are optimal for the discrete problem (relays must be deployed on grid points) of maximizing the lifetime of first death. The first algorithm is a dynamic programming algorithm, while the second conducts a binary search on lifetimes for the optimum. In Section 4.5, for the case where the batteries are not too small, it is shown that both algorithms are FPTASs for the general problem, as their running times are polynomial in the size of the input and in $1/\varepsilon$ and their solutions are within a factor of $(1+\varepsilon)$ from the optimal for a given ϵ. It is shown that the dynamic programming algorithm has better running time on grid points, but the binary search algorithm has better running time for the general problem. Section 4.6 presents results of MaxFD when relays are initial located at basestation (the endpoints of the transmission interval). It is shown that if all relays are initially located at the basestations, then there is an ordering which is *dominant*, and thus we may solve the discrete version of this instance optimally and give an FPTAS if batteries are not too small by ordering with such an order and applying the dynamic programming or binary search algorithms.

Chapter 2

"Green" Barrier Coverage

This chapter is partitioned into the following sections. Section 2.1 formally defines the notation, model, and problems. Our results for SumVar are provided in Section 2.2. Section 2.3 deals with MaxVar. In Section 2.4 we study MaxFix. Results for SumFix are provided in Section 2.5. Finally, section 2.6 lists some open problems and future directions.

2.1 Preliminaries

Model. There are n mobile sensors initially located on a barrier represented by the interval $[0, 1]$.The initial position of sensor i is denoted by x_i, and let $x = (x_1, \ldots, x_n)$ be the initial position vector. The system follows the *set-up and sense* model [18, 17, 19, 40, 10], where sensors first move to their desired destinations and then begin sensing. The destination position of sensor i is denoted by y_i, and let $y = (y_1, \ldots, y_n)$ be the deployment vector. The system

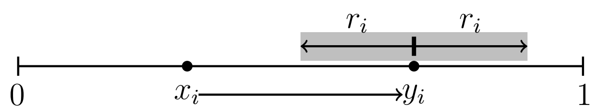


Figure 2.1: Sensor i moves from x_i to y_i and covers the interval $[y_i - r_i, y_i + r_i]$.

works in two phases. In the *deployment phase*, sensor i moves from its initial position x_i to its destination y_i. This phase is said to occur at time 0. In the *covering phase*, sensor i is assigned a sensing radius r_i and covers the interval $[y_i - r_i, y_i + r_i]$; let $r = (r_1, \ldots, r_n)$ be the radii vector. This interval is called the *covering interval* of sensor i. An example of movement and coverage by one sensor is given in Figure 2.1. It is required that the sensors collectively cover the unit interval, i.e. $[0, 1] \subseteq \sum_i [y_i - r_i, y_i + r_i]$. A pair (y, r) is called *feasible* if it covers $[0, 1]$.

Sensor i expends energy both in moving and sensing. Given a deployment point y_i, the energy sensor i spends in movement is proportional to the distance i has traveled, and given by $a|x_i - y_i|$, where a is the constant of proportionality, also referred to as the cost of *friction*. The energy sensor i uses for sensing *per time unit* is r_i^α, where $\alpha \geq 1$ a constant. Given a solution radii assignment r, a sensor i is called *active* if $r_i > 0$, and otherwise it is called *inactive*. Given a deployment y, a radii assignment r, and a time t, sensor i needs at least

$$E_i^t(y, r) \stackrel{\text{def}}{=} a|y_i - x_i| + t r_i^\alpha$$

energy in order to maintain coverage of the interval $[y_i - r_i, y_i + r_i]$ for t time. (We usually omit t and write $E_i(y, r)$, when t is clear from the context.)

Problems. Given an instance (x, t), we seek a feasible pair (y, r) that is "green" with energy expenditure or energy-efficient. We consider two objective functions: (i) minimizing the sum of the energy used, namely we would like to find a pair (y, r) that minimizes $\sum_i E_i(y, r)$; and, (ii) minimizing the maximum amount of energy expended, i.e., we look for (y, r) that minimizes $\max_i E_i(y, r)$. We also consider two variants of the problem that are distinguished by whether the radii are given as part of the input. In the *variable radii* case the goal is to find a radii assignment r such that $r_i \geq 0$, for every i, while in the *fixed radii* case the input contains a radii vector ρ, and the goal is to find a radii assignment r, such that $r_i \in \{0, \rho_i\}$, for every i. Thus, we get four variants:

1. Minimum Sum Energy with Variable Radii (SumVar)
2. Minimum Sum Energy with Fixed Radii (SumFix)
3. Minimum Max Energy with Variable Radii (MaxVar)
4. Minimum Max Energy with Fixed Radii (MaxFix)

Sometimes when we consider a specific friction parameter a we add a subscript a to the problem name. For example, SumVar_0 stands for the problem of finding a pair (y, r), where r is variable, that minimizes $\sum_i E_i(y, r)$ for $a = 0$.

Given a SumFix or a MaxFix instance (x, ρ, t), we say that the radii vector ρ is *uniform* if $\rho_i = \rho_j$, for every sensors i and j. Also, we assume that $\sum_i 2\rho_i \geq 1$ throughout the paper, since otherwise there is no feasible solution.

A solution (y, r) (or a deployment y) is called *non swapping* if $x_i < x_j$ implies $y_i \leq y_j$.

2.2 Minimum Energy Sum with Variable Radii

In this section we consider SumVar. We show that SumVar$_0$ can be solved in linear time, and the main result of the section is an FPTAS for the case where $a > 0$. Our FPTAS is based on the approach of Li et al. [36] who gave an FPTAS for the case where $a = \infty$ and $\alpha = 1$. We note that several new and non-trivial ideas were introduced in order to cope with mobility and with $\alpha > 1$.

2.2.1 Zero Friction

We start with the case where $a = 0$.

Theorem 2.2.1. SumVar$_0$ *can be solved in* $O(n)$ *time with optimum* $nt\left(\frac{1}{2n}\right)^{\alpha}$.

Proof. Given a SumVar$_0$ assignment (x,t), let $r_i = \frac{1}{2n}$, for all i, and let $y_i = \sum_{j=1}^{i-1} 2r_j + r_i$, for every i. We show that (y,r) is an optimal solution. This solution assignment clearly covers $[0,1]$. Consider any radii assignment $r' \neq r$ that covers the line. It follows that $\sum_i r_i \geq \frac{1}{2} = \sum_i r_i$. Since sensors are free to move without energy consumption, by Jensen's Inequality we have that $\sum_i E_i(y,r) = nt\left(\frac{1}{2n}\right)^\alpha \leq \sum_i t(r'_i)^\alpha = \sum_i E_i(y',r')$. Thus, (y,r) is optimal as well. Finally, notice that (y,r) can be computed in linear time. $\square$

Observe that the optimal solution may only decrease as a decreases. Hence, $nt\left(\frac{1}{2n}\right)^\alpha$ may serve as a lower bound for the case where $a > 0$. We use this lower bound in the sequel.

2.2.2 Non-Zero Friction

We now turn to the non-zero friction case. We present an FPTAS that is obtained by the following approach. We first show that any SumVar instance has a non-swapping optimal solution. Then, we show that we pay an approximation factor of $(1+\varepsilon)$ for only considering a certain family of solutions. Finally, we design a dynamic programming algorithm that computes an optimal solution within this family.

Lemma 2.2.2. *Any* SumVar *instance has a non-swapping optimal solution.*

Proof. Let (x,t) be a SumVar instance, and let (y,r) be an optimal solution for (x,t) that minimizes the number of swaps. If there are no swaps, then we are done. Otherwise, we show that the number of swaps may be decreased. If there are swaps, then there must exist at least one swap due to a pair of adjacent sensors. Let i and j be such sensors. Consider a solution (y',r') swapping locations and radii of sensors i and j in (y,r), i.e., with $y'_i = y_j$ and $r'_i = r_j$, $y'_j = y_i$ and $r'_j = r_i$, and $y'_k = y_k$ and $r'_k = r_k$, for every $k \neq i,j$.

Clearly, the barrier $[0,1]$ remains covered. We show that the energy sum does not decrease, since the total distance traveled by the sensors does not increase. If both sensors move to the right in y, then we have that $x_i < x_j \leq y_j < y_i$. In this case $(y'_i - x_i) + (y'_j - x_j) = (y_i - x_i) + (y_j - x_j)$, and we are done. The case where both sensors move to the left is symmetric. Suppose that i moves to the right while j moves to the left. If $x_i \leq y_j < y_i \leq x_j$ or $y_j < x_i < x_j < y_i$, then both i and j move less in y'. If $x_i \leq y_j \leq x_j < y_i$, then $(y_i - x_i) + (x_j - y_j) \geq y_i - x_i + y_j - x_j = (y'_i - x_i) + (y'_j - x_j)$. The case where $y_j < x_i \leq y_i \leq x_j$ is symmetric. It follows that (y',r') is an optimal solution with less swaps than (y,r). A contradiction. □

Let m be a large integer to be determined later. We consider solutions in which the sensors must be located on certain points. More specifically, we define $\mathcal{G} = \{x_i : i \in \{1, \ldots, n\}\} \cup \{\frac{j}{m} : j \in \{0, \ldots, m\}\}$. The points in $\mathcal{G}$ are called *grid points*. Let $g_0, \ldots, g_{n+m}$ be a ordering of grid points such that $g_i \leq g_{i+1}$. Given a point $p \in [0,1]$, let p^+ be the left-most grid point to the right of p, namely $p^+ = \min\{g \in \mathcal{G} : g \geq p\}$. Similarly, $p^- = \max\{g \in \mathcal{G} : g \leq p\}$ is the right-most grid point to the left of p.

A solution (y, r) is called *discrete* if (i) $y_i \in \mathcal{G}$, for every sensor i, and (ii) for every $j \in \{1, \ldots, n+m\}$ there exists a sensor i such that $[g_{j-1}, g_j] \subseteq [y_i - r_i, y_i + r_i]$. That is, in a discrete solution sensors must be deployed at grid points. Also, each segment between grid points is contained in the covering interval of some sensor.

We show that we lose a factor of $(1+\varepsilon)$ by focusing on discrete solutions.

Lemma 2.2.3. *Let $\varepsilon \in (0,1)$, and let $m = 8\lceil \alpha\mu/\varepsilon \rceil$, where $\mu = 2n/\varepsilon^{1/\alpha}$. Then, for any non-swapping solution (y, r) there exists a non-swapping discrete solution (y', r') such that $\sum_i E_i(y', r') \leq (1+2\varepsilon)\sum_i E_i(y, r)$.*

Proof. Given a SumVar instance (x, t) and a solution (y, r) we construct a discrete solution (y', r') as follows. First, each sensor i is taken back from y_i to the direction of x_i, until it hits a grid point: $y'_i = y_i^+$ if $y_i \leq x_i$ and $y'_i = y_i^-$ otherwise. Also, the radii are increased to compensate for the new deployment, and in order to obtain a discrete solution: $r'_i =$

$\max\{y'_i - (y_i - r_i)^-, (y_i + r_i)^+ - y'_i\}$. The pair (y', r') is feasible, since $[y_i - r_i, y_i + r_i] \subseteq [y'_i - r'_i, y'_i + r'_i]$ by construction. Moreover, notice that if $(g_j, g_{j+1}) \cap [y_i - r_i, y_i + r_i] \neq \emptyset$, then $[g_j, g_{j+1}] \subseteq [y'_i - r'_i, y'_i + r'_i]$. Hence, (y', r') is discrete. We also note that (y', r') is non-swapping.

It remains to show that $\sum_i E_i(y', r') \leq (1 + 2\varepsilon) \sum_i E_i(y, r)$. Since y'_i can only be closer than y_i to x_i, we have that $|y'_i - x_i| \leq |y_i - x_i|$. In addition, the radius of sensor i may increase due to its movement from y_i to y'_i and due to covering up to grid points. Hence, $r'_i \leq r_i + \frac{2}{m}$.

If $r_i \geq \frac{1}{2\mu}$, then $r'_i \leq r_i + \frac{2}{m} \leq r_i + \frac{\varepsilon}{4\alpha\mu} \leq r_i(1 + \frac{\varepsilon}{2\alpha})$. Hence,

$$\begin{aligned}
E_i(y', r') =& a|y'_i - x_i| + t(r'_i)^\alpha \\
\leq& a|y_i - x_i| + t r_i^\alpha (1 + \frac{\varepsilon}{2\alpha})^\alpha \\
\leq& (1 + \frac{\varepsilon}{2\alpha})^\alpha E_i(y, r) \\
\leq& e^{\varepsilon/2} E_i(y, r) \ .
\end{aligned}$$

Otherwise, if $r_i < \frac{1}{2\mu}$, then $r'_i \leq r_i + \frac{2}{m} \leq \frac{1}{2\mu} + \frac{\varepsilon}{4\alpha\mu} \leq \frac{1}{\mu}$. Hence,

$$E_i(y', r') = a|y'_i - x_i| + t(r'_i)^\alpha \leq a|y_i - x_i| + t\frac{1}{\mu^\alpha} = a|y_i - x_i| + t\frac{\varepsilon}{2^\alpha n^\alpha}$$

Putting it all together we get

$$\sum_i E_i(y', r') \leq e^{\varepsilon/2} \sum_i E_i(y, r) + nt\frac{\varepsilon}{2^\alpha n^\alpha} \leq (1+\varepsilon) \sum_i E_i(y, r) + \varepsilon \text{OPT} \quad (2.1)$$

where the second inequality follows from (i) $e^{\varepsilon/2} \leq 1+\varepsilon$, for any $\varepsilon \in (0,1)$, and (ii) $\textsc{opt} \geq nt\frac{1}{(2n)^\alpha}$, as observed in Theorem 2.2.1. □

Lemma 2.2.3 implies that there is a discrete non-swapping solution which is $(1+\varepsilon)$-approximate. We now present a dynamic programming algorithm for finding the optimal discrete non-swapping solution.

Lemma 2.2.4. *There exists an $O(nm^4)$ time algorithm that finds the optimal discrete non-swapping solution.*

Proof. The dynamic programming table Π is constructed as follows. The entry $\Pi(i,\ell,k)$, where i is a sensor, $\ell \in \{0,\ldots,n+m\}$, and $k \in \{0,\ldots,n+m\}$ stands for the minimum energy sum needed by a non-swapping discrete solution that uses the first i sensors, such that the i-th sensor is located at $[0,g_\ell]$, to cover the interval $[0,g_k]$. Observe that the size of the table is $O(nm^2)$. Also, the optimum is given by $\Pi(n,n+m,n+m)$.

In the base case $\Pi(0,\ell,0)=0$, for all ℓ. Otherwise, we have

$$\Pi(i,\ell,k) = \min_{\ell'\leq\ell}\left\{a|g_{\ell'}-x_i| + \min\left\{\Pi(i-1,\ell',k), \min_{k'<k}\{\Pi(i-1,\ell',k')+tr_i^\alpha\}\right\}\right\}, \tag{2.2}$$

where $r_i = \max\{g_{\ell'}-g_{k'}, g_k-g_{\ell'}\}$. Notice that $g_{\ell'}-g_{k'}$ or $g_k-g_{\ell'}$ may be negative, but not both. The first term in (2.2) is the energy that is required by sensor i to arrive at $g_{\ell'}$. Then, we have two options, either i participates in the cover or it does not. In the first case, sensors 1 to $i-1$ need to cover $[0, g_k]$, and $i-1$ may stand anywhere in $[0, g_{\ell'}]$. Otherwise, r_i is determined such that i can cover $[g_{k'}, g_k]$ while standing at $g_{\ell'}$. The rest of the barrier, i.e. $[0, g_{k'}]$ is covered by sensors 1 to $i-1$, and $i-1$ may stand anywhere in $[0, g_{\ell'}]$.

Computing each entry takes $O(m^2)$ time. Hence, the total running time is $O(nm^4)$. We note that the above algorithm computes the minimum energy sum, but may also be used to compute the solution that achieves this value using standard techniques. □

Lemma 2.2.3 and the above dynamic programming algorithm lead to an FPTAS for SumVar.

Corollary 2.2.5. *There is an FPTAS for* SumVar *whose running time is* $O(n^5/\varepsilon^{4(1+1/\alpha)})$.

In the case of static sensors (i.e., $a = \infty$) the dynamic programming can be simplified, since there is no reason to deal with the location of the sensors. In this case we have only $O(nm)$ entries, where $\Pi(i, k)$ stands for the minimum energy sum needed by a discrete solution that uses the first i

sensor to cover the interval $[0, g_k]$. Also, (2.2) is changed to

$$\Pi(i,k) = \min\left\{\Pi(i-1,k), \min_{k'<k}\{\Pi(i-1,k') + tr_i^\alpha\}\right\} \ . \tag{2.3}$$

An entry can be computed in $O(m)$, and the total running time is $O(nm^2)$. We get the following result.

Corollary 2.2.6. *There is an FPTAS for* SumVar$_\infty$ *whose running time is* $O(n^3/\varepsilon^{2(1+1/\alpha)})$*.*

2.3 Minimum Energy Max with Variable Radii

In this section we consider MaxVar. As in SumVar, we show that MaxVar$_0$ can be solved in linear time, and provide an FPTAS for the case where $a > 0$. We also give a linear time algorithm for the case where $a = \infty$.

2.3.1 Zero Friction

Theorem 2.3.1. MaxVar$_0$ *can be solved in* $O(n)$ *time.*

Proof. We show that the optimal radii assignment for MaxVar is $r_i = \frac{1}{2n}$, for all i. This radii assignment clearly covers $[0, 1]$.

Consider any radii assignment $r' \neq r$ that covers the line. Since $r' \neq r$, it follows that there exists a sensor i for which $r'_i > r_i$. It follows that $E(y', r') \geq E_i(y', r') > E_i(y, r) = E(y, r)$ (where y and y' are the deployments that correspond to r and r'). Hence, r is optimal for MaxVar. □

As in SumVar, the optimal value $t\left(\frac{1}{2n}\right)^{\alpha}$ for $a = 0$ may serve as a lower bound for the case where $a > 0$.

2.3.2 Infinite Friction

Let $\Delta = \max\left\{x_1, 1 - x_n, \frac{1}{2}\max_{i=2}^{n}\{x_i - x_{i-1}\}\right\}$.

Lemma 2.3.2. *Let (x, T) be a* MaxVar$_\infty$ *instance, and let (x, r) be a feasible solution. Then,* $\max_i E_i(x, r) \geq t\Delta^{\alpha}$.

Proof. Let $E = \max_i E_i(x, r)$, and let $R = \sqrt[\alpha]{E/t}$. (x, R^n) is feasible, since $R \geq r_i$, for every i. Also, $\max_i E_i(x, R^n) = \max_i E_i(x, r')$. The prove the lemma by showing that $\max_i E_i(x, R^n) \geq t\Delta^{\alpha}$.

Consider the segment (x_i, x_{i+1}), for some sensor i. Since all radii are the same, it follows that this segment is covered by sensors i and $i+1$. The best way to cover the segment with these sensors is to let each one cover exactly half the segment. Hence, $R \geq \frac{1}{2}(x_{i+1} - x_i)$. A similar one sided argument applies for the segments $(0, x_1)$ and $(x_n, 1)$. □

We now show a solution that matches the above lower bound.

Theorem 2.3.3. MaxVar$_\infty$ *can be solved in $O(n)$ time.*

Proof. It is not hard to verify that the pair (x, Δ^n) covers $[0, 1]$. Also, $\max_i E_i(x, \Delta^n) = t\Delta^{\alpha}$, which means that it is optimal due to Lemma 2.3.2.

Finally, Δ can be computed in $O(n)$ time. $\square$

2.3.3 Non-zero Finite Friction: FPTAS

We give an FPTAS using the same approach as we used for SumVar. We first show that any MaxVar instance has a non-swapping optimal solution. Then, we show that we pay an approximation factor of $(1+\varepsilon)$ for considering non-swapping discrete solutions. Finally, we design a dynamic programming algorithm that computes an optimal non-swapping discrete solution.

We show that there is no need to consider solutions which swap sensors. Our approach is similar to the approach taken to prove Lemma 2.2.2, but this time the proof is more involved and requires case analysis. Before proving that there is a non-swapping optimal solution for any MaxVar instance we need the following definition. Given a solution (y, r) we define $d_i = |y_i - x_i|$. Also, given an energy level E, we define $\beta_i(p, E) = (E - a|p - x_i|)/t$ and $r_i(p, E) = \sqrt[\alpha]{\beta_i(p, E)}$. The radius $r_i(p, E)$ is the maximum possible radius that can be maintained for t time, assuming that i moves to p and that $E - a|p - x_i| > 0$.

We also need the following technical lemma.

Lemma 2.3.4. *Let $\eta_1, \eta_2, \gamma_1, \gamma_2 \geq 0$ such that (i) $\gamma_1 < \eta_1 \leq \eta_2$, and (ii) $\eta_1 + \eta_2 \geq \gamma_1 + \gamma_2$. Also let $\alpha \geq 1$. Then, $\sqrt[\alpha]{\eta_1} + \sqrt[\alpha]{\eta_2} \geq \sqrt[\alpha]{\gamma_1} + \sqrt[\alpha]{\gamma_2}$.*

Proof. The case where $\alpha = 1$ is immediate, so henceforth we assume that $\alpha > 1$. Let $s = \eta_1 + \eta_2$, and let $\gamma_2' = s - \gamma_1$. We prove that $\gamma_1^{1/\alpha} + (s-\gamma_1)^{1/\alpha} < \eta_1^{1/\alpha} + (s - \eta_1)^{1/\alpha}$. Since $\gamma_2' > \gamma_2$ the lemma follows.

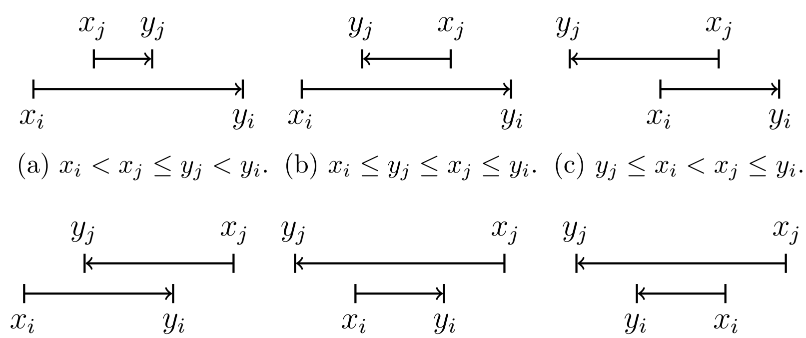
Define $f(x) = x^{1/\alpha} + (s-x)^{1/\alpha}$. The derivative is:

$$f'(x) = \frac{x^{1/\alpha - 1}}{\alpha} - \frac{(s-x)^{1/\alpha-1}}{\alpha} = \frac{1}{\alpha x^{1-1/\alpha}} - \frac{1}{\alpha(s-x)^{1-1/\alpha}} .$$

$f'(x) = 0$ implies that $x = \frac{s}{2}$ and $f'(x) > 0$ for $0 \leq x < \frac{s}{2}$. It follows that $f(x)$ is an increasing function in the interval $(0, \frac{s}{2})$. Thus we have $f(\eta_1) > f(\gamma_1)$. $\square$

Lemma 2.3.5. *Any* MaxVar *instance has a non-swapping optimal solution.*

Proof. Let (x, t) be a MaxVar instance, and let (y, r) be an optimal solution for (x, t) using maximum energy E that minimizes the number of swaps. Throughout the proof we assume that the radius of sensor i is $r_i(y_i, E)$, for all i. If there are no swaps, then we are done. Otherwise, we show that the number of swaps can be decreased. Assume to the contrary that there are swaps, and consider a swap between a pair of adjacent sensors i and j. That is, $x_i < x_j$, $y_j < y_i$, and $y_k \notin (y_j, y_i)$ for every $k \neq i, j$. There are six possible configurations for such a pair of sensors as shown in Figure 2.2.



(a) $x_i < x_j \le y_j < y_i$. (b) $x_i \le y_j \le x_j \le y_i$. (c) $y_j \le x_i < x_j \le y_i$.

(d) $x_i \le y_j < y_i \le x_j$. (e) $y_j \le x_i \le y_i \le x_j$. (f) $y_j < y_i \le x_i < x_j$.

Figure 2.2: Six configurations of a swapping pair.

If the barrier can be covered without i, then i is moved to y_j. Sensor i has enough energy for moving to y_j, since either $|y_j - x_i| \le |y_i - x_i|$ or $|y_j - x_i| \le |y_j - x_j|$. Similarly, if the barrier is covered without j, then j is moved to y_i. Sensor j has enough energy to move to y_i, since either $|y_i - x_j| \le |y_j - x_j|$ or $|y_i - x_j| \le |y_i - x_i|$. In both cases we get a solution with less swaps than (y, r), hence we may assume in the following that both sensors are necessary for covering the barrier (i.e., the removal of either i or j breaks coverage). We define the coverage interval of i and j to be $[u, v] = [y_j - r_j, y_i + r_i]$.

For each of the six cases (shown in Figure 2.2) we provide a solution (y', r') such that $y'_k = y_k$ and $r'_k = r_k$, for $k \ne i, j$, $y'_i \le y'_j$, and the interval $[u, v]$ is covered by i and j. Moreover, $E_k(y', r') \le E_k(y, r) = E$, for every k. Then, we eliminate any new swaps that may have been created by moving i and j.

The resulting solution has less swaps than (y, r), and we get a contradiction.

We start with cases (c) and (d), since they are easier. Then, we move to deal with the other cases. Newly created swaps will be considered later on.

Case (c): $y_j \leq x_i < x_j \leq y_i$.

Swap the positions and radii of sensors i and j, namely set $y'_i = y_j$, $r'_i = r_j$, $y'_j = y_i$, and $r'_j = r_i$. Observe that $[u, v]$ is covered, and no new swaps are created. Also, $d'_i \leq d_j$ and $d'_j \leq d_i$, which means that $E_i(y', r') \leq E_j(y, r)$ and $E_j(y', r') \leq E_i(y, r)$.

Case (d): $x_i \leq y_j < y_i \leq x_j$.

First, notice that since both i and j participate in the cover, we have that $y_j \leq \frac{u+v}{2} \leq y_i$. Place sensor i at $y'_i = u + r_i$ with radius $r'_i = r_i$ and sensor j at $y'_j = v - r_j$ with radius $r'_j = r_j$. Observe that $[u, v]$ remains covered as $y'_j - y'_i = y_i - y_j$. Also, we have that $y'_i \leq \frac{u+v}{2} \leq y_i$ and $y'_j \geq \frac{u+v}{2} \geq y_j$. If $y'_i \geq x_i$, then $d'_i \leq d_i$. Otherwise, if $y'_i < x_i$, then

$$d'_i = x_i - y'_i \leq y_j - y'_i < y'_j - y'_i = y_i - y_j \leq y_i - x_i = d_i \ ,$$

which means that i moves less. Hence, $E_i(y', r') \leq E_i(y, r)$. A similar argument can be made for sensor j.

Cases (a): $x_i < x_j \leq y_j < y_i$.

First, place sensor i at the location y'_i such that $y'_i - r'_i(y'_i, E) = u$, namely to the point where the left endpoint of the covering interval of i is u while using energy E. Since $x_i \leq y_j$ and $r_i(x_i, E) > r_j(y_j, E)$, we have that $x_i - r_i(x_i, E) < y_j - r_j(y_j, E) = u$. Furthermore, $y_i - r_i(y_i, E) > u$. Since the function $g_i(z) = z - r_i(z, E)$ is continuous and also strictly increasing for $z \geq x_i$, there exists one location $y'_i \in [x_i, y_i]$, for which $y'_i - r_i(y'_i, E) = u$.

Next, place sensor j at the rightmost location y'_j such that $y'_j \leq y_i$ and $y'_j - r'_j(y'_j, E) \leq y'_i + r'_i$. We know that $y_j - r_j(y_j, E) = u < y'_i + r'_i$. Also, observe that j can reach $y_i > y'_i$, since i can. Since $g_j(z) = z - r_j(z, E)$ is continuous and strictly increasing for $z \geq x_j$, we have that there exists one location $y'_j > x_j$, for which $y'_j - r_j(y'_j, E) = y'_i + r'_i$.

If $y'_j = y_i$, we get that $y'_j + r'_j > v$. Otherwise, observe that $y'_i < y_i$, $y'_i < y_j$, and $y'_j < y_i$. It follows that $d'_i < d_i$, $d'_j < d_i$, and $d'_i + d'_j < d_i + d_j$. Hence, $\beta_i(y_i, E) \leq \beta_i(y'_i, E), \beta_j(y'_j, E)$ and $\beta_i(y_i, E) + \beta_j(y_j, E) \leq \beta_i(y'_i, E) + \beta_j(y'_j, E)$. By Lemma 2.3.4 we have that $r'_i + r'_j > r_i + r_j$, and thus $y'_j + r'_j = u + 2r'_i + 2r'_j > u + 2r_i + 2r_j \geq v$.

Case (b): $x_i \leq y_j \leq x_j \leq y_i$.

In this case we have two options. First, if $d_j = x_j - y_j \geq y_j - x_i$, switch places and radii between i and j as done in case (c).

Otherwise, $d_j < y_j - x_i$. In this case place sensors i and j as done in case (a). Notice that it may be that $y'_j < x_j$. However, it is enough that $g_j(z)$ is continuous for our purposes. If $y'_j = y_i$, we get that $y'_j + r'_j > v$. Otherwise, observe that $y'_i < y_i$ and $y'_j \in (y_j, y_i)$, which means that $d'_i, d'_j < d_i$. Finally, if $y'_j \leq x_j$, then $d'_j < d_j$, and we have $d'_i + d'_j \leq d_i + d_j$. Otherwise, if $y'_j > x_j$, we have that $d'_i + d'_j = (y'_i - x_i) + (y'_j - x_j) < y_i - x_i = d_i$, since $y'_i > x_i$. Again, apply Lemma 2.3.4 to show that $r'_i + r'_j > r_i + r_j$, and it follows that $y'_j + r'_j > v$.

Case (e): $y_j \leq x_i \leq y_i \leq x_j$.

Symmetric to case (b).

Case (f): $y_j \leq y_i \leq x_i < x_j$.

Symmetric to case (a).

It remains to deal with newly created swaps.

If $y'_i < y_i$, there may be a sensor k such that $y_k \in (y'_i, y_i]$, and by moving i to y'_i a new swap is created, if $x_k < x_i$. Let $S_L = \{k : x_k < x_i \wedge y_k \in (y'_i, y_i]\}$, and let $S_R = \{k : x_k \geq x_i \wedge y_k \in (y'_i, y_i]\}$. By moving left to y'_i, i creates new swaps with sensors in S_L, but eliminates swaps with sensors in S_R. Let

$\ell = \text{argmin}_{k \in S_L}(y_k - r_k)$. If $y_\ell - r_\ell \geq u$, then the sensors in S_L are not needed for coverage and are moved left to y'_i. Consider a sensor $k \in S_L$. If $x_k \leq y'_i$, then y'_i is closer to x_i than y_k. Otherwise y'_i is closer to x_k than to x_i. Hence, in both cases k can reach y'_i. On the other hand, if $y_\ell - r_\ell < u$, it follows that $[y'_i - r'_i, y'_i + r'_i] \subset [y_\ell - r_\ell, y_\ell + r_\ell]$, which means that i is not needed for coverage, and can be moved to $\max_{k \in S_L} y_k$. In both cases all new swaps are eliminated.

The case of $y'_i > y_i$ can be treated in a symmetric manner. Also, any new swaps created by j, can be eliminated in a similar manner.

Finally, it follows that there is a solution with minimum maximum energy E with less swaps. A contradiction. $\square$

Next, we show that we can focus on non-swapping discrete solutions.

Lemma 2.3.6. *Let $\varepsilon \in (0,1)$, and let $m = 8\lceil \alpha\mu/\varepsilon \rceil$, where $\mu = 2n/\varepsilon^{1/\alpha}$. Then, for any non-swapping solution (y, r) there exists a non-swapping discrete solution (y', r') such that* $\max_i E_i(y', r') \leq (1 + 2\varepsilon) \max_i E_i(y, r)$.

Proof. The proof is almost the same as the proof of Lemma 2.2.3. The only difference is that Equation 2.1 should be replaced by

$$\max_i E_i(y', r') \leq e^{\varepsilon/2} \max_i E_i(y, r) + t\frac{\varepsilon}{2^\alpha n^\alpha} \leq (1 + \varepsilon) \max_i E_i(y, r) + \varepsilon\text{OPT} .$$

The second inequality is due to (i) $e^{\varepsilon/2} \leq 1+\varepsilon$, for any $\varepsilon \in (0,1)$, and (ii) $\textsc{opt} \geq t\frac{1}{(2n)^\alpha}$. ☐

We use dynamic programming to find the best non-swapping discrete solution.

Lemma 2.3.7. *There exists an $O(nm^4)$ time algorithm that finds the optimal non-swapping discrete solution.*

Proof. This proof is basically the same as the proof of Lemma 2.2.4. The main difference is that Equation 2.2 should be replaced by

$$\Pi(i,\ell,k) = \min_{\ell' \leq \ell} \Bigg\{ \min \Bigg\{ \max\left\{a|g_{\ell'} - x_i|, \Pi(i-1,\ell',k)\right\}, \\ \min_{k'<k} \max\left\{\Pi(i-1,\ell',k'), a|g_{\ell'} - x_i| + tr_i^\alpha\right\} \Bigg\}\Bigg\},$$

where $r_i = \max\{g_{\ell'} - g_{k'}, g_k - g_{\ell'}\}$. If i does not contribute to the cover, then we take the maximum between the energy it requires to move and the min-max energy that is required by sensors 1 to $i-1$ to cover $[0, g_k]$. If i participates in the cover, r_i is determined such that i can cover $[g_{k'}, g_k]$ while standing at $g_{\ell'}$. In this case we take the maximum between the energy consumed by i and the min-max energy that is required by sensors 1 to $i-1$ to cover $[0, g_{k'}]$, where $i-1$ may stand anywhere in $[0, \ell']$ ☐

Corollary 2.3.8. *There is an FPTAS for* MaxVar *whose running time is* $O(n^5/\varepsilon^{4(1+1/\alpha)})$.

2.4 Minimum Energy Max with Fixed Radii

In this section we study MaxFix. We give an $O(n \log n)$ time algorithms for both MaxFix_0 and MaxFix_∞. Czyzowicz et al. [18] presented an algorithm for MaxFix with uniform radii and $t = 0$. Chen et al. [17] improved upon the running time for MaxFix with uniform radii and $t = 0$ and gave a polynomial time algorithm for MaxFix with $t = 0$. We show that, for $a \in (0, \infty)$, MaxFix is NP-hard even if $x = (\frac{1}{2})^n$, and that it is strongly NP-hard when radii are non-uniform. We note that our reductions are based on the fact that $t > 0$.

2.4.1 Zero Friction

We describe a simple algorithm for solving MaxFix_0.

Theorem 2.4.1. MaxFix_0 *can be solved in* $O(n \log n)$ *time.*

Proof. First, observe that if $\sum_i 2\rho_i < 1$, the maximum lifetime is 0. Otherwise, initialize $S = \emptyset$. As long as $\sum_{i \in S} 2\rho_i < 1$, add $i = \text{argmin}_{i \notin S} \rho_i$ to S. Finally, assign $r_i = \rho_i$, for $i \in S$, and $r_i = 0$, for $i \notin S$. The correctness of this algorithm is straightforward. The running time of the algorithm is $O(n \log n)$, since we need to sort the sensors by their radii. $\square$

2.4.2 Infinite Friction

Theorem 2.4.2. $\textsc{MaxFix}_\infty$ *can be solved in* $O(n \log n)$ *time.*

Proof. Given a solution (x, r), observe that if $r_i = \rho_i$, for some sensor i, then we may assume without loss of generality that $r_j = \rho_j$, for every j such that $\rho_j \leq \rho_i$. This motivates the following algorithm. Initialize $S = \emptyset$. As long as $[0, 1]$ is not covered, add $i = \operatorname{argmin}_{i \notin S} \rho_i$ to S. Finally, assign $r_i = \rho_i$, for all $i \in S$, and $r_i = 0$, for all $i \notin S$. The value of the solution is $T\rho_{i^*}^{\alpha}$, where i^* was the last sensors to join S. The correctness of this algorithm follows from the above observation.

Sorting the sensors by their radii takes $O(n \log n)$ time. We maintain a list of uncovered segments that is initialized by the segment $[0, 1]$. Such a list can be updated in $O(n)$, since it contains at most n segments. Consequently, the total running time is $O(n^2)$. We can obtain a more efficient implementation by storing the segments in a balanced search tree. Since there are $O(n)$ insertions and deletions, the total running time is $O(n \log n)$. $\square$

2.4.3 Non-zero Finite Friction

As mentioned earlier, Czyzowicz et al. [18] presented an polynomial time algorithm for MaxFix with uniform radii. Their result is based on showing that there exists a non-swapping optimal solution for the special case of uniform radii. We show that SumFix is NP-hard, even if $x = (\frac{1}{2})^n$, using a reduction from Partition. This result also implies that it is NP-hard to find an optimal ordering of a MaxFix instance.

Theorem 2.4.3. MaxFix *is NP-hard even if $x = (\frac{1}{2})^n$, for every $a \in (0, \infty)$ and $\alpha \geq 1$.*

Proof. Given a Partition instance $(s_1, \ldots, s_n)$, we construct a MaxFix instance with $n+1$ sensors as follows. $x_i = \frac{1}{2}$, for every i, $\rho_i = \frac{s_i}{4\sum_j s_j}$, for $i \leq n$, and $\rho_{n+1} = \frac{1}{4}$. Also, let $t = a4^\alpha$. The MaxFix instance can be constructed in linear time. We show that $(s_1, \ldots, s_n) \in$ Partition if and only if there is a solution (y, r) such that $\max_i E_i(y, r) = a$.

Suppose that $(s_1, \ldots, s_n) \in$ Partition, and let $I \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in I} s_i = \sum_{i \notin I} s_i$. Set $r_i = \rho_i$, for every i. Use sensor $n+1$ to cover the interval $[\frac{1}{4}, \frac{3}{4}]$, the sensors that correspond to I to cover the interval $[0, \frac{1}{4}]$, and the rest of the sensors to cover the interval $[\frac{3}{4}, 1]$. This is possible, since $\sum_{i \in I} 2\rho_i = \sum_{i \in \{1,\ldots,n\} \setminus I} 2\rho_i = \frac{1}{4}$. A sensor i, where $i \leq n$, needs less than $\frac{a}{2}$

energy to move, and at most $a4^\alpha \cdot \frac{1}{8}^\alpha = \frac{a}{2^\alpha} \le \frac{a}{2}$ for coverage, therefore it can stay alive for $a4^\alpha$ time. Sensor $n+1$ stays put and requires $a4^\alpha \cdot \frac{1}{4^\alpha} = a$ energy. Hence, maintaining cover for $a4^\alpha$ time can be obtained with maximum energy a.

Now suppose that there exists a solution (y, r) such that $\max_i E_i(y, r) = a$. Notice that $\sum_i 2\rho_i = 1$, and thus it must be that $r_i = \rho_i$, for every i. Since sensors $n+1$ requires all its energy for covering, it must be that $y_{n+1} = x_{n+1} = \frac{1}{2}$. It follows that the interval $[0, \frac{1}{4}]$ is covered by a set of sensors I that satisfy $\sum_{i \in I} 2\rho_i = \frac{1}{4}$. Hence $\sum_{i \in I} s_i = \frac{1}{2}\sum_i s_i$, which means that $(s_1, \ldots, s_n) \in$ PARTITION. $\square$

We use a similar approach to describe a reduction from 3-PARTITION. This implies strong NP-hardness.

Theorem 2.4.4. MAXFIX *is strongly NP-hard, for every* $a \in (0, \infty)$ *and* $\alpha \ge 1$.

Proof. Given a 3-PARTITION instance $(s_1, \ldots, s_n)$, where $n = 3m$, $\sum_i s_i = mQ$, and $s_i \in (\frac{Q}{4}, \frac{Q}{2})$, for every i, we construct the following MAXFIX instance with $n + m - 1$ sensors as follows. $x_i = \frac{1}{2}$ and $\rho_i = \frac{s_i}{2(2m-1)Q}$, for every $i \le n$, and $x_i = \frac{2(i-n)-\frac{1}{2}}{2m-1}$ and $\rho_i = \frac{1}{2(2m-1)}$, for $i > n$. Also, let $T = a2^\alpha(2m-1)^\alpha$. The instance can be constructed in linear time. We show that $(s_1, \ldots, s_n) \in$ 3-PARTITION if and only if there exists a solution (y, r) such that $\max_i E_i(y, r) = a$.

Suppose that $(s_1, \ldots, s_n) \in$ 3-PARTITION, and let $I_1, \ldots, I_m \subseteq \{1, \ldots, n\}$, such that $|I_k| = 3$ and $\sum_{i \in I_k} s_i = Q$, for every k. Set $r_i = \rho_i$, for every i. Use sensor $n+k$, for $k \in \{1, \ldots, m-1\}$ to cover the interval $[\frac{2k-1}{2m-1}, \frac{2k}{2m-1}]$ by assigning $y_{n+k} = x_{n+k}$. Also, use the sensors in I_k, for $k \in \{1, \ldots, m\}$ to cover the interval $[\frac{2k-2}{2m-1}, \frac{2k-1}{2m-1}]$. This is possible, since $\sum_{i \in I_k} 2\rho_i = \sum_{i \in I_k} \frac{s_i}{(2m-1)Q} = \frac{1}{2m-1}$. Now, sensor i, where $i \leq n$, needs less than $\frac{a}{2}$ energy to move, and

$$tr_i^\alpha < t(\tfrac{1}{4(2m-1)})^\alpha = a2^\alpha(2m-1)^\alpha \cdot \tfrac{1}{4^\alpha(2m-1)^\alpha} = \tfrac{a}{2^\alpha} \leq \tfrac{a}{2}$$

for coverage, therefore it can stay alive for t time with energy a. Sensor i, where $i > n$ stays put and consumes $a2^\alpha(2m-1)^\alpha \cdot \frac{1}{2^\alpha(2m-1)^\alpha} = a$ energy. Hence, maintaining coverage for T time can be obtained with maximum energy a.

Now suppose that there exists a solution (y, r) such that $\max_i E_i(y, r) = a$. Notice that $\sum_i 2\rho_i = 1$, and thus it must be that $r_i = \rho_i$, for every i. Since sensor i, for $i > n$, requires all its energy for covering, it must be that $y_i = x_i$, for $i > n$. It follows that the interval $[\frac{2k-2}{2m-1}, \frac{2k-1}{2m-1}]$, for $k \in \{1, \ldots, m\}$, is covered by a set of sensors I_k that satisfy $\sum_{i \in I_k} 2\rho_i \geq \frac{1}{2m-1}$. Hence $\sum_{i \in I_k} s_i \geq Q$, which means that $(s_1, \ldots, s_n) \in$ 3-PARTITION. □

2.5 Minimum Energy Sum with Fixed Radii

In this section we consider SumFix. Li et al. [36] solved SumFix_∞ using an elegant reduction to the shortest path problem. We show that SumFix_0 is NP-hard, if $\alpha = 1$, but admits an FPTAS, for any α. Czyzowicz et al. [19] showed that it is NP-hard to approximate the special case of SumFix_1 where $t = 0$ to within any constant c. We show their approach can be used for a stronger result, namely that it is NP-hard to approximate SumFix, for any $a \in (0, \infty)$, to within a factor of $O(n^c)$, for any constant c.

We note that the optimal solution and energy invested in movement may change dramatically with the increase of the required lifetime t. Assume $a = 1$ and consider an instance in which there are $n - 1$ sensors, where $x_i = \frac{i}{n-1}$ and $\rho_i = \frac{1}{2(n-1)}$, for $i \leq n - 1$, and $x_n = \frac{1}{2}$ and $\rho_n = \frac{1}{2}$. If $t = 0$, we can use sensor n to cover the barrier without moving any sensor. However, if t is large enough, it is better to deploy sensor i at $y_i = \frac{2i-1}{2(n-1)}$, for $i \leq n - 1$, and cover the barrier without the help of sensor n. In this case the optimal value is $\frac{1}{2} + \frac{t}{2^\alpha (n-1)^\alpha}$.

2.5.1 Zero Friction

We show that SumFix_0 is NP-hard, for $\alpha = 1$, and that it has an FPTAS, for any α. We start with the hardness result.

Theorem 2.5.1. SUMFIX$_0$ *is NP-hard, for* $\alpha = 1$.

Proof. We present a reduction from PARTITION. Given a PARTITION instance $(s_1, \ldots, s_n)$, let $S = \sum_i s_i$. We construct a SUMFIX$_0$ instance with n sensors as follows. First, $t = 1$. Also, $\rho_i = \frac{s_i}{S}$, for every i. Notice that $\sum_i \rho_i = 1$. The instance can be constructed in linear time. We prove that $(s_1, \ldots, s_n) \in$ PARTITION if and only if there is solution (y, r) such that $\sum_i E_i(y, r) \leq \frac{1}{2}$.

Suppose that $(s_1, \ldots, s_n) \in$ PARTITION. It follows that there exists an index set I such that $\sum_{i \in I} s_i = \frac{1}{2}S$. Let $r_i = \rho_i$, if $i \in I$, and $r_i = 0$ otherwise. Also, let $y_i = \sum_{j=1}^{i-1} 2r_j + r_i$, for every i. Notice that $\sum_i r_i = \sum_i \frac{s_i}{S} = \frac{1}{2}$. Hence, (y, r) covers $[0, 1]$ and $\sum_i E_i(y, r) = \sum_i tr_i = \frac{1}{2}$ as required.

Now suppose that there exists a solution (y, r) that satisfies $\sum_i E_i(y, r) \leq \frac{1}{2}$. Since r covers $[0, 1]$ we have that $\sum_i r_i \geq \frac{1}{2}$. On the other hand, $\sum_i E_i(y, r) = t \sum_i r_i \leq \frac{1}{2}$, which means that $\sum_i r_i = \frac{1}{2}$. Now let $I = \{i : r_i = \rho_i\}$, and we have $\sum_{i \in I} s_i = S \sum_{i \in I} r_i = S \sum_i r_i = \frac{1}{2}S$. □

The FPTAS for SUMFIX$_0$ is implied by a reduction to MINIMUM KNAPSACK.

Theorem 2.5.2. SUMFIX$_0$ *has an FPTAS.*

Proof. We show a reduction from SUMFIX$_0$ to MINIMUM KNAPSACK. Given a SumFix$_0$ instance (x, ρ) and t, we construct a Minimum Knapsack instance as follows. The covering requirement is 1. Also, there are n items, where the coverage of item i is $2\rho_i$, and its cost is $t\rho_i^\alpha$. Any solution (y, r) can be mapped to the set $I = \{i : r_i = \rho_i\}$ of items that has the same cost. Any set of items I can be mapped to the solution (y, r) with the same cost such that $r_i = \rho_i$, if $i \in I$, and $r_i = 0$ otherwise, and $y_i = \sum_{j=1}^{i-1} 2r_j + r_i$, for every i. Since the Minimum Knapsack problem has an FPTAS [31], SumFix$_0$ also has an FPTAS, for any $\alpha \geq 1$. □

2.5.2 Non-Zero Finite Friction and Uniform Radii

We present a polynomial time algorithm that computes solutions within an additive factor ε, for any constant $\varepsilon > 0$, for uniform SumFix instances. Our algorithm is also based on the non-swapping property and placing the sensors on grid point. However, as opposed to the variable case, we cannot change the radii, only the locations. This is problematic when there is very little excess coverage. We cope with this issue by considering two solution types, small excess and large excess.

We start the section with proving that there exists a non-swapping optimal solution. The proof of the next lemma is identical to the proof of Lemma 2.2.2. One only needs to notice that a switch can be made since

$\rho_i = \rho_j$, for every $i \neq j$.

Lemma 2.5.3. *Any uniform* SumFix *instance has a non-swapping optimal solution.*

Given SumFix instance with uniform radii, denote $\rho = R^n$, i.e., all sensors have radius R. Given a feasible solution (y, r), let $X(r)$ denote the *excess* coverage of the solution, namely $X(r) = \sum_i 2r_i - 1$. Clearly $X(r) \geq 0$.

Let $\varepsilon > 0$. We first show that there is a polynomial time algorithm that computes a solution within an additive factor ε for any uniform SumFix instance that has an optimal solution (y, r) such that $X(r) > \frac{\varepsilon}{an}$.

Define $m = an^2/\varepsilon$. We consider solutions in which the active sensors must be located on grid points $\mathcal{G} = \{\frac{j}{m} : j \in \{0, \ldots, m\}\}$. We also introduce a slightly different notion of "non-swapping". A solution (y, r) (or a deployment y) is called *weakly non swapping* if: (i) $x_i < x_j$ implies $y_i \leq y_j$, if both i and j are active, and (ii) $y_i = x_i$, if i is inactive.

We prove that we only lose a small additive factor by focusing on weakly non-swapping deployments that use grid points for active sensors.

Lemma 2.5.4. *Let $\varepsilon > 0$, and let (x, ρ, t) be a uniform* SumFix *instance that has a non-swapping optimal solution (y, r) with $X(r) > \frac{\varepsilon}{an}$. There is a weakly non-swapping deployment y' such that (i) (y', r) is feasible, (ii) $y'_i \in \mathcal{G}$,* *if i is active, and (iii) $\sum_i E_i(y', r) \le \sum_i E_i(y, r) + \varepsilon$.*

Proof. First assume that all sensors are active (for ease of notation). Going from $i = 1$ to n, let y'_i be the rightmost grid point such that $y'_i \le y_i + \frac{n}{m}$ and $y'_i \le y'_{i-1} + 2R$, for $i > 1$, or $y'_1 \le R$.

We claim that (y', r) is feasible. Assume that it is not, namely that $y'_n + R < 1$. We prove by induction (from n to 1) that $y'_i < y_i + \frac{n-i}{m}$. In the base case, we have that $y'_n < y_n$. For the inductive step, note that $y'_{i+1} < y_{i+1} + \frac{n-(i+1)}{m}$ due to the inductive hypothesis. It follows that $y'_{i+1} = y'_i + \frac{\lfloor 2Rm \rfloor}{m}$. Hence,

$$y'_i = y'_{i+1} - \frac{\lfloor 2Rm \rfloor}{m} < y_{i+1} + \frac{n-(i+1)}{m} - 2R + \frac{1}{m} \le y_i + \frac{n-i}{m} \ .$$

$y'_1 \le y_1 + \frac{n-1}{m}$ implies that $y'_1 = \frac{\lfloor Rm \rfloor}{m}$. It follows that

$$y'_n + R = \frac{\lfloor Rm \rfloor}{m} + (n-1)\frac{\lfloor 2Rm \rfloor}{m} + R \ge 2nR - \frac{n}{m} = 2nR - \frac{\varepsilon}{an} > 2nR - X(r) = 1 \ ,$$

in contradiction to $y'_n + R < 1$.

To bound the cost of the solution, we prove that $y'_i \ge y_i - \frac{i}{m}$ by induction on i. For the base case, observe that $y'_1 \ge \frac{\lfloor my_1 \rfloor}{m} > y_1 - \frac{1}{m}$. For the inductive step, we have two options. If $y'_i \ge y_i$, then we are done. Otherwise, $y'_i < y_i$, and in this case

$$y'_i = y'_{i-1} + \frac{\lfloor 2Rm \rfloor}{m} \ge y_{i-1} - \frac{i-1}{m} + 2R - \frac{1}{m} \ge y_i - \frac{i}{m} \ .$$

It follows that $|y'_i - y_i| \leq \frac{n}{m}$. Hence, $\sum_i E_i(y',r) \leq \sum_i E_i(y,r) + an\frac{n}{m} = \sum_i E_i(y,r) + \varepsilon$.

Finally, we deploy inactive sensors in their initial positions. This only decreases the energy consumption. Also, observe that $y_i \in \mathcal{G}$, for any active sensors, and that y' is weakly non-swapping by construction. $\square$

In light of Lemma 2.5.4, we describe a directed acyclic graph G with a source s and a destination d, such that a path from s to d corresponds to a solution for the SumFix instance. The vertex set of G contains a vertex for every pair of sensor and grid point and two additional vertices, i.e., $V(G) = \{s,d\} \cup \{(i,j) : i \in \{1,\ldots,n\}, j \in \{0,\ldots,m\}\}$. An arc connects two vertices (i,j) and (i',j'), if $i < i'$ and $\frac{j'}{m} \leq \frac{j}{m} + 2R$. An arc connects s and (i,j) if $\frac{j}{m} \leq R$, and an arc connects (i,j) and d if $\frac{j}{m} \geq 1 - R$. The length of the arcs leaving a vertex (i,j) is $a|x_i - \frac{j}{m}| + tR^\alpha$, and the length of the arcs leaving s is zero. There is a one to one mapping between paths from s to d in G to grid solutions of a SumFix instance. If follows that, given a uniform SumFix instance (x,ρ,t) such that $X(r) > \frac{\varepsilon}{an}$, we can compute a solution within additive factor ε by constructing the above graph and finding a shortest path from s to d. Notice that the running time of this algorithm is polynomial since $m = O(n^2)$.

It remains to consider instances with an optimal solution (y, r) such that $X(r) \leq \frac{\varepsilon}{an}$. We show that, in this case, we do not lose much by assuming that active sensors are located at the predetermined positions: $\mathcal{P} = \{R(2\ell - 1) : \ell \in \{1, \dots, \chi\}\}$, where $\chi = \lceil \frac{1}{2R} \rceil$.

Lemma 2.5.5. *Let $\varepsilon > 0$, and let (x, ρ, t) be a uniform* SumFix *instance that has a non-swapping optimal solution (y, r) with $X(r) \leq \frac{\varepsilon}{an}$. There is a weakly non-swapping feasible solution (y', r') such that (i) there are χ active sensors deployed at $\mathcal{P}$, and (ii) $\sum_i E_i(y', r') \leq \sum_i E_i(y, r) + \varepsilon$.*

Proof. Let $i_1, \dots, i_k$ be the active sensors. Let $\delta_1 = 0 - (y_{i_1} - R)$ be the excess coverage to the left of 0; let $\delta_{k+1} = (y_{i_k} + R) - 1$ be the excess coverage to the right of 1; and let $\delta_j = (y_{i_{j-1}} + R) - (y_{i_j} - R)$, for $j \in \{2, \dots, k\}$, be the excess coverage due to cover overlaps. We have that $X(r) = \sum_{j=1}^{k+1} \delta_j$.

To create the deployment y' we first place all inactive sensors in their initial positions. Then, we go from $j = 1$ to k, and move active sensors $i_q \geq i_j$ by δ_j rightwards. That is, $y'_{i_q} = y_{i_q} + \sum_{j \leq q} \delta_j$. Inactive sensors simply remain at their initial positions. Next, we deactivated any active sensor that does not cover $[0, 1]$, and move these sensors back to the their original location. Observe that y' is weakly non-swapping by construction.

We show that by induction on j that $y'_{i_j} = R(2j - 1)$, for every j. For the base case, we have that $y'_{i_1} = y_{i_1} + \delta_1 = y_{i_1} + 0 - (y_{i_1} - R) = R$. For the inductive step,

$$y'_{i_j} = y_{i_j} + \sum_{\ell=1}^{j} \delta_j = y_{i_{j-1}} + 2R - \delta_j + \sum_{\ell=1}^{j} \delta_j = R(2(j-1)-1) + 2R = R(2j-1) \ .$$

It follows that there are χ active sensors located at $\mathcal{P}$, and that (y', r') is feasible. Also, $|y'_i - y_i| \leq X(r)$, for every i, and therefore the additional movement cost is bounded by $anX(r) \leq \varepsilon$. Hence $\sum_i E_i(y', r) \leq \sum_i E_i(y, r) + \varepsilon$. □

Assuming that exactly χ sensors are located at the predetermined positions, we construct a directed acyclic graph H as follows. The vertex set is $V(H) = (i, \ell) : i \in \{1, \ldots, n\}, \ell \in \{1, \ldots, \chi\}] \cup (0,0), (n+1, \chi+1)$, where $(0,0)$ is the source and $(n+1, \chi+1)$ is the destination. The arc set is $E(H) = ((i, \ell), (i', \ell+1)) : i < i'$. The length of arcs leaving (i, ℓ), where $i > 0$, is $a|x_i - R(2\ell - 1)|$, while arcs leaving $(0,0)$ are of length zero. (There is no need to consider coverage energy, since we use exactly χ sensors.) As before, there is a one to one mapping between paths from $(0,0)$ to (n, χ) in G to solutions for the SumFix instance induced by Lemma 2.5.5. Hence a shortest path from $(0,0)$ to $(n+1, \chi+1)$ corresponds to an optimal solution on the predetermined locations.

It follows that an approximate solution can be found by running both algorithms, and taking the better solution. This leads to the following result.

Theorem 2.5.6. *There exists a polynomial time algorithm that computes solutions within additive factor ε, for any constant $\varepsilon > 0$, for* SumFix *with uniform radii.*

We finish the section by observing that, if $a = 0$, an optimal solution uses χ active sensors. Also, moving to the predetermined locations cost nothing.

Theorem 2.5.7. *There exists a polynomial time algorithm for* SumFix_0 *with uniform radii.*

2.5.3 Non-zero Finite Friction and General Radii

We show that no-swapping does not hold in general for non-uniform instances.

Lemma 2.5.8. *There are* SumFix *instances in which no-swapping does not hold, for any $a > 0$ and $\alpha \geq 1$. Moreover, the ratio between the value of best no swapping solution and the optimum is $\Omega(n)$.*

Proof. Consider the following SumFix instance: two sensors are located at 0 both with radius $\frac{1}{4}$, and $n - 2$ sensors at $\frac{1}{4}$ all with radius $\frac{1}{4(n-2)}$. Also, let $t = a$. First, assume no swapping. Let p be the maximum point that is covered by one of the first two sensors. If $p \geq \frac{3}{4}$, it follows that sensor 2, without loss of generality, was deployed at $y_2 \geq \frac{1}{2}$. In this case $y_i \geq y_2 \geq \frac{1}{2}$, for $i \geq 3$. Hence, the movement energy is at least $(n-2)\frac{a}{4}$. Otherwise, if $p < \frac{3}{4}$, we have that $[\frac{3}{4}, 1]$ is covered by at least $\frac{n-2}{2}$ sensors. Hence, the movement energy is at least $\frac{n-2}{2}\frac{a}{2} = (n-2)\frac{a}{4}$. It follows that at least $(n-2)\frac{a}{4}$ energy must be consumed if swapping is disallowed. If swapping is allowed, we may cover the barrier with sensors 1 and 2. We deploy sensors 1 and 2 at $\frac{1}{4}$ and $\frac{3}{4}$, respectively, and assign $r_1 = r_2 = \frac{1}{4}$, and $r_i = 0$, for every $i > 2$. The movement energy in this case is exactly a, and the coverage energy is $2t\frac{1}{4^\alpha} = \frac{2a}{4^\alpha}$. Hence, the total energy consumption is $a + \frac{2a}{4^\alpha} \leq \frac{3a}{2}$. If $n > 8$, we have that $(n-2)\frac{a}{4} > \frac{3a}{2}$. □

Czyzowicz et al. [19] proved that the special case of SumFix in which $t = 0$ cannot be approximated within any constant. We show their approach can be used for a stronger result, namely that it is NP-hard to approximate SumFix, for any $a \in (0, \infty)$, to within a factor of $O(n^c)$, for any constant c. Our reduction is almost identical to the reduction from [19]. The proof is given in full detail for completeness.

Theorem 2.5.9. SumFix *cannot be approximated to within a factor of* $O(n^c)$*, for any constant* c*, for every* $a \in (0, \infty)$ *and* $\alpha \geq 1$*, unless P=NP.*

Proof. We show that it is NP-hard to approximate SumFix within a factor of $Bn^c/8$, for any constants $c \in \mathbb{N}$ and $B \in \mathbb{N}$.

Let $\ell = 8Bm(Q+1)n^{c+1}$. Given a 3-PARTITION instance $(s_1, \ldots, s_n)$, where $n = 3m$ and $\sum_i s_i = mQ$, we construct the following SUMFIX instance with $n + m\ell$ sensors as follows. First, $x_i = 0$ and $\rho_i = \frac{s_i}{2m(Q+1)}$, for every $i \leq n$. We also add a $(\frac{j(Q+1)}{m(Q+1)}, \frac{1}{m(Q+1)}, \ell)$-block, for every $j \in \{0, \ldots, m-1\}$, where a (z, Δ, ℓ)-block is a set of ℓ sensors whose positions are at $\left\{z + \frac{\Delta}{\ell}(i - \frac{1}{2}) : i \in \{1, \ldots, \ell\}\right\}$ and their uniform radius is $\frac{\Delta}{2\ell}$. Also, let $T = 0$. The running time is polynomial, since Q and ℓ are polynomial in n. We show that (i) if $(s_1, \ldots, s_n) \in$ 3-PARTITION, then exists a solution (y, r) such that $\sum_i E_i(y, r) \leq na$, and (ii) if $(s_1, \ldots, s_n) \notin$ 3-PARTITION, then $\sum_i E_i(y, r) > aBn^{c+1}/8$, for any solution (y, r). It follows that it is NP-hard to approximate SUMFIX within a factor of $Bn^c/8$.

Now suppose that $(s_1, \ldots, s_n) \in$ 3-PARTITION, and let $I_1, \ldots, I_m \subseteq \{1, \ldots, n\}$, such that $|I_k| = 3$ and $\sum_{i \in I_k} s_i = Q$, for every k. Set $r_i = \rho_i$, for every i. Use the sensors in I_k to cover the segment $[\frac{k(Q+1)-Q}{m(Q+1)}, \frac{k(Q+1)}{m(Q+1)}]$. This is possible, since $\sum_{i \in I_k} 2\rho_i = \sum_{i \in I_k} \frac{s_i}{m(Q+1)} = \frac{Q}{m(Q+1)}$. Also, use the sensors in the $(\frac{k(Q+1)}{m(Q+1)}, \frac{1}{m(Q+1)}, \ell)$-block to cover $[\frac{k(Q+1)}{m(Q+1)}, \frac{k(Q+1)+1}{m(Q+1)}]$, for every $k \in \{0, \ldots, m-1\}$. Now, sensor i, where $i \leq n$, needs less than a energy to move. Hence, the total energy consumption is less than na.

Next, suppose that $(s_1, \ldots, s_n) \notin$ 3-PARTITION, and let (y, r) be a feasible solution such that $\sum_i E_i(y, r) \leq aBn^{c+1}/8$. Notice that $\sum_i 2\rho_i = 1$, and thus it must be that $r_i = \rho_i$, for every i. It follows that there are no coverage overlaps. Consider the $(\frac{k(Q+1)}{m(Q+1)}, \frac{1}{m(Q+1)}, \ell)$-block, for some k. We would like to bound the number of sensors from this block that deploy outside the segment $[\frac{k(Q+1)}{m(Q+1)} - \frac{1}{8m(Q+1)}, \frac{k(Q+1)+1}{m(Q+1)} + \frac{1}{8m(Q+1)}]$. Each such sensor needs more than $a\frac{1}{8m(Q+1)}$ energy, so their number is at most $Bn^{c+1}/8 \cdot 8m(Q+1) = Bn^{c+1}m(Q+1)$. Hence at least $7\ell/8$ sensors from the kth block remain in the above segment. This means that all large radii sensors must be located in $[\frac{k(Q+1)-Q}{m(Q+1)} - \frac{1}{4m(Q+1)}, \frac{k(Q+1)}{m(Q+1)} + \frac{1}{4m(Q+1)}]$, for some k, and therefore the blocks still act as static delimiters. It follows that $(s_1, \ldots, s_n) \in$ 3-Partition. A contradiction. □

2.6 Open Problems and Future Directions

An obvious open questions is to come up with an approximation algorithm or a lower bound for MaxFix. Another research direction is to consider a model in which sensors are allowed to move and to change their covering radii at any given time. In another natural extension, sensors are located on a barrier and are required to cover a region (e.g., sensors on a coastline covering the sea). In the dual model, sensors could be located anywhere in the plane and are asked to cover a boundary (e.g., sensors in the sea covering the coastline). In an even more general model, a sensor network is required to cover a region in the plane and the initial locations of the sensors are anywhere in the plane.

Chapter 3

Maximizing Barrier Coverage Lifetime

This chapter is divided in the following way. Section 3.1 gives the preliminaries. Section 3.2 deals with the no friction and infinite friction case. Section 3.3 deals with arbitrary non-extreme friction when order is predefined. Section 3.4 resolves the case where sensors are initially located at the edges of the barrier. Section 3.5 discusses the uniform battery (and radii for BCFR) case. In Section 3.6, hardness results are given. Finally, Section 3.7 concludes the chapter and discusses open problems.

3.1 Preliminaries

In this section we formally define the problems and introduce the notation that will be used throughout the chapter.

Model. We consider a setting in which n mobile sensors with finite batteries are located on a barrier represented by the interval $[0, 1]$. The initial position and battery power of sensor i is denoted by x_i and b_i, respectively. We denote $x = (x_1, \ldots, x_n)$ and $b = (b_1, \ldots, b_n)$. The sensors are used to cover the barrier, and they can achieve this goal by moving and sensing. In our model the sensors first move, and afterwards each sensor covers an interval that is determined by its sensing radius. In motion, energy is consumed in proportion to the distance traveled, namely a sensor consumes $a \cdot d$ units of energy by traveling a distance d, where a is a constant. A sensor i consumes r_i^α energy per time unit for sensing, where r_i is the sensor's radius and $\alpha \geq 1$ is a constant. More formally, the system works in two phases. In the *deployment phase* sensors move from the initial positions x to new positions y. This phase is said to occur at time 0. In this phase, sensor i consumes $a|y_i - x_i|$ energy. Notice that sensor i may be moved to y_i only if $a|y_i - x_i| \leq b_i$. In the *covering phase* sensor i is assigned a sensing radius r_i and covers the interval $[y_i - r_i, y_i + r_i]$. (An example is given in Figure 2.1.) A pair (y, r), where y is a deployment vector and r is a sensing radii vector, is called *feasible* if (i) $a|y_i - x_i| \leq b_i$, for every sensor i, and (ii) $[0, 1] \subseteq \sum_i [y_i - r_i, y_i + r_i]$. Namely, (y, r) is feasible, if the sensors have enough power to reach y and each point in $[0, 1]$ is covered by some sensor.

Given a feasible pair (y, r), the *lifetime* of a sensor i, denoted $L_i(y, r)$, is the time that transpires until its battery is depleted. If $r_i > 0$, $L_i(y, r) = \frac{b_i - a|y_i - x_i|}{r_i^\alpha}$, and if $r_i = 0$, we define $L_i(y, r) = \infty$. Given initial locations x and battery powers b, the *barrier coverage lifetime* of a feasible pair (y, r), where y is a deployment vector and r is a sensing radii vector is defined as $L(y, r) = \min_i L_i(y, r)$. We say that a t is *achievable* if there exists a feasible pair such that $L_i(y, r) = t$.

Problems. We consider two problems which are distinguished by whether the radii are given as part of the input. In the Barrier Coverage with Variable Radii problem (BCVR) we are given initial locations x and battery powers b, and the goal is to find a feasible pair (y, r) of locations and radii that maximizes $L(y, r)$. In the Barrier Coverage with Fixed Radii problem (BCFR) we are also given a radii vector ρ, and the goal is to find a feasible pair (y, r), such that $r_i \in \{0, \rho_i\}$ for every i, that maximizes $L(y, r)$. Notice that a necessary condition for achieving non-zero lifetime is $\sum_i 2\rho_i \geq 1$.

Given a total order $\prec$ on the sensors, we consider the *constrained* variants of BCVR and BCFR, in which the deployment y must satisfy the following: $i \prec j$ if and only if $y_i \leq y_j$. That is, we are asked to maximize barrier coverage lifetime subject to the condition that the sensors are ordered by $\prec$ (this includes sensors that do not participate in the cover). Without loss of generality, we assume that the sensors are numbered according to the total order.

3.2 Extreme Movement Costs

In this section we consider the two extreme cases, the static case ($a = \infty$) and the fully dynamic case ($a = 0$).

3.2.1 The Static Case

In the static case the initial deployment is the final deployment, i.e., $y = x$, and therefore a feasible solution is a radii assignment r, such that $[0, 1] \subseteq \cup_i [x_i - r_i, x_i + r_i]$.

We describe a simple algorithm for static BCFR. First, if $[0, 1] \not\subseteq \cup_i [x_i - \rho_i, x_i + \rho_i]$, then the maximum lifetime is 0. Otherwise, compute $t_i = b_i/\rho_i^\alpha$ for every i, and let $S = \emptyset$. Then, as long as S does not cover the barrier, add $i = \text{argmax}_{i \notin S}\, t_i$ to S. Finally, assign $r_i = \rho_i$, for $i \in S$, and $r_i = 0$, for $i \notin S$. The correctness of this algorithm is straightforward.

Bar-Noy *et al.* [7] presented a polynomial time algorithm for static BCVR with $\alpha = 1$. This algorithm readily extends to static BCVR with $\alpha > 1$.

We refer the reader to [7] for the details.

3.2.2 Fully Dynamic Case

In the fully dynamic case movement is for free, and therefore any radii vector r, such that $\sum_i 2r_i \geq 1$, has a deployment vector y such that (y, r) is a feasible pair. (e.g., $y_i = \sum_{j=1}^{i-1} 2r_j + r_i$, for every i.)

We describe a simple algorithm for fully dynamic BCFR. First, if $\sum_i 2\rho_i < 1$, the maximum lifetime is 0. Otherwise, compute $t_i = b_i/\rho_i^\alpha$ for every i, and let $S = \emptyset$. Then, as long as $\sum_{i \in S} 2\rho_i < 1$, add $i = \text{argmax}_{i \notin S}\, t_i$ to S. Finally, assign $r_i = \rho_i$, for $i \in S$, and $r_i = 0$, for $i \notin S$. The correctness of this algorithm is straightforward.

We now consider fully dynamic BCVR. Given a feasible radii vector r and a corresponding deployment vector y, the lifetime of sensors i is simply $L_i(y, r) = b_i/r_i^\alpha$, and the lifetime of the system is $L(y, r) = \min_i L_i(y, r)$.

Theorem 3.2.1. *Let $a = 0$. Given a* BCVR *instance, the radii assignment $r_i = \frac{\sqrt[\alpha]{b_i}}{2\sum_j \sqrt[\alpha]{b_j}}$, for every i, is optimal.*

Proof. First, observe that $2\sum_i r_i = \sum_i \frac{\sqrt[\alpha]{b_i}}{\sum_j \sqrt[\alpha]{b_j}} = 1$, which means that r is feasible. Furthermore,

$$L_i(r) = b_i/r_i^\alpha = b_i \cdot \left(\frac{2\sum_j \sqrt[\alpha]{b_j}}{\sqrt[\alpha]{b_i}} \right)^\alpha = \left(2\sum_j \sqrt[\alpha]{b_j} \right)^\alpha ,$$

for every i. Hence, $L(r) = (2\sum_j \sqrt[\alpha]{b_j})^\alpha$.

We show that $L(r) < L(r')$, for any radii assignment $r' \neq r$. Since r' is feasible, we have that $2\sum_i r'_i \geq 1$. It follows that there exists i such that $r'_i > r_i$. Hence, $L(r') \geq L_i(r') > L_i(r) = L(r)$. □

3.3 Constrained Problems and Parametric Search

In this section we present polynomial time algorithm that, given $t > 0$, decides whether t is achievable for constrained BCFR. In addition, we give a similar algorithm that, given $t > 0$ and any accuracy parameter $\epsilon > 0$, decides whether $t - \varepsilon$ is achievable for constrained BCVR. If the answer is in the affirmative, a corresponding solution is computed by both algorithms. We use these algorithms to design parametric search algorithms for both problems.

We use the following definitions for both BCFR and BCVR. Given an order requirement $\prec$, we define:

$$l(i) \stackrel{\text{def}}{=} \max\left\{\max_{j\leq i}\left\{x_j - b_j/a\right\}, 0\right\} \quad u(i) \stackrel{\text{def}}{=} \min\left\{\min_{j\geq i}\left\{x_j + b_j/a\right\}, 1\right\}$$

$l(i)$ and $u(i)$ are the leftmost and rightmost points reachable by i.

Observation 3.3.1. *Let (y, r) be a feasible solution that satisfies an order requirement $\prec$. Then $l(i) \leq u(i)$ and $y_i \in [l(i), u(i)]$, for every i.*

Proof. If there exists i such that $u(i) < l(i)$, then there are two sensors j and k, such that where $k < j$ and $x_j + b_j/a < x_k - b_k/a$. Hence, no deployment that satisfies the total order exists. □

3.3.1 Fixed Radii

We start with an algorithm that solves the constrained BCFR decision problem.

Given a BCFR instance and a lifetime t, we define

$$s(i) \stackrel{\text{def}}{=} \max\{x_i - (b_i - t\rho_i^\alpha)/a, l(i)\} \quad e(i) \stackrel{\text{def}}{=} \min\{x_i + (b_i - t\rho_i^\alpha)/a, u(i)\}$$

If $t\rho_i^\alpha \le b_i$, then $s(i) \le e(i)$. Moreover $s(i)$ and $e(i)$ are the leftmost and rightmost points that are reachable by i, if i participates in the cover for t time. ($l(i)$ and $u(i)$ can be replaced by $l(i-1)$ and $u(i-1)$ in the above definitions.)

Observation 3.3.2. *Let (y, r) be a feasible pair with lifetime t that satisfies an order $\prec$. For every i, if $r_i = \rho_i$, it must be that $t\rho_i^\alpha \le b_i$ and $y_i \in [s(i), e(i)]$.*

Algorithm **Fixed** is our decision algorithm for constrained BCFR. It first computes l, u, s, and e. If there is a sensor i such that $l(i) > u(i)$, it outputs NO. Otherwise it deploys the sensors one by one according to $\prec$.

Algorithm 1 : Fixed (x, b, ρ, t)

Compute l, u, s, and e
if there exists i such that $u(i) < l(i)$ **then**
return NO
$z \leftarrow 0$
for $i = 1 \rightarrow n$ **do**
    if $t\rho_i^\alpha > b_i$ or $z \notin [s(i) - \rho_i, e(i) + \rho_i)$ **then**
        $y_i \leftarrow \max\{l(i), y_{i-1}\}$ and $r_i \leftarrow 0$ $\{y_0 = 0\}$
    else
        $y_i \leftarrow \min\{z + \rho_i, e(i)\}$ and $r_i \leftarrow \rho_i$
        $S \leftarrow \{k : k < i, y_i < y_k\}$
        $y_k \leftarrow y_i$ and $r_k \leftarrow 0$, for every $k \in S$
        $z \leftarrow y_i + r_i$
    end if
end for
if $z < 1$ **then**
return NO
else
return YES

Iteration i starts with checking whether i can extend the current covered interval $[0, z]$. If it cannot, i is moved to the left as much as possible (power is used only for moving), and it is powered down (r_i is set to 0). If i can extend the current covered interval, it is assigned radius ρ_i, and it is moved to the rightmost possible position, while maximizing the right endpoint of the currently covered interval (i.e., $[0, z]$). If i is located to the left of a sensor j, where $j < i$, then j is moved to y_i.

As for the running time, l, u, s and e can be computed in $O(n)$ time. There are n iterations, each takes $O(n)$ time. Hence, the running time of

Algorithm **Fixed** is $O(n^2)$. It remains to prove the correctness of the algorithm.

Theorem 3.3.3. *Given a constrained* BCFR *instance and* t*, Algorithm* ***Fixed*** *decides whether* t *is achievable.*

Proof. If $u(i) < l(i)$ for some i, then no deployment that satisfies the order $\prec$ exists by Observation 3.3.1. Hence, the algorithm responds correctly.

We show that if the algorithm outputs YES, then the computed solution is feasible. First, notice that $y_{i-1} \leq y_i$, for every i, by construction. We prove by induction on i, that $y_j \in [l(j), u(j)]$ and that $y_j \in [s(j), e(j)]$, if $r_j = \rho_j$, for every $j \leq i$. Consider the ith iteration. If $t\rho_i^\alpha > b_i$ or $z \notin [s(i) - \rho_i, e(i) + \rho_i)$, then $y_i \in [l(i), u(i)]$, since $\max\{l(i), y_{i-1}\} \leq \max\{u(i), u(i-1)\} \leq u(i)$. Otherwise, $y_i = \min\{z + \rho_i, e(i)\} \geq s(i)$, since $z \geq s(i) - \rho_i$. Hence, if $r_i = \rho_i$, we have that $y_i \in [s(i), e(i)]$. Furthermore, if $j < i$ is moved to the left to i, then $y_j = y_i \geq s(i) \geq l(i) \geq l(j)$. Finally, let z_i denote the value of z after the ith iteration. (Initially, $z_0 = 0$.) We proof by induction on i that $[0, z_i]$ is covered. Consider iteration i. If $r_i = 0$, then we are done. Otherwise, $z_{i-1} \in [y_i - \rho_i, y_i + \rho_i]$ and $z_i = y_i + \rho_i$. Furthermore, the sensors in S can be powered down and moved, since $[y_j - r_j, y_j + r_j] \subseteq [y_i - \rho_i, y_i + \rho_i]$, for every $j \in S$.

Finally, we show that if the algorithm outputs NO, there is no feasible solution. We prove by induction that $[0, z_i]$ is the longest interval that can be covered by sensors $1, \ldots, i$. In the base case, observe that $z_0 = 0$ is optimal. For the induction step, let y' be a deployment of $1, \ldots, i$ that covers the interval $[0, z'_i]$. Let $[0, z'_{i-1}]$ be the interval that y' covers by $1, \ldots, i-1$. By the inductive hypothesis, $z'_{i-1} \leq z_{i-1}$. If $t\rho_i^\alpha > b_i$ or $z_{i-1} < s(i) - \rho_i$, it follows that $z'_i = z'_{i-1} \leq z_{i-1} = z_i$. Otherwise, observe that $y'_i \leq y_i$ and therefore $z'_i \leq z_i$. □

3.3.2 Variable Radii

We present an algorithm that solves the constrained BCVR decision problem.

Before presenting our algorithm, we need a few definitions. Given a BCVR instance (x, b) and $t > 0$, if sensor i moves from x_i to $p \in [l(i), u(i)]$, then we may assume without loss of generality that its radius is as large as possible, namely that $r_i(p, t) = \sqrt[\alpha]{(b_i - a|p - x_i|)/t}$.

Similarly to Algorithm **Fixed**, our algorithm tries to cover $[0, 1]$ by deploying sensors one by one, such that the length of the covered prefix $[0, z]$ is maximized. This motivates the following definitions. Let $d \in [-\frac{b_i}{a}, \frac{b_i}{a}]$ denote the distance traveled by sensor i, where $d > 0$ means traveling right,

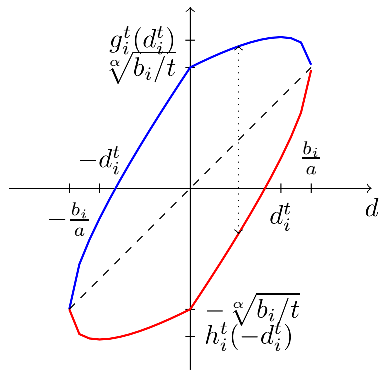


Figure 3.1: Depiction of the functions $g_i^t(d)$ and $h_i^t(d)$ for $a = 2$, $\alpha = 2$, $b_i = 1$, and $t = 4$. The top curve corresponds to $g_i^t(d)$, and the bottom curve corresponds to $h_i^t(d)$. The dashed line corresponds to the location of sensor i, while the vertical interval between the curves is the interval that is covered by i at distance d from x_i.

and $d < 0$ means traveling left. If a sensor travels a distance d, then its lifetime t sustaining radius is given by $\sqrt[\alpha]{(b_i - a|d|)/t}$. Given t, we define: $g_i^t(d) \stackrel{\text{def}}{=} d + \sqrt[\alpha]{(b_i - a|d|)/t}$. $g_i^t(d)$ is the *right reach* of sensor i at distance d from x_i, i.e., the rightmost point that i covers when it has traveled a distance of d and the required lifetime is t. Similarly define $h_i^t(d) \stackrel{\text{def}}{=} g_i^t(-d)$ is the *left reach* of sensor i at distance d from x_i. See depiction in Figure 3.1.

There is a distance d_i^t that maximizes $g_i^t(d)$. Furthermore, d_i^t is a function of the input of the decision version of BCVR. This is a result of the next lemma.

Lemma 3.3.4. *Let $t > 0$. For any i, the distance d_i^t maximizes $g_i^t(d)$, where*

$$d_i^t = \begin{cases} \frac{b_i}{a} - \frac{1}{\alpha}\sqrt[\alpha-1]{\frac{a}{\alpha t}} & \alpha > 1 \\ \frac{b_i}{a} & \alpha = 1, a < t \\ 0 & \alpha = 1, a \geq t \end{cases}$$

$$g_i^t(d_i^t) = \begin{cases} \frac{b_i}{a} + \left(1 - \frac{1}{\alpha}\right)\sqrt[\alpha-1]{\frac{a}{\alpha t}} & \alpha > 1 \\ \frac{b_i}{\min\{a,t\}} & \alpha = 1 \end{cases}$$

If $\alpha > 1$ or $a \neq t$, g_i^t is increasing for $d < d_i^t$, and decreasing for $d > d_i^t$. If $\alpha = 1$ and $a = t$, g_i^t is constant, for $d \geq 0$, and it is increasing for $d < 0$.

Proof. First consider the case where $\alpha > 1$. For $d \in [b_i/a, 0)$ we get

$$\frac{\partial h_i^t}{\partial d} = 1 + \frac{a}{\alpha t}\left(\frac{b_i + ad}{t}\right)^{\frac{1}{\alpha}-1} > 0 \ .$$

For $d \in (0, b_i/a]$, the derivative of g_i^t is given by

$$\frac{\partial g_i^t}{\partial d} = 1 - \frac{a}{\alpha t}\left(\frac{b_i - ad}{t}\right)^{\frac{1}{\alpha}-1} \ .$$

It follows that $\frac{\partial g_i^t}{\partial d}(d) = 0$ when

$$d = d_i^t = \frac{b_i}{a} - \frac{t}{a}\left(\frac{a}{\alpha t}\right)^{\alpha/(\alpha-1)} = \frac{b_i}{a} - \frac{1}{\alpha}\sqrt[\alpha-1]{\frac{a}{\alpha t}} \ .$$

Furthermore, $\frac{\partial g_i^t}{\partial d}(d) > 0$ when $d < d_i^t$, and $\frac{\partial g_i^t}{\partial d}(d) < 0$ when $d > d_i^t$. The radius at this distance is $\sqrt[\alpha-1]{\frac{a}{\alpha t}}$. The maximum reach is thus

$$g_i^t(d_i^t) = \frac{b_i}{a} + \left(1 - \frac{1}{\alpha}\right)\sqrt[\alpha-1]{\frac{a}{\alpha t}} \ .$$

For $\alpha = 1$ we have

$$g_i^t(d) = \begin{cases} d(1 - a/t) + b_i/t & d \geq 0, \\ d(1 + a/t) + b_i/t & d < 0. \end{cases}$$

Hence,

$$\frac{\partial g_i^t(d)}{\partial d} = \begin{cases} 1 - a/t & d > 0, \\ 1 + a/t & d < 0. \end{cases}$$

If $d > 0$, we have several cases. If $a > t$, the maximum occurs at $d_i^t = 0$ and $g_i^t(d_i^t) = \frac{b_i}{t}$. If $a = t$, $g_i^t(d) = \frac{b_i}{t}$, for any $d \leq \frac{b_i}{a}$. If $a < t$, the function is increasing for any $d \leq \frac{b_i}{a}$, and thus $d_i^t = \frac{b_i}{a}$ and $g_i^t(d_i^t) = \frac{b_i}{a}$. Hence, $g_i^t(d_i^t) = \frac{b_i}{\min\{a,t\}}$. □

Given a point $z \in [0, 1]$, the *attaching position* of sensor i to z, denoted by $p_i(z, t)$, is the position p for which $p - r_i(p, t) = z$ such that $p + r_i(p, t)$ is maximized, if such a position exist. If such a point does not exist we define $p_i(z, t) = \infty$. Observe that by Lemma 3.3.4 there may be at most two points that satisfy the equation $p - r_i(p, t) = z$. Such a position can either be found explicitly or numerically as it involves solving an equation of degree α.

Algorithm **Variable** is our decision algorithm for BCVR. It first computes u and l. If there is a sensor i, such that $l(i) > u(i)$, it outputs NO. Then, it deploys the sensors one by one according to $\prec$ with the goal of extending the coverage interval $[0, z]$. If i cannot increase the covering interval it is placed at $\max\{l(i), y_{i-1}\}$ so as not to block sensor $i+1$. If i can increase coverage, it is placed in $[l(i), u(i)]$ such that z is covered and coverage to the right is maximized. It may be the case that the best place for i is to the

Algorithm 2 : **Variable** (x, b, t)

Compute l and u
if there exists i such that $u(i) < l(i)$ **then**
return NO
$z \leftarrow 0$
for $i = 1 \to n$ **do**
  $q_L(i) \leftarrow \min\left\{\max\left\{x_i - d_i^t, l(i)\right\}, u(i)\right\}$
  $q_R(i) \leftarrow \max\left\{\min\left\{x_i + d_i^t, u(i)\right\}, l(i)\right\}$
  if $z \notin [q_L(i) - r_i(q_L(i), t), q_R(i) + r_i(q_R(i), t)]$ **then**
    $y_i \leftarrow \max\{l(i), y_{i-1}\}$ and $r_i \leftarrow 0$ $\{y_0 = 0\}$
  else
    $y_i \leftarrow \max\left\{\min\left\{p_i(z, t), u(i), x_i + d_i^t\right\}, l(i)\right\}$ and $r_i \leftarrow r_i(y_i, t)$
    $S \leftarrow \{k : k < i, y_i < y_k\}$
    $y_k \leftarrow y_i$ and $r_k \leftarrow 0$, for every $k \in S$
    $z \leftarrow y_i + r_i$
  end if
end for
if $z < 1$ **then**
return NO
else
return YES

left of previously positioned sensors. In this case the algorithm moves the sensors such that coverage and order are maintained. Finally, if $z < 1$ after placing sensor n, the algorithm outputs NO, and otherwise it outputs YES.

l and u can be computed in $O(n)$ time. There are n iterations of the main loop, each taking $O(n)$ time (assuming that computing $p_i(z, t)$ takes $O(1)$ time), thus the running time of the algorithm is $O(n^2)$.

In order to analyze Algorithm **Variable** we define

$$P(i) = \{p : p \in [l(i), u(i)] \text{ and } z \in [p - r_i(p, t), p + r_i(p, t)]\} \ .$$

$P(i)$ is the set of points from which sensor i can cover z. Observe that $P(i)$ is an interval due to Lemma 3.3.4. Hence, we write $P(i) = [p_L(i), p_R(i)]$.

In the next two lemmas it is shown that when the algorithm checks whether $z \notin [q_L(i) - r_i(q_L(i), t), q_R(i) + r_i(q_R(i), t)]$ it actually checks whether $P(i) = \emptyset$, and that $y_i^* \stackrel{\text{def}}{=} \max\left\{\min\left\{p_i(z,t), u(i), x_i + d_i^t\right\}, l(i)\right\}$ is equal to $\operatorname{argmax}_{p \in P}\left\{p + r_i(p,t)\right\}$. Hence, in each iteration we check whether $[0, z]$ can be extended, and if it can, we take the best possible extension.

Lemma 3.3.5. $[p_L(i), p_R(i)] \subseteq [q_L(i), q_R(i)]$*. Moreover,* $P(i) = \emptyset$ *if and only if* $z \notin [q_L(i) - r_i(q_L(i), t), q_R(i) + r_i(q_R(i), t)]$.

Proof. By Lemma 3.3.4 $q_L(i)$ is the location that maximized coverage to the left, and $q_R(i)$ is the location that maximized coverage to the right. □

Lemma 3.3.6. *If* $P(i) \neq \emptyset$*, then* $y_i^* = \operatorname{argmax}_{p \in P(i)}\left\{p + r_i(p,t)\right\}$.

Proof. By Lemma 3.3.4, there are three cases:

- If $x_i + d_i^t \in P(i)$, then $\operatorname{argmax}_{p \in P(i)}\left\{p + r_i(p,t)\right\} = x_i + d_i^t$.

  $y_i^* = x_i + d_i^t$, since $p_i(z,t) \geq x_i + d_i^t$.

- If $x_i + d_i^t > p_R(i)$, then $\operatorname{argmax}_{p \in P(i)}\left\{p + r_i(p,t)\right\} = p_R(i)$.

  $y_i^* = \min\left\{p_i(z,t), u(i)\right\}$, since $p_R(i) = \min\left\{p_i(z,t), u(i)\right\} \geq l(i)$.

- If $x_i + d_i^t < p_L(i)$, then $\text{argmax}_{p \in P(i)} \{p + r_i(p,t)\} = p_L(i)$.

  $y_i^* = l(i)$, since $q_L(i) = l(i) > x_i + d_i^t \geq \min\{p_i(z,t), u(i), x_i + d_i^t\}$.

$\square$

The following proof of correctness of **Variable** relies on the following precise computation assumptions: (i) real arithmetic, and (ii) the ability to solve equations of degree α (note that equations of degree $\alpha = 1, 2$ have a closed-form solution).

Theorem 3.3.7. *Given a constrained* BCVR *instance and $t > 0$, Algorithm* ***Variable*** *correctly decides whether t is achievable assuming precise computation.*

Proof. If $u(i) < l(i)$ for some i, then no deployment that satisfies the order $\prec$ exists by Observation 3.3.1. Hence, the algorithm responds correctly.

We show that if the algorithm outputs YES, then the computed solution is feasible. First, notice that $y_{i-1} \leq y_i$, for every i, by construction. We prove by induction on i, that $y_j \in [l(j), u(j)]$ for every $j \leq i$. Consider the ith iteration. If $z \notin [q_L(i) - r_i(q_L(i),t), q_R(i) + r_i(q_R(i),t)]$, then $y_i \in [l(i), u(i)]$, since $\max\{l(i), y_{i-1}\} \leq \max\{u(i), u(i-1)\} \leq u(i)$. Otherwise, $y_i = \max\{\min\{p_i(z,t), u(i), x_i + d_i^t\}, l(i)\} \in [l(i), u(i)]$. Furthermore, if $j < i$ is moved to the left of i, then $y_j = y_i \geq l(i) \geq l(j)$. Finally, let z_i denote the value of z after the ith iteration. (Initially, $z_0 = 0$.) We prove by induction on i that $[0, z_i]$ is covered. Consider iteration i. If $r_i = 0$, then we are done. Otherwise, $z_{i-1} \in [y_i - r_i, y_i + r_i]$ and $z_i = y_i + r_i$, and the sensors in S can be powered down and moved, since $[y_j - r_j, y_j + r_j] \subseteq [y_i - r_i, y_i + r_i]$, for every $j \in S$.

Finally, we show that if the algorithm outputs NO, there is no feasible solution. We prove by induction that $[0, z_i]$ is the longest interval that can be covered by sensors $1, \ldots, i$. In the base case, observe that $z_0 = 0$ is optimal. For the induction step, let y' be a deployment of $1, \ldots, i$ that covers the interval $[0, z'_i]$. Let $[0, z'_{i-1}]$ be the interval that it covers by $1, \ldots, i-1$. By the inductive hypothesis, $z'_{i-1} \leq z_{i-1}$. If $z'_i \leq z_{i-1}$, then we are done. Otherwise, we have that $y'_i + r_i(y'_i, t) > z_{i-1}$. In this case we have that $y'_i \in P(i)$. It follows, by Lemma 3.3.5, that we place i at $y_i = y^*_i$. By Lemma 3.3.6 we have y_i is better than y'_i in terms of coverage to the right, namely $z_i = y_i + r_i(y_i, t) \geq y'_i + r_i(y'_i, t) = z'_i$. □

Alternatively, assume that numbers are represented using k bits, ($k = 32$ or "double" precision is common). The problem is now changed as the space of possible solutions is a subset of the actual BCVR solution space, and thus we are not guaranteed to find an optimal solution for BCVR. For this k bit model, we can solve equations of degree α using binary search in polynomial time (in the input size and in k), and moreover decide if t is feasible in polynomial time. Furthermore, under this assumption we can find an optimal k bit solution, since we can run the parametric search algorithm in the next subsection with an additive error ϵ that is smaller than the granularity that is provided by the k bit model.

3.3.3 Parametric Search Algorithms

Since we have algorithm that, given t and an order $\prec$, decides whether there exists a solution that satisfies $\prec$ with lifetime t, we can perform a binary search on t. The maximum lifetime of a given instance is bounded by the lifetime of this instance in the case where $a = 0$. For $a = 0$, the network lifetime in the fixed case is at most $\max_i \{b_i/\rho_i^\alpha\}$, and it is $(2\sum_j \sqrt[\alpha]{b_j})^\alpha$ (Theorem 3.2.1) in the variable radii case. These expression serve as upper bounds for the case where $a > 0$. Hence, the running time of the parametric search in polynomial in the input size and in the $\log \frac{1}{\varepsilon}$, where ε is the accuracy parameter.

3.4 Sensors are Located on the Edges of the Barrier

Consider the case where the initial locations are on either edge of the barrier, namely, $x \in \{0, 1\}^n$. For both BCVR and BCFR we show that, given an achievable lifetime t, there exists a solution with lifetime t in which the sensors satisfy a certain ordering. In the case of BCVR, the ordering depends only on the battery sizes, and hence we may use the parametric search algorithm for constrained BCVR from Section 3.3. In the case of BCFR, the ordering depends on t, and therefore may change. Even so, we may use parametric search for this special case of BCFR since, given t, the ordering can be computed in polynomial time.

Fixed radii. We start by considering the special case of BCFR in which all sensors are located at $x = 0$. The case where $x = 1$ is symmetric. Given a BCFR instance $(0, b, \rho)$ and a lifetime t, the *maximum reach* of sensor i is defined as the farthest point from its initial position that sensors i can cover while maintaining lifetime t, and is given by: $f_t(i) = \frac{1}{a}(b_i - t\rho_i^\alpha) + \rho_i$, if $t\rho_i^\alpha \leq b_i$, and $f_t(i) = 0$, otherwise. We assume without loss of generality that the sensors are ordered according to *reach ordering*, namely that $i < j$ if and only if $f_t(i) < f_t(j)$. Also, we ignore sensors with zero reach, since they must power down. Hence, if $f_t(i) = 0$, we place i at 0 and set its radius to 0. Let t be an achievable lifetime, we show that there exists a solution (y, r) with lifetime t such that sensors are deployed according to reach ordering.

Lemma 3.4.1. *Let* $(0, b, \rho)$ *be a* BCFR *instance and let* $p \in (0, 1]$*. Suppose that there exists a solution that covers* $[0, p]$ *for* t *time. Then, there exists a solution that covers* $[0, p]$ *for* t *time that satisfies reach ordering.*

Proof. We first prove that we may focus on feasible solutions where $r = \rho$. Given a feasible solution (y, r) that covers $[0, p]$ with lifetime t, we define $y'_i = y_i$, if $r_i = \rho_i$, and $y'_i = 0$, otherwise. The pair (y', ρ) clearly covers $[0, p]$ with lifetime t. (Recall that we ignore sensors with zero reach.)

Given a solution that covers $[0, p]$ with lifetime t, a pair of sensors is said to violate reach ordering if $i < j$ and $y_i > y_j$. Let (y, ρ) be a solution with lifetime t for $(0, b, \rho)$ that minimizes reach ordering violations. If there are no violations, then we are done. Otherwise, we show that the number of violations can be decreased.

If y has ordering violations, then there must exist at least one violation due to a pair of adjacent sensors. Let i and j be such sensors. If the barrier is covered without i, then i is moved to y_j. (Namely $y'_k = y_k$, for every $k \neq i$, and $y'_i = y_j$.) y' is feasible, since i moves to the left. Otherwise, if the barrier is covered without j, then j is moved to $y'_j = \min\{y_i, f_t(j) - \rho_j\}$. If $y'_j = y_i$, then we are done. If $y'_j < y_i$, then $[y_i - \rho_i, y_i + \rho_i] \subseteq [y_j - \rho_j, y_j + \rho_j]$, since $f_t(j) > f_t(i)$. It follows that the barrier is covered without i, and so we can move i to y'_j. Since $y'_j \leq f_t(j) - \rho_j$, and i moves to the left, we get a feasible deployment.

If both sensors participate in the cover, we define a new deployment y' by moving i to $y'_i = y_j + (\rho_i - \rho_j)$ and moving j to $y'_j = y_i + (\rho_i - \rho_j)$. The interval $[0, p]$ is covered, since $[y_j - \rho_j, y_i + \rho_i]$ is covered. Also, $y'_i \leq y'_j$. Furthermore, i and j can maintain their radii for t time, since $y'_i \leq y_i$ and $f_t(j) > f_t(i)$. Since i moves to the left, it may bypass several sensors. In this case we move all sensors with smaller reach that were bypassed by i, to y'_i. Since j moves to the right, it may bypass several sensors. As long as there is a sensor with larger reach that was bypassed by j, let k be the rightmost such sensor, and move both j and k to $\min\left\{y'_j, f_t(k) - \rho_k\right\}$. Notice that k is not needed for covering to the left of y'_j, and thus it can be moved to the right, as long as it has the power to do so. If k cannot move to y'_j, it follows that j is not needed for covering to the right of y'_k.

In all cases, we get a deployment y' that covers $[0, p]$ with lifetime t with a smaller number of violations than y. A contradiction. $\square$

Variable radii. We now consider BCVR with $x = 0$. As before, the case of $x = 1$ is symmetric. Given a BCVR instance $(0, b)$ and a lifetime t, the *maximum reach* of sensor i is $g_i^t(d_i^t)$. Note that if the sensors are ordered by battery size, namely that $i < j$ if and only if $b_i < b_j$, they are also ordered by reach. Thus, we assume in the following that sensors are ordered by battery size. Let t be an achievable lifetime. We show that there exists a deployment y with lifetime t such that sensors are deployed according to the battery ordering, namely $b_i \le b_j$ if and only if $y_i \le y_j$.

For the proof we need the result of Lemma 2.3.4 which is restated below.

Lemma 2.3.4. *Let $\eta_1, \eta_2, \gamma_1, \gamma_2 \ge 0$ such that (i) $\gamma_1 < \eta_1 \le \eta_2$, and (ii) $\eta_1 + \eta_2 \ge \gamma_1 + \gamma_2$. Also let $\alpha \ge 1$. Then, $\sqrt[\alpha]{\eta_1} + \sqrt[\alpha]{\eta_2} \ge \sqrt[\alpha]{\gamma_1} + \sqrt[\alpha]{\gamma_2}$.*

Lemma 3.4.2. *Let $(0, b)$ be a* BCVR *instance and let $p \in (0, 1]$. Suppose that there exists a deployment that covers $[0, p]$ for t time. Then, there exists a deployment that covers $[0, p]$ for t time that satisfies battery ordering.*

Proof. Given a solution that covers $[0, p]$ with lifetime t, a pair of sensors is said to violate battery ordering if $b_i < b_j$ and $y_i > y_j$. Let y be a solution with lifetime t for $(0, b)$ that minimizes battery ordering violations. If there are no violations, then we are done. Otherwise, we show that the number of violations can be decreased. If y has ordering violations, then there must exist at least one violation due to a pair of adjacent sensors. Let i and j be such sensors. We assume, without loss of generality, that the batteries of both i and j are depleted at t, namely that $r_k = \sqrt[\alpha]{(b_k - a|y_k - x_k|)/t}$, for $k = i, j$.

If the barrier is covered without i, then i is moved to y_j. (Namely $y'_k = y_k$, for every $k \neq i$, and $y'_i = y_j$.) y' is feasible, since i moves to the left. Otherwise, if the barrier is covered without j, then j is moved to y_i and j's radius is decreased accordingly. Otherwise, both sensors actively participate in covering the barrier, which means that the interval $[y_j - r_j, y_i + r_i]$ is covered by i and j. In this case, we place i at y'_i with radii r'_i, such that $y'_i - r'_i = y_j - r_j$. We place j at the rightmost location y'_j such that $y'_j \leq y_i$ and $y'_j - r'_j \leq y'_i + r'_i$. If $y'_j = y_i$ then we are done, as sensor j has more battery power at y_i than i does at y_i. Otherwise, we may assume that $y'_j - r'_j = y'_i + r'_i$. We show that it must be that $y'_j + r'_j \geq y_i + r_i$. We have that $y'_i < y_j$ and $y'_j < y_i$. It follows that $\beta'_i + \beta'_j > \beta_i + \beta_j$, where $\beta_i = b_i - a y_i$. Also, notice that $\beta_i < \beta'_j < \beta_j$ and $\beta_i < \beta'_i < \beta_j$. It follows that $r'_i + r'_j = \sqrt[\alpha]{\beta'_i/t} + \sqrt[\alpha]{\beta'_j/t} > \sqrt[\alpha]{\beta_i/t} + \sqrt[\alpha]{\beta_j/t} = r_i + r_j$, where the inequality is due to Lemma 2.3.4. Hence, $y'_j + r'_j = (y_j - r_j) + 2r'_i + 2r'_j > (y_j - r_j) + 2r_i + 2r_j \geq y_i + r_i$.

Since i moves to the left, it may bypass several sensors. In this case we move all sensors with smaller batteries that were bypassed by i, to y'_i. This can be done since these sensors are not needed for covering to the right of $y'_i - r'_i$. Similarly, since j moves to the right, it may bypass several sensors. As long as there is a sensor with larger reach that was bypassed by j, let k be the rightmost such sensor. Notice that k is not needed for covering to the left of y'_j. Hence, if $y_k + r_k \geq y'_j + r'_j$, we move j to y_k. Otherwise, we move k to y'_j.

In all cases, we get a deployment y' that covers $[0, p]$ with lifetime t with a smaller number of violations than y. A contradiction. □

Separation. We are now ready to tackle the case where $x \in \{0,1\}^n$. We start with the fixed radii case. Given a BCFR instance (x, b, r) and a lifetime t, we assume without loss of generality that the sensors are ordered according to the following *bi-directional reach* order: sensors initially located at 0 are positioned to the left of sensors initially located at 1, sensors initially located at 0 are positioned according to reach order, and sensors initially at 1 are positioned according to reverse reach order. We show that we may assume that the sensors are deployed using the bi-directional reach order. The first step is to show that the sensors that are located at 0 are deployed to the left of the sensors that are placed at 1.

Lemma 3.4.3. *Let* (x, b, ρ) *be a* BCFR *instance, where* $x \in \{0,1\}^n$*, and let*

t be an achievable lifetime. Then, there exists a feasible solution (y, r) with lifetime t such that $y_i \leq y_j$, for every $i \leq \ell < j$, where ℓ is the number of sensors initially located at 0.

Proof. Given a deployment y for (x, b, r), a pair of sensors is called *bad* if $i \leq \ell < j$ and $y_i > y_j$. Let y be a deployment with lifetime t for (x, b, r) that minimizes the number of bad pairs. If there are no bad pairs, then we are done. Otherwise, we show that the number of bad pairs can be decreased. If y has a bad pair, then there must exist at least one bad pair of adjacent sensors. Let i and j be such sensors. We construct a new deployment vector y' as follows.

If the barrier is covered without i, then i is moved to 0, namely $y'_k = y_k$, for every $k \neq i$, and $y'_i = 0$. Otherwise, if the barrier is covered without j, then j is moved to 1, namely $y'_k = y_k$, for every $k \neq j$, and $y'_i = 1$. In both cases the pair (y', r) is feasible and has lifetime t. Furthermore the number of bad pairs decreases. A contradiction.

If both i and j are essential to the cover, we define y' as follows:

$$y'_k = \begin{cases} y_j + (\rho_i - \rho_j) & k = i, \\ y_i + (\rho_i - \rho_j) & k = j, \\ y_k & k \neq i, j. \end{cases}$$

We show that (y', r) is a feasible solution. First, notice that $y'_i = y_j + (\rho_i - \rho_j) < y_i$, since otherwise the barrier can be covered without j. Similarly,

$y'_j = y_i + (\rho_i - \rho_j) > y_j$. Hence, $y'_k \leq y_k$, for $k \leq \ell$, and $y'_k \geq y_k$, for $k > \ell$, which means that y' consumes less power than y. Also the barrier is covered, since the interval $[y_j - \rho_j, y_i + \rho_i]$ is covered by i and j. Finally, $y'_i = y_j + (\rho_i - \rho_j) \leq y_i + (\rho_i - \rho_j) = y'_j$, and therefore the number of bad pair decreases. A contradiction. □

Next we show that we may assume that the sensors are deployed using the bi-directional reach order.

Theorem 3.4.4. *Let (x, b, ρ) be a* BCFR *instance where $x \in \{0,1\}^n$ and let t be an achievable lifetime. Then there exists a feasible solution (y, r) with lifetime t such that the sensors are deployed using bi-directional reach order.*

Proof. By Lemma 3.4.3 we know that there exists a deployment y, such that $y_i \leq y_j$, for every $i \leq \ell < j$, where ℓ is the number of sensors initially located at 0. It follows that sensors from 0 cover $[0, p_0]$ while sensors from 1 cover $[p_1, 1]$, where $p_0 \geq p_1$. Lemma 3.4.1 implies that there is a deployment y^0 of the sensors from 0 that covers $[0, p_0]$ that satisfies reach order, and that there is a deployment y^1 of sensors from 1 that covers $[p_1, 1]$ that satisfies reverse reach order. Define

$$y'_i = \begin{cases} y^0_i & i \leq \ell, \\ \max\{y^1_i, y^0_\ell\} & i > \ell. \end{cases}$$

y' covers $[0, 1]$ and it satisfies the bi-directional reach order. □

We treat the variable radii case similarly. Given a BCVR instance (x, b), we assume without loss of generality that the sensors are ordered according to a *bi-directional battery* order: sensors initially located at 0 are positioned to the left of sensors initially located at 1, sensors initially located at 0 are positioned according to battery order, and sensors initially at 1 are positioned according to reverse battery order. The proofs of the next lemma and theorem are nearly identical to the proofs of Lemma 3.4.3 and Theorem 3.4.4 respectively with the only differences being ρ replaced by r, (x, b, r) replaced by the instance for BCVR (x, b), and the references to Lemmas 3.4.3 and 3.4.1 in the proof of Theorem 3.4.4 being replaced by Lemmas 3.4.5 and 3.4.2 repectively in the proof of Theorem 3.4.6.

Lemma 3.4.5. *Let (x, b) be a* BCVR *instance where $x \in \{0, 1\}^n$, and let t be an achievable lifetime. Then, there exists a feasible solution (y, r) with lifetime t such that $y_i \leq y_j$, for every $i \leq \ell < j$, where ℓ is the number of sensors initially located at* 0.

Theorem 3.4.6. *Let (x, b) be a* BCVR *instance where $x \in \{0, 1\}^n$, and let t be an achievable lifetime. Then there exists a feasible solution (y, r) with lifetime t such that the sensors are deployed using bi-directional battery order.*

3.5 Uniform Instances

In this section we consider uniform instances for both BCFR and BCVR. We define the uniform instances for each problem in the following. We show that there exist non-swapping optimal solutions for these instances and that they can therefore be solved by the parametric search algorithms. A solution is called non-swapping if the initial order of the sensors is the same as the final order, that is, $y_i \leq y_j$ for $i < j$.

3.5.1 Fixed Radii

A uniform instance for BCFR is defined to be an instance where $b_i = b$ and $\rho_i = \rho$ for every i. It is an instance where the battery levels of all sensors are the same and equal to b, and the coverage radii of all sensors are equivalent and equal to ρ.

Lemma 3.5.1. *Let (x, b, ρ) be a uniform* BCFR *instance. Then there exists a non-swapping solution with maximum lifetime.*

Proof. Let (y, r) be a feasible solution with lifetime t. First, notice that if $r_i = 0$, for some sensor i, then we can deploy i at x_i and set $r_i = \rho_i$. Sensor i can work for t time, since there is at least one other sensor that works for t time, if $t > 0$. Hence, we may assume that $r = \rho$. We show that there exists a deployment y' such that (y', ρ) is feasible and $L(y', \rho) \geq t$.

Given a solution (y, ρ), a pair of sensors is said to swap if $x_i < x_j$ and $y_i > y_j$. Let (y, ρ) be a solution with lifetime t for (x, b, ρ) that minimizes the number of swaps. If there are no swaps, then we are done. Otherwise, we show that the number of swaps may be decreased.

If there are swaps, then there must exist at least one swap due to a pair of adjacent sensors. Let i and j be such sensors. Consider the solution y' obtained from y by interchanging the final positions of i and j, i.e., $y'_i = y_j$, $y'_j = y_i$, and $y'_k = y_k$, for $k \neq i, j$. We show that the lifetime does not decrease, since the maximum distance traveled by a sensor does not increase.

If both sensors move to the right, then we have that $x_i < x_j \leq y_j < y_i$. In this case $y'_i - x_i = y_j - x_i < y_i - x_i$ and $y'_j - x_j = y_i - x_j < y_i - x_i$. The case where both sensors move to the left is symmetric. Suppose that i moved to the right and j moved to the left. If $x_i \leq y_j < y_i \leq x_j$, then both i and j move less in y'. If $y_j < x_i < x_j < y_i$, then assume wlog that the largest distance traveled in y is by sensor j. In y', both sensor i and j travel less than sensor j in y. If $x_i \leq y_j \leq x_j < y_i$, then i moves less in y' and j moves less in y' than i in y since $y_i - x_j < y_i - x_i$. The case where $y_j < x_i \leq y_i \leq x_j$ is symmetric. It follows that (y', ρ) has lifetime at least t and y' has less swaps than y. A contradiction. $\square$

Theorem 3.3.3 implies that uniform BCFR can be solved using parametric search.

Theorem 3.5.2. *Uniform* BCFR *can be solved in polynomial time.*

3.5.2 Variable Radii

A uniform instance of BCVR is defined to be an instance where $b_i = b$ for every i, i.e. battery levels are uniform.

Lemma 3.5.3. *Let* (x, b) *be a uniform* BCVR *instance. There there exists a non-swapping solution* (y, r) *with maximum lifetime.*

Proof. Let t^* be the optimal solution for a uniform BCVR instance. Assume that every solution for (x, b) with lifetime t^* is swapping. By Lemma 2.3.5, there is an optimal solution for sensors initially at x which maximizes the minimum energy used for t^* time that is non-swapping. This solution must have optimal value at most b as the instance of BCVR has optimal solution t^* where initial battery levels are b. Consider the solution with same position and radii for the uniform BCVR instance. Clearly such a solution is feasible along with being non-swapping. This solution has lifetime t^*. Contradiction. □

Theorem 3.3.7 implies that uniform BCVR can be solved using parametric search.

Theorem 3.5.4. *Uniform* BCVR *can be solved in polynomial time assuming precise computations.*

3.6 Hardness Results

In this section we show that (i) BCFR is NP-hard, even if $x \in p^n$, for any $p \in (0, 1)$. (ii) There is no polynomial time multiplicative approximation algorithm for BCFR, unless P=NP, even if $x = p^n$. (iii) There is no polynomial time algorithm that computes a solution within an additive factor ε, for some constant $\varepsilon > 0$, unless P=NP, even if $x = p^n$. (iv) BCVR is strongly NP-hard. The hardness results apply to any $a \in (0, \infty)$ and $\alpha \geq 1$.

We note that throughout the section we assume that α is integral for ease of presentation. More specifically, we assume that exponentiation with exponent α can be done in polynomial time. Our constructions can be fixed by taking a numerical approximation which is slightly larger than the required power.

3.6.1 Fixed Radii

The first result is obtained using a reduction from PARTITION.[1] Roughly speaking, our reduction uses a sensor that cannot move if it is required to maintain its radius for one unit of time. This sensor splits the line into two segments, and therefore the question of whether the given numbers can be partitioned into two subsets of equal sum translates into the question of whether we can cover the two segments for some time interval.

Lemma 3.6.1. BCFR *is NP-hard, for any* $a \in (0, \infty)$ *and* $\alpha \geq 1$*, even if* $x = \frac{1}{2}^n$*. Furthermore, in this case it is NP-hard to decide whether the maximum lifetime is zero or at least* a*.*

Proof. Given a PARTITION instance $a_1, ..., a_n$, let $B = \sum_i a_i$. We construct a BCFR instance with $n + 1$ sensors as follows: $x_i = \frac{1}{2}$, for every i;

$$\rho_i = \begin{cases} \frac{a_i}{2(B+1)} & i \leq n, \\ \frac{1}{2(B+1)} & i = n+1; \end{cases} \qquad b_i = \begin{cases} a\rho_i^\alpha + \frac{a}{2} & i \neq n+1, \\ a\rho_i^\alpha & i = n+1. \end{cases}$$

We show that $(a_1, \ldots, a_n) \in$ PARTITION implies that there exists a solution with lifetime a, and that the maximum lifetime is zero if $(a_1, \ldots, a_n) \notin$ PARTITION.

Suppose that $(a_1, \ldots, a_n) \in$ PARTITION, and let $I \subseteq \{1, \ldots, n\}$ be such that $\sum_{i \in I} a_i = \frac{1}{2} \sum_i a_i$. Set $r_i = \rho_i$, for every sensor i. Use sensor $n + 1$

[1] A PARTITION instance consists of a list $a_1, \ldots, a_n$ of positive integers, and the goal is to decide whether there exists $I \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in I} a_i = \sum_{i \notin I} a_i$.

to cover the interval $[\frac{1}{2} - \frac{1}{2B+2}, \frac{1}{2} + \frac{1}{2B+2}]$, the sensors that correspond to I to cover the interval $[0, \frac{1}{2} - \frac{1}{2B+2}]$, and the rest of the sensors to cover the interval $[\frac{1}{2} + \frac{1}{2B+2}, 1]$. This is possible, since $\sum_{i \in I} 2\rho_i = \frac{1}{2} - \frac{1}{2B+2}$, and $\sum_{i \in \{1,\ldots,n\} \setminus I} 2\rho_i = \frac{1}{2} - \frac{1}{2B+2}$. It is not hard to verify that a lifetime of a is achievable.

Suppose that $(a_1, \ldots, a_n) \notin$ Partition, and assume that there exists a solution (y, r) with non-zero lifetime. It must be that $r_i = \rho_i$, for every i, since $\sum_i 2\rho_i = 1$. Since $\alpha \geq 1$, sensor $n + 1$ cannot move more than $\frac{1}{2B+2}$. It follows that $y_{n+1} = \frac{1}{2}$, since all radii are multiples of $\frac{1}{2B+2}$. Thus there is a subset $I \subseteq \{1, \ldots, n\}$ of sensors that covers $[0, \frac{1}{2} - \frac{1}{2B+2}]$, and $\sum_{i \in I} a_i = (B + 1) \sum_{i \in I} 2\rho_i = \frac{1}{2}B$. Hence, $(a_1, \ldots, a_n) \in$ Partition. A contradiction. □

The next step is to prove a similar result for any $p \in (0, 1)$. Since we already considered $p = \frac{1}{2}$, we assume, without loss of generality, that $p < \frac{1}{2}$.

Lemma 3.6.2. BCFR *is NP-hard, for any $a \in (0, \infty)$ and $\alpha \geq 1$, even if $x = p^n$, where $p \in (0, \frac{1}{2})$. Furthermore, in this case it is NP-hard to decide whether the maximum lifetime is zero or at least a.*

Proof. Given a Partition instance $a_1, ..., a_n$, let $B = \sum_i a_i$. We construct a BCFR instance with $n+3$ sensors as follows: $x_i = p$, for every i;

$$\rho_i = \begin{cases} \frac{a_i d}{2(B+1)} & i \le n, \\ \frac{d}{2(B+1)} & i = n+1, \\ \frac{p-d/2}{2} & i = n+2, \\ \frac{1-p-d/2}{2} & i = n+3; \end{cases} \qquad b_i = \begin{cases} a\rho_i^\alpha + a & i \ne n+1, \\ a\rho_i^\alpha & i = n+1. \end{cases}$$

where $d = \min\{p, 1-2p\}$. We show that $(a_1, \ldots, a_n) \in$ PARTITION implies that there exists a solution with lifetime a, and that the maximum lifetime is zero if $(a_1, \ldots, a_n) \notin$ PARTITION.

Suppose that $(a_1, \ldots, a_n) \in$ PARTITION, and let $I \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in I} a_i = \sum_{i \notin I} a_i$. Define $\bar{I} = \{1, \ldots, n\} \setminus I$. Set $r_i = \rho_i$, for every i, and use the following deployment:

1. Sensor $n+1$ does not move and covers $[p - \frac{d}{2B+2}, p + \frac{d}{2B+2}]$.

2. Sensor $n+2$ moves to $\frac{p-d/2}{2}$ and covers $[0, p - d/2]$.

3. Sensor $n+3$ moves to $\frac{1+p+d/2}{2}$ and covers $[p + d/2, 1]$.

4. The sensors that correspond to I deploy such that they cover $[p - d/2, p - \frac{p}{2B+2}]$.

5. The sensors that correspond to $\bar{I}$ deploy such that they cover $[p + \frac{p}{2B+2}, p + d/2]$.

(See example in Figure 3.2.) This is possible, since $\sum_{i \in I} 2\rho_i = \sum_{i \in I} \frac{a_i d}{B+1} =$

$\frac{Bd}{2(B+1)} = \frac{d}{2} - \frac{d}{2B+2}$, and similarly $\sum_{i \in \bar{I}} 2r_i = \frac{d}{2} - \frac{d}{2B+2}$. It is not hard to verify that a lifetime of a is achievable.

Suppose that $(a_1, \ldots, a_n) \notin$ PARTITION, and assume that there exists a solution (y, r) with non-zero lifetime. Notice that $\sum_i 2\rho_i = 1$, and thus it must be that $r_i = \rho_i$, for every i. Since $\alpha \geq 1$, the battery of sensor $n+1$ is depleted if it moves a distance of $\frac{d}{2B+2}$. This means that $y_{n+1} \in (p - \frac{d}{2B+2}, p + \frac{d}{2B+2})$. Since $y_{n+1} < p + \frac{d}{2B+2} \leq p + \frac{d}{2} \leq p + (\frac{1}{2} - p) = \frac{1}{2}$, and $\rho_{n+3} = \frac{1}{2} - \frac{p}{2} - \min\left\{\frac{p}{4}, \frac{1}{4} - \frac{p}{2}\right\} = \max\left\{\frac{1}{2} - \frac{3p}{4}, \frac{1}{4}\right\} \geq \frac{1}{4}$, it follows that $n+3$ must be deployed such that its covering interval is to the right of the interval of $n+1$, namely $y_{n+3} - \rho_{n+3} \geq y_{n+1} + \rho_{n+1}$. Next, observe that $\rho_{n+2} + \rho_{n+3} = \frac{p-d/2}{2} + \frac{1-p-d/2}{2} = \frac{1-d}{2}$. Since $y_{n+1} + \rho_{n+1} > p - \frac{d}{2B+2} + \frac{d}{2B+2} = p \geq d$, it follows that sensor $n+2$ must be deployed such that its covering interval is to the left of the interval of $n+1$, namely $y_{n+2} + \rho_{n+2} \leq y_{n+1} - \rho_{n+1}$. Without loss of generality we assume that sensors $n+2$ and $n+3$ are adjacent to 0 and 1, respectively. Since all remaining radii are multiples of $\frac{d}{2B+2}$, it follows that $y_{n+1} = p$. Hence there is a subset $I \subseteq \{1, \ldots, n\}$ of sensors that covers the remaining uncovered area to the left of $p - \frac{d}{2B+2}$, while the rest of the sensors cover the remaining uncovered area to the right of $p + \frac{d}{2B+2}$. Thus

$$\sum_{i \in I} a_i = \frac{B+1}{d} \sum_{i \in I} 2\rho_i = \frac{B+1}{d}\left(\frac{d}{2} - \frac{d}{2B+2}\right) = \frac{1}{2}B \; .$$

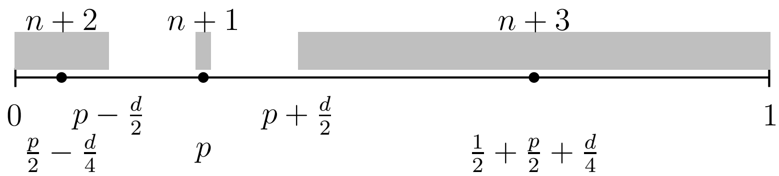


Figure 3.2: Depiction of the deployment and radii assignment of sensors $n+1$, $n+2$, and $n+3$.

Hence, $(a_1, \ldots, a_n) \in$ PARTITION. A contradiction. □

The following results are implied by Lemmas 3.6.1 and 3.6.2.

Corollary 3.6.3. *There is no polynomial time multiplicative approximation algorithm for* BCFR, *unless P=NP, for any* $a \in (0, \infty)$ *and* $\alpha \geq 1$, *even if* $x = p^n$, *where* $p \in (0, 1)$.

Corollary 3.6.4. *There is no polynomial time algorithm for* BCFR *that computes a solution within an additive factor* ε, *for some* $\varepsilon > 0$, *unless P=NP, for any* $a \in (0, \infty)$ *and* $\alpha \geq 1$, *even if* $x = p^n$, *where* $p \in (0, 1)$.

3.6.2 Variable Radii

For BCVR we show strong NP-hardness using a reduction from 3-PARTITION[2] that is based on the notion of *block*, which is a set of evenly spaced sensors

[2]A 3-PARTITION instance consists of a list $a_1, \ldots, a_n$ of $n = 3m$ positive integers such that $\frac{Q}{4} < a_i < \frac{Q}{2}$, for every i, and $\sum_i a_i = mQ$, and the goal is to decide whether the list can be partitioned into m triples all having the same sum Q. 3-PARTITION remains NP-hard even if Q is bounded above by a polynomial in n. In other words, the problem remains NP-hard even when representing the integers in the input instance in unary representation [23].

with relatively small batteries. A block battery cannot move much, but together the block batteries can cover a long interval, assuming they stay in their initial locations. Formally, a block $B = (z, \ell, b, \rho)$ is a set of ℓ sensors located at $z + (2i - 1)\rho$, for $i \in \{1, \dots, \ell\}$. The radius of each block sensor is ρ, and the battery power of each sensor is b. Typically, ρ would be small, while ℓ would be large.

Observation 3.6.5. *Let $B = (z, \ell, b, \rho)$ be a block. (i) B can cover the interval $[z, z + 2\ell\rho]$ for b/ρ^α time, and (ii) no block sensor can cover points outside $[z - \frac{b}{a}, z + 2\ell\rho + \frac{b}{a}]$.*

Proof. If a block battery remains in its initial position, it can stay alive for b/ρ^α time. Since the batteries are at distance 2ρ from their neighbors, the interval $[z, z + 2\ell\rho]$ is covered. A sensor can move at most b/a, hence the leftmost and rightmost point that can be reached by a block sensor are $z+\rho-b/a$ and $z+2\ell\rho-\rho+b/a$. Hence, no point outside $[z-b/a, z+2\ell\rho+b/a]$ can be covered by a block sensor. □

We are now ready to present the reduction.

Theorem 3.6.6. BCVR *is strongly NP-hard, for every $a \in (0, \infty)$ and $\alpha \geq 1$.*

Proof. Given an BCVR instance and T, we show that it is NP-hard to determine whether the instance can stay alive for T time.

Given a 3-Partition instance, we construct the following BCVR instance. Let $\delta = \frac{1}{(2m-1)Q}$ and $T = 2aQ[2(2m-1)Q]^\alpha$. There is a sensor for each input number: $x_i = 0$, and $b_i = T(a_i\delta/2)^\alpha + a$, for every $i \in \{1, \ldots, n\}$. We also add $m-1$ blocks: $B_j = ((2j-1)Q\delta, \lceil Q\delta/2\rho \rceil, T\rho^\alpha, \rho)$, for every j, where $\rho = \frac{\delta}{4} \cdot \frac{1}{[2(2m-1)Q]^\alpha}$.

The running time of the reduction is polynomial, since each block contains $O(m^\alpha Q^{\alpha+1})$ sensors, and there are $m-1$ blocks.

We show that if $(a_1, \ldots, a_n) \in$ 3-Partition, then there exists a solution with lifetime T. Since this instance belongs to 3-Partition, there is partition of $\{1, \ldots, n\}$ into m index subsets $I_1, \ldots, I_m$, such that $|I_j| = 3$ and $\sum_{i \in I_j} a_i = Q$, for any j. We set $r_i = \delta a_i/2$ for every $i \leq n$, and we deploy the sensors in I_j such that they cover $[2jQ\delta, (2j+1)Q\delta]$. Observe that the three sensors in I_j can cover the interval, since $\sum_{i \in I_j} 2r_i = \sum_{i \in I_j} a_i\delta = Q\delta$. Also, each such sensor uses at most a energy for deployment, and hence it has enough energy to stay alive for T time. Block sensors are not moved and their radii are set to ρ. Hence, block sensors can stay alive for T time. Furthermore, due to Observation 3.6.5, the sensors of block j can cover the interval $[(2j-1)Q\delta, (2j-1)Q\delta + 2\rho \lceil Q\delta/(2\rho) \rceil]$ during their lifetime. Observed that this interval contains $[(2j-1)Q\delta, 2jQ\delta]$. Hence $[0,1]$ can be covered for T time.

Now supposed that there is a solution with lifetime T. It follows that the block sensors radii cannot be larger than ρ. Hence, Observation 3.6.5 implies that the sensors of block j do not cover points outside

$$[(2j-1)Q\delta - T\rho^\alpha/a, (2j-1)Q\delta + 2\rho\lceil Q\delta/(2\rho)\rceil + T\rho^\alpha/a]\ .$$

Since

$$T\rho^\alpha/a = 2Q[2(2m-1)Q]^\alpha \cdot \tfrac{\delta^\alpha}{4^\alpha} \cdot [2(2m-1)Q]^{-2\alpha} \leq \tfrac{1}{2}Q\delta \cdot [2(2m-1)Q]^{-\alpha} \leq \tfrac{\delta}{8}$$

and

$$2\rho = \tfrac{2\delta}{4} \cdot \tfrac{1}{[2(2m-1)Q]^\alpha} \leq \tfrac{\delta}{8}\ ,$$

we have that the sensors of block j do not cover points outside $[(2j-1)Q\delta - \frac{\delta}{8}, 2jQ\delta + \frac{\delta}{4}]$. It follows that the interval $[2jQ\delta + \frac{\delta}{4}, (2j+1)Q\delta - \frac{\delta}{8}]$ must be covered by a subset of the first n sensors whose sum of radii is at least $(Q - \frac{3}{8})\delta$.

Since

$$T(a_i\delta/2)^\alpha = 2aQ[2(2m-1)Q]^\alpha a_i^\alpha [2(2m-1)Q]^{-\alpha} = 2aQa_i^\alpha\ ,$$

we have that the battery power of sensor i is

$$b_i = 2aQa_i^\alpha + a \leq 2aQa_i^\alpha \cdot \tfrac{2Q+1}{2Q} \leq T(a_i\delta/2)^\alpha \cdot (\tfrac{2Q+1}{2Q})^\alpha\ .$$

Hence, the radius that can be maintained by sensor i for T time is at most $\frac{a_i\delta}{2} \cdot \frac{2Q+1}{2Q}$. Since $a_i < Q/2$, this radius is smaller that δQ, and therefore the n sensors can be partitioned into m subsets $I_1, \ldots, I_m$, each covering an interval of length $(Q - \frac{3}{8})\delta$. We claim that $\sum_{i \in I_j} a_i \geq Q$ for every subset j. If this is not the case, then $\sum_{i \in I_j} a_i \leq Q - 1$, for some j. Hence,

$$\sum_{i \in I_j} \frac{a_i\delta}{2} \cdot \frac{2Q+1}{2Q} \leq (Q-1)\delta \cdot \frac{2Q+1}{2Q} = \frac{2Q^2-Q-1}{2Q} \cdot \delta < (Q - \tfrac{1}{2})\delta < (Q - \tfrac{3}{8})\delta \ .$$

Hence, we can partition $a_1, \ldots, a_n$ into m subsets each of sum at least Q, which means that $(a_1, \ldots, a_n) \in$ 3-Partition. $\square$

3.7 Discussion and Open Problems

We briefly discuss some research directions and open questions. We showed that BCVR is strongly NP-Hard. Finding an approximation algorithm or showing hardness of approximation remains open. In a natural extension model, sensors could be located anywhere in the plane and asked to cover a boundary or a circular boundary. In a more general model the sensors need to cover the plane or part of the plane where their initial locations could be anywhere. Another model which can be considered is the *duty cycling* model in which sensors are partitioned into shifts that cover the barrier. Bar-Noy *et al.* [6] considered this model for stationary sensors and $\alpha = 1$. Extending it to moving sensors and $\alpha > 1$ is an interesting research direction. Finally, in the most general covering problem with the goal of maximizing the coverage lifetime, sensors could change their locations and sensing ranges at any time. Coverage terminates when all the batteries are drained.

Chapter 4

Maximizing Communication Lifetime

This chapter is divided into the following sections. Section 4.1 formally defines the model and problems. The case where movement is free of cost is analyzed in Section 4.2. Section 4.3 gives some structural results. Section 4.4 considers the discrete version of the single deployment model. Section 4.5 extends the grid algorithms FPTASs when batteries are not too small. Section 4.6 considers the instance where relays are initially located at base stations. Finally, Section 4.7 concludes the chapter and gives open problems as well as directions for future research.

4.1 Preliminaries

A transmitter would like to transmit information to a receiver. To aid in the transmission of information, n mobile relays are distributed on the line

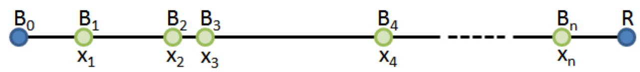


Figure 4.1: n relays on the line between the transmitter and the receiver.

of communication between the transmitter and the receiver (as in Fig. 4.1). Each relay has the task of passing along the information and thus maintaining the chain of communication. The distance between the transmitter and the receiver is denoted by D. We assume wlog that the transmitter is located at 0, while the receiver is located at D. The initial locations of the nodes are denoted by $x_0, x_1, \ldots, x_n$, where $x_0 = 0$ is the location of the transmitter.

The initial battery power of node $i \in \{0, 1, \ldots, n\}$, is denoted by B_i. Communicating costs energy and the energy required depends entirely on the distance over which the information must be transmitted. Following the model of Moscibroda *et al.* [39], a node transmitting data over a distance d invests $P(d) = d^\alpha$ energy per time unit, where $\alpha > 1$ is a constant. The cost of mobility, or *friction*, is proportional to the distance traveled, k, as in Wang *et al.* [43], i.e. $M(k) = ak$, where a is a constant.

We allow relays to occupy the same location. If two or more nodes are located at the same point, then one of them (say the node with the highest index) must transmit to the next live node which is not at the same location, while the other nodes do not consume any power.

Using the above definitions, we represent the network by the vector

$$N = \langle D, B_0, \ldots, B_n, x_1, \ldots, x_n, \alpha, a \rangle,$$

which gives the distance from transmitter to receiver, the initial battery powers and positions of the nodes, and the cost parameters, α and a. When $a = 0$ we say that there is *no friction*.

Single Deployment and Lifetime of First Death. Our first model corresponds to the case where relays are allowed to be set once. Relays are to be deployed at time 0 after which transmission may commence. The lifetime of the network is determined by its weakest link, namely by the lifetime of the relay whose battery is depleted first, thus breaking the chain of communication.

We have n relays initially located at $x_1, x_2, \ldots, x_n$ between 0 and D on the line. The relays are to be deployed to some locations $y_1, y_2, \ldots, y_n$. Moving relay i from location x_i to location y_i decreases relay i's battery by $a|x_i - y_i|$. Let B'_i be the new battery level of relay i after the movement, namely $B'_i = B_i - a|x_i - y_i|$. (Clearly $B'_0 = B_0$.) Let right_i denote the left-most node to the right of i, after the relays are deployed. A node $i \in \{0, \ldots, n\}$ must transmit to right_i. Let d_i be the transmission range of node i, for every $i \in \{0, \ldots, n\}$, namely $d_i = y_{\mathrm{right}_i} - y_i$, for every $i \in \{0, \ldots, n\}$, where $y_{\mathrm{right}_i} = D$ if the receiver is the rightmost node to the left of node i.

Once the relays arrive at locations $y = (y_1, y_2, \ldots, y_n)$, they may begin transmitting and must transmit from their respective locations for the duration of their lifetimes. The lifetime of relay i is defined as $L_i(y_i) = B'_i/d_i^\alpha$. The lifetime of the system given the deployment to y is the time in which the first relay dies, and it is defined as

$$L_F(N, y) = \min_{0 \le i \le n} L_i(y_i) \ . \tag{4.1}$$

We shall refer to this lifetime as *Lifetime of First Death.* This notion of network lifetime was considered by El-Moukaddem *et al.* [22] for a max data mobile relay configuration.

The MAXIMUM LIFETIME OF FIRST DEATH problem (MAXFD) is the problem of finding a deployment that maximizes the lifetime of first death.

In this paper we focus on the *no swapping* case, where the relays are to be deployed to some locations $y_1, y_2, \ldots, y_n$ preserving the initial order between relays, i.e. $x_i \le x_j$ implies $y_i \le y_j$, for every $i \ne j$. In this case, $d_i = y_{i+1} - y_i$ for $i \in \{0, \ldots, n-1\}$ and $d_n = D - y_n$. This problem can be represented by the following nonlinear (and non-convex) program:

$$\begin{aligned} &\max_{y_1,\cdots,y_n} \min \left\{ \tfrac{B_0}{y_1^\alpha}, \tfrac{B_1 - a|x_1 - y_1|}{(y_2 - y_1)^\alpha}, \ldots, \tfrac{B_n - a|x_n - y_n|}{(D - y_n)^\alpha} \right\} \\ &\text{s.t.} \ \ 0 \le y_1 \le \cdots \le y_n \le D \end{aligned} \tag{4.2}$$

The reason for focusing on this case is threefold. First, there are applications in which swapping is disallowed. For example, Kansal et al. [30] suggest using relays on a track. Second, in some scenarios adding this no swapping restriction gives additional structure that may be used for solving the problem without affecting the solution space. For instance, when n identical relays are initially located at the same point. Finally, we show (in Section 4.2) that swapping is unnecessary in the non-friction case.

Another nice implication of the no-swapping assumption is multiple deployments do not help. More specifically, we can show that any solution where relays redeploy after time 0 can be replaced by a single deployment solution whose lifetime is not worse, see corollary 4.3.2.

Furthermore, we assume that all relays must participate in communication. This may have a drawback in that if any relay has a very small initial battery and a sizable transmission range, the lifetime of first death would be short. Note that if $B_i \geq a \max\{x_i, D - x_i\}$, for all i, then a relay may effectively deactivate by moving arbitrarily close to the next relay. Thus, deactivation is encompassed by the model given if $B_i \geq a \max\{x_i, D - x_i\}$, for all i. We also note that our FPTASs for MaxFD are based on a stronger assumption (see Section 4.5).

Multiple Deployment and Transmission Lifetime. In our second model we assume relays can be deployed multiple times and can readjust their transmission ranges after a deployment or after the death of a relay. In this model, we wish to maximize the length of time the transmitter can communicate with the receiver. We call such a lifetime the *Transmission Lifetime*.

A solution, or *solution path*, is defined as $\mathcal{P} := \{x_1(t), \ldots, x_n(t)\}_{t=0}^{\infty}$, where $x_i(t)$ is the location of relay i at time t for $i \in \{1, \ldots, n\}$. Let $\text{left}_i(t)$ and $\text{right}_i(t)$ denote the right-most node to the left of node i and the left-most node to the right of i at time t. A node $i \in \{0, \ldots, n\}$ must transmit to $\text{right}_i(t)$. If relay i dies at some time t then the node $\text{left}_i(t)$ increases its transmission range to transmit to $\text{right}_i(t)$. The transmitting range of a live node i at time t, denoted by $d_i(t)$, is the distance between that node and $\text{right}_i(t)$, i.e., $d_i(t) = x_{\text{right}_i(t)}(t) - x_i(t)$. The remaining battery power of node i at time t is denoted by $B_i(t)$.

Let $X_i(t)$ be the total distance traveled by node i up to time t (note $X_0(t) = 0 \ \forall t$). Using this notation, the lifetime of node i under solution path $\mathcal{P}$, $L_i(\mathcal{P})$, satisfies:

$$B_i = aX_i(L_i(\mathcal{P})) + \int_0^{L_i(\mathcal{P})} d_i(t)^\alpha dt \tag{4.3}$$

A solution path $\mathcal{P}$ is feasible if (i) $x_i(0) = x_i$ for every $i \in \{1, \ldots, n\}$, and

(ii) $x_i(t) = x_i(L_i(\mathcal{P}))$, for every $i \in \{1, ..., n\}$ and $t \geq L_i(\mathcal{P})$.

We define the *Transmission Lifetime* for a solution path $\mathcal{P}$ and network N, denoted by $L_T(N, \mathcal{P})$, to be the length of time the transmitter can send data to the receiver in a solution path $\mathcal{P}$ for a given network N. This is equivalent to the lifetime of the transmitter under a solution path $\mathcal{P}$, $L_0(\mathcal{P})$. Thus:

$$L_T(N, \mathcal{P}) = L_0(\mathcal{P}) \ . \tag{4.4}$$

The Maximum Transmission Lifetime problem (abbreviated MaxTL) is the problem of finding a solution path $\mathcal{P}$ that maximizes the transmission lifetime of a given network N.

Observe that the maximum transmission lifetime is never smaller than the maximum lifetime of first death, namely

$$\max_y L_F(N, y) \leq \max_{\mathcal{P}} L_T(N, \mathcal{P})$$

for every network N, since $L_F(N, y) \leq L_T(N, \mathcal{P}_y)$, for every deployment y, where $\mathcal{P}_y$ is the solution path that corresponds to y, i.e., $\mathcal{P}_y$ satisfies $x_i(t) = y_i$ for every relay i.

4.2 MaxTL and MaxFD without Friction

In this section we consider the no friction case, namely the case where $a = 0$. Goldenberg *et al.* [25] consider the case where there is no friction and all nodes have equal battery power. They show that the energy cost function $P(d(t))$ is a non-decreasing convex function, and that the optimal positions of the relay nodes must lie entirely on the line between the source and the destination and must be evenly spaced along the line.

We can extend this result to the case of non-equal battery powers. We show that the lifetime of the network is optimized when we choose the transmission ranges $d_i(t)$'s to be such that the lifetimes of all nodes are equal, i.e. $L_i = L_j$, $\forall i, j \in \{0, \ldots, n\}$ and fixed for the lifetime of the network: $d_i(t) = d_i$ for the lifetime (both transmission and first death) of the network. In this case, relays only need to move once to their optimal locations and transmission lifetime is equal to lifetime of first death.

Lemma 4.2.1. *The transmission lifetime and lifetime of first death of a network with n relays where there is no friction is maximized when $L_i = L_j$, for every $i, j \in \{0, \ldots, n\}$.*

Proof. Let T be the transmission lifetime of an optimal solution. Define $\overline{d_i}$

to be the time average of $d_i(t)$ from 0 to T for each $i \in \{0, \ldots, n\}$:

$$\overline{d_i} = \frac{1}{T} \int_0^T d_i(t)\, dt \ .$$

We first show that $\sum_{i=0}^{n} \overline{d_i} = D$:

$$\begin{aligned} \sum_{i=0}^{n} \overline{d_i} &= \sum_{i=0}^{n} \frac{1}{T} \int_0^T d_i(t)\, dt \\ &= \frac{1}{T} \int_0^T \sum_{i=0}^{n} d_i(t)\, dt = \frac{1}{T} \int_0^T D\, dt = D \ . \end{aligned}$$

A feasible placement in which i's range is $\overline{d_i}$ can be obtained by placing i at $\sum_{j=0}^{i-1} \overline{d_j}$, for every i. We now show that for arbitrary i:

$$\int_0^T (\overline{d_i})^\alpha\, dt \leq \int_0^T d_i(t)^\alpha\, dt$$

This follows with the use of Jensen's Inequality:

$$\begin{aligned} \int_0^T (\overline{d_i})^\alpha\, dt &= T \cdot (\overline{d_i})^\alpha \leq T \cdot \overline{d_i(t)^\alpha} \\ &= T \cdot \frac{1}{T} \int_0^T d_i(t)^\alpha\, dt = \int_0^T d_i(t)^\alpha\, dt \ . \end{aligned}$$

It thus suffices to consider the solutions in which the transmission ranges $d_i(t)$, $i \in \{0, \ldots, n\}$ are fixed for the duration of lifetime. Of all such solutions, the one in which all node lifetimes are equivalent, $L_i = L_j$, $\forall i, j \in \{0, \ldots, n\}$, maximizes the lifetime of the network.

Assume this is not the case. Consider the time M when the transmitter dies. Note that if a relay dies before time M the solution does not have fixed transmission ranges. Consider the leftmost relay which is still alive at time M. WLOG, assume it is relay k. Shift the first k relays to the left by an amount $\delta > 0$, where $\delta \leq \min\{d_0(M), \epsilon\}$ and $\epsilon = \sqrt[\alpha]{\frac{B_k}{M}} - d_k$, i.e., ϵ satisfies $B_k/(d_k + \epsilon)^\alpha = M$. The shift decreases d_0, while leaving $d_1, d_2, \ldots, d_{k-1}$ unchanged. Consequently the lifetime of this solution is greater than the lifetime of the optimal, a contradiction.

If $L_i = L_j$, $\forall i, j \in \{0, \ldots, n\}$, lifetime of first death equals transmission lifetime. Since lifetime of first death is never larger than transmission lifetime, it follows that lifetime of first death is maximized as well. $\square$

Theorem 4.2.2. *If $a = 0$, an optimal solution for* MaxTL *and* MaxFD *is obtained by placing relay i at $\sum_{j=0}^{i-1} d_j$, where*

$$d_i = D \cdot \frac{\sqrt[\alpha]{B_i}}{\sum_{j=0}^{n} \sqrt[\alpha]{B_j}} \tag{4.5}$$

for every $i \in \{1, \ldots, n\}$. The corresponding lifetime is $D^{-\alpha}(\sum_{j=0}^{n} \sqrt[\alpha]{B_j})^\alpha$.

Proof. Consider relay i. Due to Lemma 4.2.1 we know that $L_i = L_j$ for every $j \in \{0, \ldots, n\}$. It follows that $d_j = \sqrt[\alpha]{B_j/B_i} \cdot d_i$, for every $j \in \{0, \ldots, n\}$. Since $\sum_{j=0}^{n} d_j = D$, we have that

$$\frac{d_i}{\sqrt[\alpha]{B_i}} \sum_{j=0}^{n} \sqrt[\alpha]{B_j} = D \ ,$$

which gives the result. $\square$

Corollary 4.2.3. *If $a = 0$, there exists a single deployment and no swapping optimal solution for* MaxTL.

4.3 MaxTL without Swapping

In this section, we consider MaxTL with friction. We provide some structure for this case by giving necessary conditions for any optimal solution.

First, we note that Theorem 4.2.2 does not hold in the non-zero friction case. Consider the case where we have one relay located at 0.25 with $B_0 = B_1 = 1$ and $D = 1$. Also, assume that $a = 4$. This means that going from 0.25 to 0.5 depletes the relay. However, located at [0.25,0.5) the relay dies before the transmitter.

We prove that in an optimal solution, relays do not need to move unless some relay dies.

Lemma 4.3.1. *Given a* MaxTL *instance, any solution where a relay moves at a time that does not correspond to the death of another relay can be replaced by a stationary solution with at least as large a lifetime.*

Proof. Assume that in an optimal solution no relay dies in the time interval (t_0, t_2). Furthermore, assume that in this optimal solution, relays move at an instance t_1, where $t_0 < t_1 < t_2$ and are stationary otherwise.

Consider the time average location of relay i from t_0 to t_2, which has moved from location $x_i(t_0)$ to $x_i(t_1)$ at time instance t_1:

$$\overline{x}_i = \frac{x_i(t_0) \cdot (t_1 - t_0) + x_i(t_1) \cdot (t_2 - t_1)}{t_2 - t_0} .$$

We claim that a solution that places relay i at location $\overline{x}_i$ at time t_0 and does not move i before t_2 has lifetime at least as large as the original solution. First, notice that $\overline{x}_i \leq \overline{x}_{i+1}$ for every i, namely relay ordering is maintained by the new solution. Also, since location $\overline{x}_i$, the time average, is between locations $x_i(t_0)$ and $x_i(t_1)$, the cost of movement from $x_i(t_0)$ to $\overline{x}_i$ to $x_i(t_2)$ is at most the cost of the original movement from $x_i(t_0)$ to $x_i(t_1)$ to $x_i(t_2)$.

Let us now consider the cost of transmission if relay i is placed at location $\overline{x}_i$ at time t_0. Let $d_i(t_0) = x_{i+1}(t_0) - x_i(t_0)$ and $d_i(t_1) = x_{i+1}(t_1) - x_i(t_1)$. Define $\overline{d}_i$ to be the time average from t_0 to t_2 of the transmitting distance between node i and node $i + 1$:

$$\overline{d}_i = \frac{d_i(t_0) \cdot (t_1 - t_0) + d_i(t_1) \cdot (t_2 - t_1)}{t_2 - t_0} .$$

Observe that

$$\begin{aligned}
\overline{d}_i =& \frac{d_i(t_0)\cdot(t_1-t_0)+d_i(t_1)\cdot(t_2-t_1)}{t_2-t_0} \\
=& \frac{(x_{i+1}(t_0)-x_i(t_0))(t_1-t_0)}{t_2-t_0} \\
&+ \frac{(x_{i+1}(t_1)-x_i(t_1))(t_2-t_1)}{t_2-t_0} \\
=& \frac{x_{i+1}(t_0)\cdot(t_1-t_0)+x_{i+1}(t_1)\cdot(t_2-t_1)}{t_2-t_0} \\
&- \frac{x_i(t_0)\cdot(t_1-t_0)+x_i(t_1)\cdot(t_2-t_1)}{t_2-t_0} \\
=& \overline{x}_{i+1}-\overline{x}_i
\end{aligned}$$

so that it is feasible to place relays i and $i+1$ at their time-averaged locations and obtain the time-averaged transmission distance between them.

By Jensen's inequality,

$$(\overline{d}_i)^\alpha \leq \overline{d_i^\alpha} = \frac{(t_1-t_0)d_i(t_0)^\alpha+(t_2-t_1)d_i(t_1)^\alpha}{t_2-t_0}$$

and thus,

$$(t_2-t_0)(\overline{d}_i)^\alpha \leq \overline{d_i^\alpha} = (t_1-t_0)d_i(t_0)^\alpha+(t_2-t_1)d_i(t_1)^\alpha.$$

That is, the amount of battery depleted for relay i in the time-average case is at most the amount of battery depleted in the case when the relay moves at time t_1. We thus have the desired result. $\square$

Corollary 4.3.2. *One deployment suffices when maximizing the lifetime of first death.*

It also holds that in any optimal solution the transmitter is one of the last to die.

Lemma 4.3.3. *Given a* MaxTL *instance, any solution in which the transmitter is not among the last to die is suboptimal.*

Proof. Assume the transmitter dies at time M, but relay k is the leftmost relay which is still alive in the optimal solution, i.e. relays $1, \ldots, k-1$ die prior to or at time M in the optimal solution. Assume relays $i_1, i_2, \ldots, i_l$ are among the first $k-1$ relays and die at time M with the transmitter.

Move relay k an amount ϕ_k to the left at time $M-\tau$, where τ is small enough such that no relay dies in the time interval $[M-\tau, M)$. Since no relay dies, we can assume by Lemma 4.3.1 that the relays are stationary in this time interval. Let ϕ_k be an amount which precludes relay k from dying before time M and is less than $\min\{d_0(M-\tau), d_{i_1}(M-\tau), \ldots, d_{i_l}(M-\tau)\}$. Sequentially move relays $j = i_l, i_{l-1}, \ldots, i_1$ at time $M-\tau$, an amount ϕ_j to the left where $\phi_j < \phi_{j+1}$ and is such that

$$\frac{B_j - a\phi_j}{(d_j(M-\tau) - (\phi_{j+1} - \phi_j))^\alpha} = \frac{B_j}{d_j(M-\tau)^\alpha}.$$

Note that such a movement is always feasible: consider the function

$$J(x) = \frac{B_j - ax}{(d_j(M-\tau) - (\phi_{j+1} - x))^\alpha}.$$

We have that $J(0) = B_j/(d_j(M-\tau)-\phi_{j+1})^\alpha > B_j/d_j(M-\tau)^\alpha$. Furthermore, $J(x)$ is a strictly decreasing function on the interval $[0, \phi_{j+1}]$ and

$$J(\phi_{j+1}) = \frac{B_j - a\phi_{j+1}}{d_j(M-\tau)^\alpha} < \frac{B_j}{d_j(M-\tau)^\alpha}.$$

Since $J(x)$ is continuous, there is some point $\phi_j \in [0, \phi_{j+1}]$ for which

$$\frac{B_j - a\phi_j}{(d_j(M-\tau) - (\phi_{j+1} - \phi_j))^\alpha} = \frac{B_j}{d_j(M-\tau)^\alpha}.$$

Note that since

$$\phi_k < \min\{d_0(M-\tau), d_{i_1}(M-\tau), \ldots, d_{i_l}(M-\tau)\}$$

and $\phi_k > \phi_{i_l} > \cdots > \phi_{i_1}$, relays can never swap order with this shift. This shift of the relays decreases $d_0(M-\tau)$, resulting in a solution path with a longer transmission lifetime, a contradiction.

$\square$

4.4 MaxFD on Grid Points

In this section we consider a discrete version of MaxFD in which relays are deployed on grid points. More specifically, we assume that the final locations $y_1, \ldots, y_n$ of the relays must be one of the points $\{jD/m : 0 \leq j \leq m\}$, where m is a pre-determined integer. That is, we partition the interval $[0, D]$

into m sub-intervals each of length $\sigma = D/m$ and restrict the final locations from being in the interior of any sub-interval.

We provide two approaches to solve the problem: a dynamic programming algorithm and a parametric search algorithm, the running times are $O(nm^2)$ and $O(nm^2 \log(nm))$, respectively.

We note that our algorithms for the discrete version of MaxFD can be extended to deal with relay deactivations. Also, note that deactivations may be ignored if $B_i \geq a \max\{x_i, D - x_i\}$, for every i. (Recall that in this case a relay may effectively deactivate by moving arbitrarily close to the next relay.)

4.4.1 Dynamic Programming

We present a dynamic programming algorithm that solves discrete MaxFD called *Lifetime-DP*, shown as Algorithm 3. The idea behind this algorithm is to try solving all possible instances for covering the prefix segment $[0, jD/m]$ for any $j \in \{0, \dots, m\}$ using the i relays closest to the transmitter for $i \in \{0, \dots, n\}$.

For $j \in \{0, \dots, m\}$ and $i \in \{0, \dots, n\}$, let $f(i, j)$ be the lifetime of first death of the solution with the i relays that are closest to the transmitter that covers $[0, jD/m]$ (this is equivalent to moving the receiver to point jD/m). The desired final output will be $f(n, m)$. Initially, for $i = 0$ and any j, $f(i, j)$

is the time it takes the transmitter to die when it transmits a distance jD/m. Also, let $y(i,j) = (y_1, \ldots, y_i)$ be the vector of optimal positions assigned to the i relays when transmitting to j. Namely, $y(i,j)$ corresponds to $f(i,j)$.

Let $L_F(i,h,j)$ be the lifetime of first death of relay i after moving to grid point h and transmitting to gridpoint $j \geq h$. That is,

$$L_F(i,h,j) = \begin{cases} \frac{B_i - a\left|x_i - \frac{hD}{m}\right|}{\left(\frac{jD}{m} - \frac{hD}{m}\right)^\alpha} & \left|x_i - \frac{hD}{m}\right| \leq \frac{B_i}{a}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.6}$$

$f(i,j)$ can be computed using the following recursive rule:

$$f(i,j) = \begin{cases} \max\limits_{0 \leq h \leq j} \min\{f(i-1,h), L_F(i,h,j)\} & i,j > 0, \\ \infty & j = 0, \\ \frac{B_0}{(jD/m)^\alpha} & i = 0, j > 0. \end{cases}$$

That is, look for the lifetime maximizing position for the ith relay among the first $j+1$ possible positions. When a position is examined for the ith relay, use the optimal locations for the first $i-1$ relays transmitting to such a position to find the lifetime for that configuration. Take the maximum lifetime of all position evaluations to be $f(i,j)$.

Theorem 4.4.1. *Algorithm Lifetime-DP finds an optimal solution for* MaxFD *on grid points in $O(nm^2)$ time.*

Proof. We prove by induction on i that $f(i,j)$, for every j, is the maximum lifetime of first death of the first i relays when the receiver is located at jD/m.

Algorithm 3 : *Lifetime-DP*

```
for all j = 0 to m do f(0, j) ← B_0 / (jD/m)^α  {f(0, 0) ← ∞}
for all i = 1 to n do
   for all j = 0 to m do
      for all h = 0 to j do
         Compute L_F(i, h, j)
      end for
   end for
end for
for all i = 1 to n do
   for all j = 1 to m do
      f(i, j) ← 0
      for all h = 0 to j do
         if  min{f(i − 1, h), L_F(i, h, j)} > f(i, j) then
            f(i, j) ← min{f(i − 1, h), L_F(i, h, j)}
            y(i, j) ← (y_1(i − 1, h), . . . , y_{i−1}(i − 1, h), hD/m)
         end if
      end for
   end for
end for
```

Note that the claim is true for $i = 0$. Assume the claim is true for all $i - 1$. We would like to show that it is true for i. In order to compute $f(i, j)$ the algorithm places the ith relay at grid points $h = 0, \ldots, j$ and considers the lifetime of this placement with the maximal solution of $i - 1$ relays transmitting to h, $\boldsymbol{y}(i-1, h)$. By the induction hypothesis, $\boldsymbol{y}(i-1, h)$, for $0 \leq h \leq j$, are the positions of maximum lifetime of the first $i - 1$ relays transmitting to h. Hence, the best lifetime of first death when i is placed at hD/m and transmits to j is the minimum between $f(i - 1, h)$ and the lifetime of i, namely $L_F(i, h, j)$. Since the dynamic programming solution takes the maximum over all h, $f(i,j)$ is optimal for MaxFD on grid points, for every j.

The computation of $L_F(i,h,j)$, for any i, h, and j, is given by equation 4.6 and takes $O(1)$ time to compute. There are $O(nm^2)$ values to compute and hence the total time to compute $L_F(i,h,j)$ for all i,h,j is $O(nm^2)$. The running time of computing $f(n,m)$ is $O(nm^2)$ using the precomputed values in L_F. Hence, the total running time is $O(nm^2)$. $\square$

4.4.2 Parametric Search

In this subsection we provide a second algorithm for discrete MaxFD that is based on parametric search.

We first provide a polynomial time algorithm that decides whether a given candidate lifetime is achievable. The algorithm is called *Lifetime Check*. Intuitively, the algorithm works as follows. Initially set $y_{n+1} = D$. Going from $i = n$ to 1 we try to move relay i to the leftmost position for which relay i has lifetime T'. If we succeed for all i, we check whether the transmitter has enough power to transmit to relay 1 at least until T'.

Lemma 4.4.2. *Let T be the optimal lifetime of first death. Given $T' \geq 0$, Algorithm Lifetime Check determines whether $T' \leq T$ in $O(nm)$ time, and it computes a corresponding solution if $T' \leq T$.*

Algorithm 4 : *Lifetime Check*

for all $i = 1$ to n **do**
  $\ell(i) \leftarrow \min\{j \geq 0 : a(x_i - jD/m) \leq B_i\}$
  $\ell'(i) \leftarrow \max\{\ell(i), \ell'(i-1)\}$ $\{\ell'(0) = 0\}$
  $r(i) \leftarrow \max\{j \leq m : a(jD/m - x_i) \leq B_i\}$
end for
$y_{n+1} \leftarrow D$
for all $i = n$ to 1 **do**
  $P_i \leftarrow \left\{ j\frac{D}{m} \leq y_{i+1} : \begin{array}{c} j \in [\ell'(i), r(i)] \\ a|x_i - j\frac{D}{m}| + T'(y_{i+1} - j\frac{D}{m})^\alpha \leq B_i \end{array} \right\}$
  if $P_i = \emptyset$ **then**
    return NO
  else
    $y_i \leftarrow \min P_i$
  end if
end for
if $T' y_1^\alpha > B_0$ **then**
  return NO
end if
return YES

Proof. First, assume that the algorithm returns YES. In this case $y_i \leq y_{i+1}$, for every i, by construction. Also, $y_i \in [\ell(i), r(i)]$, which means that relay i has the enough energy to move to y_i. Furthermore, due to the definition of P_i, y_i is able to maintain the connection to y_{i+1} for T' time. Hence, y is a deployment that corresponds to T', which means that $T' \leq T$.

Now, suppose that $T' \leq T$, and let z be a deployment that correspond to T. We show that the algorithm computes a solution y whose lifetime is at least T'. We show by induction on i that (i) relays $i, \ldots, n$ can connect y_i to D for at least T' time, and (ii) $y_i \leq z_i$. In the base case, notice that $z_n \in P_n$, and therefore relay n can connect y_n to D for T' time, and also $y_n \leq z_n$. For the induction step, assume that relays $i+1, \ldots, n$ can maintain a connection from y_{i+1} to D for at least T' time, and that $y_{i+1} \leq z_{i+1}$. If $i \neq 0$, notice that since $y_{i+1} \geq \ell'(i+1) \geq \ell'(i)$, we have that $\min\{y_{i+1}, z_i\} \in P_i$, and therefore $P_i \neq \emptyset$, and $y_i \leq z_i$. Finally, relay 0 can maintain the connection to y_1, since $y_1 \leq z_1$ due to the induction hypothesis. It follows that the algorithm returns YES.

Computing ℓ, ℓ', and r can be done in $O(n)$ time. There are n iterations of the main loop, and P_i can be computed in $O(m)$. Hence, the overall running time is $O(nm)$. □

We use Algorithm *Lifetime Check* to devise an algorithm for discrete MaxFD.

Theorem 4.4.3. *There exists an algorithm that finds an optimal solution for* MaxFD *on grid points in $O(nm^2 \log(nm))$ time.*

Proof. There are $O(nm^2)$ lifetime of first death candidates: each relay has $m+1$ possible locations and $m+1$ possible ranges. We can sort these candidates in $O(nm^2 \log(nm))$ time, and then find the best lifetime using binary search with Algorithm *Lifetime Check* in $O(\log(nm))$ iterations. The total running time is $O(nm^2 \log(nm))$. □

We note that the running time of the above binary search algorithm is slower that the running time of the dynamic programming algorithm, but it will become useful in the next section.

4.5 MaxFD with Large Batteries

In this section we show that the algorithms that were given in Section 4.4 may be used to obtain FPTASs for the special case of MaxFD in which $B_i \geq aD(1+\frac{1}{n})$, for every i. Our approach is based on the following argument. Given any $\varepsilon > 0$, we show that if $B_i \geq aD(1+\frac{1}{n})$, for every i, there exists a grid density for which the discrete optimal lifetime of first death is within a factor of $(1+\varepsilon)$ from the optimal lifetime of first death.

We start the section by exploring the connection between discrete MaxFD and non-discrete MaxFD.

4.5.1 Discrete MaxFD vs. MaxFD

Let OPT_F be the optimal lifetime of first death, and let OPT_F^m be the optimal lifetime of first death for the discrete version with m grid points.

We first show that OPT_F and OPT_F^m may be far apart even when m is very large. Let m be an odd integer. Consider the following instance with three relays, $a > 0$ and $D = 1$. The relay locations are given by $x = (\frac{1}{2}-\frac{\sigma}{2}, \frac{1}{2}+\frac{\sigma}{2}, 1)$, and relay batteries are $B_1 = B_2 = \frac{a\sigma}{2}$, and $B_0 = B_3 = B$, where B is very large. The optimal deployment is $y_1 = y_2 = y_3 = \frac{1}{2}$, and in this case OPT_F is close to $\frac{B}{0.5^\alpha}$, while OPT_F^m is at most $\frac{a}{2\sigma^{\alpha-1}}$.

Such scenarios may be avoided if the deployment at grid points does not deplete the batteries. In the next lemma we make an assumption that ensures this.

Lemma 4.5.1. *Let $\varepsilon \in (0,1)$. If $B_i \geq aD(1+\frac{1}{n})$, for every i, and $m = \lceil (n+1)^2/\varepsilon \rceil$, then*

$$\text{OPT}_F^m > \frac{\text{OPT}_F}{1+2(\alpha+1)\varepsilon} \ .$$

Proof. Let $(y_1, \ldots, y_n)$ be an optimal deployment, namely a deployment whose lifetime of first death is OPT_F. Also, let $d_i = y_{i+1} - y_i$, for every $i \in \{0, \ldots, n\}$. Observe that there must exists at least one relay i for which $d_i \geq \frac{(n+1)\sigma}{\varepsilon}$, since otherwise

$$D = \sum_{i=0}^{n} d_i < (n+1)\frac{(n+1)\sigma}{\varepsilon} = \frac{(n+1)^2 D}{\varepsilon m} \leq D \ .$$

Let ℓ be such a relay.

Let $d'_i = \sigma \cdot \left\lfloor \frac{d_i}{\sigma} \right\rfloor$, for every $i \neq \ell$, and let $d'_\ell = D - \sum_{i \neq \ell} d'_i$. Observe that $d_i - \sigma < d'_i \leq d_i$, for every $i \neq \ell$, and that

$$d'_\ell < d_\ell + (n+1)\sigma \leq d_\ell + \varepsilon d_\ell = (1+\varepsilon) d_\ell \ .$$

We describe a discrete deployment y' using the new distances between relays: $y'_i = \sum_{k<i} d'_k$, for every i. Observe that

$$|y'_i - y_i| \leq n\sigma = \frac{nD}{m} \leq \frac{\varepsilon nD}{(n+1)^2} < \frac{\varepsilon D}{n} \ ,$$

for every sensor i.

Let R and R' are the remaining battery powers after moving to y and y', respectively. Since $R_i \geq \frac{aD}{n}$, for every i, it follows that

$$R'_i \geq R_i - a|y'_i - y_i| > R_i - \frac{\varepsilon aD}{n} \geq R_i - \varepsilon R_i = (1-\varepsilon)R_i \ ,$$

Putting it all together we get that

$$L'_i = \frac{R'_i}{(d'_i)^\alpha} \geq \frac{(1-\varepsilon)R_i}{(1+\varepsilon)^\alpha d_i^\alpha} > \frac{1}{(1+\varepsilon)^{\alpha+1}} \cdot L_i \geq \frac{1}{1+2(\alpha+1)\varepsilon} \cdot L_i \ ,$$

where the last inequality follows from $(1+\varepsilon)^{\alpha+1} \leq e^{(\alpha+1)\varepsilon} \leq 1 + 2(\alpha+1)\varepsilon$, for $(\alpha+1)\varepsilon \leq 1$. □

4.5.2 FPTASs

We start with the dynamic programming algorithm. Lemma 4.5.1 implies that Algorithm *Lifetime-DP* can be used as an FPTAS for the special case of MaxFD in which $B_i \geq aD(1+\frac{1}{n})$, for every i. Since $m = O(n^2/\varepsilon)$, the running time in this case is $O(nm^2) = O(n^5/\varepsilon^2)$.

Next, we move to the parametric search algorithm. We consider a variation of the algorithm that was given in Section 4.4. Instead of considering lifetime candidates, we perform binary search on an interval. Let T_a denote the lifetime for movement cost a. Observe that $T_a \geq T_b$ if $a \leq b$. It follows that $T_a \in [T_\infty, T_0]$. By Theorem 4.2.2 we have that

$$T_0 = \frac{(\sum_{j=0}^{n} \sqrt[\alpha]{B_j})^\alpha}{D^\alpha} \leq \frac{(n \sqrt[\alpha]{B_{\max}})^\alpha}{D^\alpha} = \frac{n^\alpha B_{\max}}{D^\alpha} ,$$

where $B_{\max} = \max_i B_i$. On the other hand,

$$T_\infty = \min_i \frac{B_i}{(x_{i+1} - x_i)^\alpha} \geq \frac{B_{\min}}{D^\alpha} ,$$

where $B_{\min} = \min_i B_i$. Hence if conduct a binary search until we reach an interval of length at most ε, the number of iterations of the binary search is bounded by

$$\log\left(\tfrac{T_0}{\varepsilon T_\infty}\right) \leq \log\left(\tfrac{n^\alpha B_{\max}}{\varepsilon B_{\min}}\right) = \alpha \log n + \log \tfrac{B_{\max}}{B_{\min}} + \log \tfrac{1}{\varepsilon} .$$

Since the running time of Algorithm *Lifetime Check* is $O(nm)$, it follows that the total running time is $O(\frac{n^2}{\varepsilon} \cdot (\log n + \log(\frac{1}{\varepsilon})))$. This is faster than the running time of the dynamic programming algorithm, which is $O(n^5/\varepsilon^2)$.

4.6 MaxFD with Deployment from Base Stations

In this section we consider the MaxFD scenario where relays are initially located at the base stations i.e. located at either 0 or D ($x \in \{0, D\}^n$). We show there exist an optimal solution for this instance for discrete MaxFD and an FPTAS as long as $B_i > aD(1+\frac{1}{n})$. We do so by showing that we can order the relays in a particular way (which is a *dominant* order) and then we can use *Lifetime-DP* and *Lifetime Check with Binary Search* on this order for solution.

We first consider the MaxFD instance where all relays are initially located with the transmitter, i.e. $x_i = 0$ for all i. We seek to determine an order in which to deploy the relays for which the lifetime of first death is at least as large when compared to any other ordering.

Let T be an achievable lifetime. We show that there exists a deployment y with lifetime T such that relays are deployed according to *battery order*, namely $B_i \leq B_j$ if and only if $y_i \leq y_j$. Battery order deploys larger battery relays to farther locations than smaller battery relays.

For the proof we need the result of Lemma 2.3.4 which is restated below.

Lemma 2.3.4. *Let* $\eta_1, \eta_2, \gamma_1, \gamma_2 \geq 0$ *such that (i)* $\gamma_1 < \eta_1 \leq \eta_2$*, and (ii)* $\eta_1 + \eta_2 \geq \gamma_1 + \gamma_2$*. Also let* $\alpha \geq 1$*. Then,* $\sqrt[\alpha]{\eta_1} + \sqrt[\alpha]{\eta_2} \geq \sqrt[\alpha]{\gamma_1} + \sqrt[\alpha]{\gamma_2}$*.*

Lemma 4.6.1. *Assume* $x_i = 0$ *for all* i *and let* $p \in (0, D]$*. Suppose that there exists a deployment that transmits from* 0 *to* p *for* T *time. Then, there exists a deployment that transmits from* 0 *to* p *for* T *time that satisfies battery ordering.*

Proof. Given a solution that transmits from $[0, p]$ with lifetime of first death T, a pair of relays is said to violate battery ordering if $B_i < B_j$ and $y_i > y_j$. Let y be a solution with lifetime of first death T that minimizes battery ordering violations. If there are no violations, then we are done. Otherwise, we show that the number of violations can be decreased.

If y has ordering violations, then there must exist at least one violation due to a pair of adjacent relays. Let i, j with $B_i < B_j$ but $y_j < y_i$ be such relays. Consider the interval $[y_j, y_i + d_i]$ in which relays i and j transmit. Define β_k to be the remaining battery of relay k after moving to location y_k, that is, $\beta_k = B_k - ay_k$. Let $K = \min\{L_i, L_j\}$ and note that $K \geq T$. Consider a solution y' which is equivalent to y, except for relay i located at $y'_i = y_j$ and relay j located at $y'_j = \min\{y_i, y'_i + d'_i\}$, where d'_i is such that $\beta'_i / d_i^\alpha = K$, i.e. $d'_i = \sqrt[\alpha]{\beta'_i / K}$. Note that no other sensors lifetime is affected by this reassignment. It follows that relay i remains alive for at least K time. If $y'_j = y_i$ then sensor j remains alive for at least K time as sensor i remained alive for at least K time when at y_i and $B_j > B_i$. Thus, we may assume that $y'_i + d'_i < y_i$ and $y'_j = y'_i + d'_i$.

We have that $y'_i = y_j$ and $y'_j < y_i$. It follows that $\beta'_i + \beta'_j > \beta_i + \beta_j$. Also,

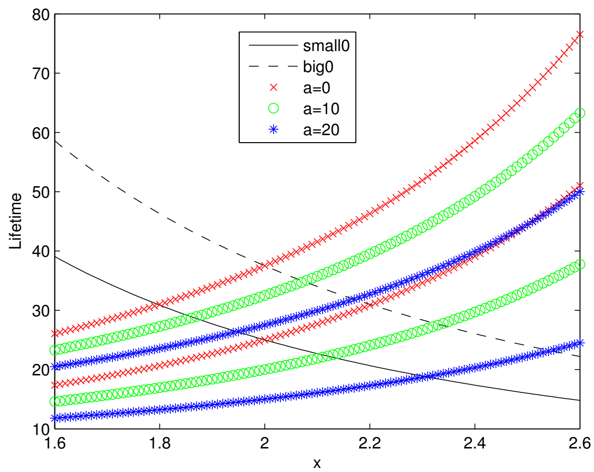


Figure 4.2: Comparing the lifetimes of sending a larger battery relay farther vs. the lifetime of sending a smaller battery relay farther for different values of a.

notice that $\beta_i < \min\{\beta_j, \beta_i', \beta_j'\}$. Then,

$$\begin{aligned} d_j' &= d_i + d_j - d_i' \\ &= d_i + d_j - \sqrt[\alpha]{\beta_i'/K} \\ &\leq \sqrt[\alpha]{\beta_i/K} + \sqrt[\alpha]{\beta_j/K} - \sqrt[\alpha]{\beta_i'/K} \\ &\leq \sqrt[\alpha]{\beta_j'/K}\ , \end{aligned}$$

where the inequality is due to Lemma 2.3.4. Consequently, $K \leq \beta_j'/{d_j'}^{\alpha} = L_j'$. We get a deployment y' that covers $[0, p]$ with lifetime T with a smaller number of violations than y. A contradiction. $\square$

In Figure 4.2, we compare the lifetime of keeping a relay with larger battery fixed and moving the relay with smaller battery to the lifetime of keeping a relay with smaller battery fixed and moving the relay with larger battery. The initial battery levels are $B_l = 150$ for the larger relay l and $B_s = 100$ for the smaller relay s. The interval of transmission is $I = (0, 4]$ and $\alpha = 2$. Big0 is the function $150/x^2$ which is the lifetime of relay l when fixed at 0 and relay s located at position x. Small0 is the function $100/x^2$ which is the lifetime of relay s when fixed at 0 and relay l located at position x. There are also the lifetime functions for the traveling relays which depend on a and position x: $L_{150}(a, x) = (150 - ax)/(4 - x)^\alpha$ is the lifetime of relay l after moving to a position x and $L_{100}(a, x) = (100 - ax)/(4 - x)^\alpha$ is the lifetime of relay s after moving to a position x. It follows that $(150 - ax)/(4 - x)^\alpha > (100 - ax)/(4 - x)^\alpha$ for a given a and for all $x \in I$.

The position x which maximizes the network lifetime is the position where the functions L_{150} and small0 (or L_{100} and big0) intersect. When $a = 0$, $L_{150}(0, x)$ and $L_{100}(0, x)$ intersect small0 and big0 respectively at exactly the same lifetime. Note that when $a = 10$, $L_{150}(10, x)$ intersects small0 at a larger lifetime than does $L_{100}(10, x)$ intersect big0. When $a = 20$, the difference in lifetimes of the intersection points and hence the networks is even larger. As a increases this trend continues.

Next, we consider the instance when all sensors are initially located at

D. We define *reverse-battery ordering* to be $B_i \leq B_j$ if and only if $y_j \leq y_i$. In this ordering larger battery relays are deployed to farther locations from D than smaller battery relays. Since the deployment occurs from D, larger battery relays have smaller final locations than smaller battery relays and are thus in decreasing (reverse) initial battery order when viewed from 0 to D.

We show that if there is a ordering that transmits for T time, then there is a *reverse-battery ordering* which transmits for T time. We consider the position p of the leftmost relay during transmission where $p \in (0, D)$ as this will be necessary for the general case.

Lemma 4.6.2. *Assume $x_i = D$ for all i and let $p \in (0, D)$. Suppose that there exists a deployment that transmits from p to D for T time where p is the position of the leftmost relay. Then, there exists a deployment that transmits from p to D for T time that satisfies reverse-battery ordering.*

Proof. The proof is quite similar to the proof of 4.6.1. Consequently only the key differences are provided.

If y has ordering violations, then there must exist at least one violation due to a pair of adjacent relays. Let j, i with $B_j < B_i$ but $y_j < y_i$ be such relays. We may assume that $L_i = L_j = K \geq T$: if $L_j > L_i$ then we may move sensor i to the right continuously increasing L_i and decreasing L_j until we reach a location where $L_i = L_j$; if $L_j < L_i$ then we may move sensor i to the left continuously decreasing L_i and increasing L_j until we reach a location where $L_i = L_j$.

Consider a solution y' which is equivalent to y, except for relay i located at $y'_i = y_j$ and relay j located at $y'_j = y'_i + d'_i$, where $d'_i = \sqrt[\alpha]{\beta'_i/K}$. Notice that sensor i can reach y_j as $B_i > B_j$ and furthermore notice that $y'_j > y_i$ as $B_i > B_j$. Also, no other sensors lifetime is affected by this reassignment. Relay i remains alive for K time with assignment y'. It follows as in the proof of 4.6.1 that $K \leq \beta'_j/{d'_j}^{\alpha} = L'_j$. $\square$

We now find a ordering for the case where $x_i \in \{0, D\}$ for all i between relays at 0 and relays at D. We assume that $i < j$ when $x_i = 0$ and $x_j = D$.

Lemma 4.6.3. *Let $x \in \{0, D\}^n$, and let T be an achievable lifetime. Then, there exists a feasible solution (y, r) with lifetime T such that $y_i \leq y_j$, for every $i \leq \ell < j$, where ℓ is the number of relays initially located at 0.*

Proof. Given a deployment y, a pair of relays is called *bad* if $i \leq \ell < j$ and $y_i > y_j$. Let y be a deployment with lifetime T that minimizes the number of bad pairs. If there are no bad pairs, then we are done. Otherwise, we show that the number of bad pairs can be decreased. If y has a bad pair, then there must exist at least one bad pair of adjacent relays. Let i and j be such relays. We construct a new deployment y' by placing relay i at y_j (closer to 0) and placing relay j at $y_j + d_i$ (closer to D), maintaining the transmission distance of both i and j, and not affecting the transmission distance of any other relay. More specifically,

$$y'_k = \begin{cases} y_j & k = i, \\ y_j + d_i & k = j, \\ y_k & k \neq i, j. \end{cases}$$

We show that y' is a feasible solution. We have that $d'_k = d_k$ for all k. Since $y'_k = y_k$ for all $k \neq i, j$ it follows that $L'_k = L_k \geq T$ for all $k \neq i, j$. Both relays i and j maintain their transmission distances and move closer to their initial positions so that $L'_k > L_k \geq T$ for $k = i, j$. Since the number of bad pairs decreases, we have a contradiction. $\square$

We show that we may assume that the relays are deployed using *bi-directional battery order*. In *bi-directional battery order* relays initially located at 0 are positioned to the left of relays initially located at D, relays initially located at 0 are positioned according to battery order, and relays initially at D are positioned according to reverse-battery order.

Theorem 4.6.4. *Let $x \in \{0, D\}^n$ and let T be an achievable lifetime. Then there exists a feasible solution y with lifetime T such that the sensors are deployed using bi-directional battery order.*

Proof. By Lemma 4.6.3 we know that there exists a deployment y, such that $y_i \leq y_j$, for every $i \leq \ell < j$, where ℓ is the number of relays initially located at 0. It follows that relays from 0 transmit in $[0, y_{\ell+1}]$ while relays initially at D transmit in $[y_{\ell+1}, D]$. Lemma 4.6.1 implies that there is a deployment y^0 of the relays from 0 that transmit in $[0, y_{\ell+1}]$ that satisfies battery order. Lemma 4.6.2 implies that there is a deployment y^1 of relays from D that transmit in $[y_{\ell+1}, D]$ that satisfies reverse battery order. Define

$$y'_i = \begin{cases} y_i^0 & i \leq \ell, \\ y_i^1 & i > \ell. \end{cases}$$

y' transmits in $[0, D]$ and it satisfies bi-directional battery order. $\square$

Knowing we may deploy the relays with initial locations at either 0 or D in bi-directional battery order, we may use either *Lifetime-DP* or *Lifetime Check* with binary search to solve discrete MaxFD for this instance and to obtain an FPTAS for this instance when in addition $B_i > aD(1 + \frac{1}{n})$ for all i.

Theorem 4.6.5. *Discrete* MaxFD *with* $x \in \{0, D\}^n$ *can be solved optimally.*

Theorem 4.6.6. *There exists an FPTAS for the instance of* MaxFD *where* $x \in \{0, D\}^n$ *and* $B_i > aD(1 + \frac{1}{n})$ *for all* i.

4.7 Conclusion

This chapter considers the problem of maximizing the transmission lifetime of a network with mobile relays. Two notions of network lifetime are considered: lifetime of first death and transmission lifetime. The scenario where there is no cost of movement is solved optimally for both lifetime of first death and transmission lifetime, and in fact, it is shown that transmission lifetime is equivalent to lifetime of first death for the no cost of movement case. Some structural results are then provided. In particular it is shown that for transmission lifetime, relays need not move at a time that does not correspond to the death of a sensor (which also shows that only one deployment suffices for lifetime of first death) and the transmitter must be the last node to die in any optimal solution.

For the objective of maximizing the lifetime of first death, the discrete model is first considered. Two algorithms, *Lifetime-DP* and *Lifetime Check with binary search* are created and shown to be optimal on the discrete model, with *Lifetime-DP* having better worst-case performance. These algorithms are shown to be FPTAS for the non-discrete model if battery sizes are not too small, with *Lifetime Check with binary search* having better worst case performance.

The scenario where relays are initially located at the base stations (end-points of the transmission interval) is then considered. It is shown that there is a dominant order in which to deploy the relays, namely in increasing order of initial battery size from 0 and in decreasing order of battery size from D with relays initially at 0 being positioned to the left of relays initially at D. Applying the algorithms with this ordering gives an optimal solution for the discrete model and an FPTAS for the non-discrete model if battery sizes are not too small.

We list several open problems:

- Is MaxTL NP-Hard when there is friction?
- Does one deployment suffice for MaxTL?

Finally, there are a number of possible natural generalizations for our problem:

- The initial and final locations of the relays can be anywhere in the plane.
- Movement takes time. In such a model relays are forced to move and transmit simultaneously, since transmission is not paused during redeployment.

Bibliography

[1] Underground spies to secure indo-pakistan border. `http://www.homelandsecuritynewswire.com/dr20120815-underground-spies-to-secure-indopakistan-border`. Accessed: 10-26-2013.

[2] Alessandro Agnetis, Enrico Grande, Pitu B. Mirchandani, and Andrea Pacifici. Covering a line segment with variable radius discs. *Computers & OR*, 36(5):1423–1436, 2009.

[3] Giuseppe Anastasi, Marco Conti, Mario Di Francesco, and Andrea Passarella. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3):537–568, 2009.

[4] R. Appuswamy, E. Atsan, C. Fragouli, and M. Franceschetti. On relay placement for deterministic line networks. In *Wireless Network Coding Conference (WiNC), 2010 IEEE*, pages 1–9, June 2010.

[5] Amotz Bar-Noy and Ben Baumer. Maximizing network lifetime on the line with adjustable sensing ranges. In *7th ALGOSENSORS*, volume 7111 of *LNCS*, pages 28–41, 2011.

[6] Amotz Bar-Noy, Ben Baumer, and Dror Rawitz. Changing of the guards: Strip cover with duty cycling. In *19th SIROCCO*, volume 7355 of *LNCS*, pages 36–47, 2012.

[7] Amotz Bar-Noy, Ben Baumer, and Dror Rawitz. Set it and forget it: Approximating the set once strip cover problem. Technical Report 1204.1082, CoRR, 2012.

[8] Amotz Bar-Noy, Theodore Brown, Matthew P. Johnson, and Ou Liu. Cheap or flexible sensor coverage. In *5th IEEE DCOSS*, volume 5516 of *LNCS*, pages 245–258, 2009.

[9] Amotz Bar-Noy, Dror Rawitz, and Peter Terlecky. "Green" barrier coverage with mobile sensors.

[10] Amotz Bar-Noy, Dror Rawitz, and Peter Terlecky. Maximizing barrier coverage lifetime with mobile sensors. In *ESA*, volume 8125 of *LNCS*, pages 97–108, 2013.

[11] Stefano Basagni, Alessio Carosi, Emanuel Melachrinoudis, Chiara Petrioli, and Z. Maria Wang. Controlled sink mobility for prolonging wireless sensor networks lifetime. *Wireless Networks*, 14(6):831–858, 2008.

[12] Binay K. Bhattacharya, Mike Burmester, Yuzhuang Hu, Evangelos Kranakis, Qiaosheng Shi, and Andreas Wiese. Optimal movement of mobile sensors for barrier coverage of a planar region. *Theor. Comput. Sci.*, 410(52):5515–5528, 2009.

[13] 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.

[14] Arnab Chakrabarti, Ashutosh Sabharwal, and Behnaam Aazhang. Using predictable observer mobility for power efficient design of sensor networks. In Feng Zhao and Leonidas J. Guibas, editors, *IPSN*, volume 2634 of *Lecture Notes in Computer Science*, pages 129–145. Springer, 2003.

[15] Erin Wolf Chambers, Sándor P Fekete, Hella-Franziska Hoffmann, Dimitri Marinakis, Joseph SB Mitchell, Venkatesh Srinivasan, Ulrike Stege, and Sue Whitesides. Connecting a set of circles with minimum sum of radii. In *Algorithms and Data Structures*, pages 183–194. Springer, 2011.

[16] Anurag Chandra, Deepak Bansal, Rajeev Shorey, Ashutosh Kulshreshtha, and Manish Gupta. Characterization of mobility patterns based on cell topography in a cellular radio system. In *IEEE ICPWC*, pages 428–432, 1999.

[17] Danny Z. Chen, Yan Gu, Jian Li, and Haitao Wang. Algorithms on minimizing the maximum sensor movement for barrier coverage of a linear domain. In *13th SWAT*, volume 7357 of *LNCS*, pages 177–188, 2012.

[18] Jurek Czyzowicz, Evangelos Kranakis, Danny Krizanc, Ioannis Lambadaris, Lata Narayanan, Jaroslav Opatrny, Ladislav Stacho, Jorge Urrutia, and Mohammadreza Yazdani. On minimizing the maximum sensor movement for barrier coverage of a line segment. In *8th ADHOC-NOW*, volume 5793 of *LNCS*, pages 194–212, 2009.

[19] Jurek Czyzowicz, Evangelos Kranakis, Danny Krizanc, Ioannis Lambadaris, Lata Narayanan, Jaroslav Opatrny, Ladislav Stacho, Jorge Urrutia, and Mohammadreza Yazdani. On minimizing the sum of sensor movements for barrier coverage of a line segment. In *9th ADHOC-NOW*, volume 6288 of *LNCS*, pages 29–42, 2010.

[20] Mario Di Francesco, Sajal K. Das, and Giuseppe Anastasi. Data collection in wireless sensor networks with mobile elements: A survey. *ACM TOSN*, 8(1):7:1–7:31, August 2011.

[21] Stefan Dobrev, Stephane Durocher, Mohsen Eftekhari, Konstantinos Georgiou, Evangelos Kranakis, Danny Krizanc, Lata Narayanan, Jaroslav Opatrny, Sunil Shende, and Jorge Urrutia. Complexity of barrier coverage with relocatable sensors in the plane. In *Algorithms and Complexity*, volume 7878 of *LNCS*, pages 170–182. 2013.

[22] Fatmé El-Moukaddem, Eric Torng, and Guoliang Xing. Maximizing data gathering capacity of wireless sensor networks using mobile relays. In *MASS*, pages 312–321. IEEE, 2010.

[23] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

[24] M. Gibson and K. Varadarajan. Decomposing coverings and the planar sensor cover problem. In *FOCS*, pages 159–168, 2009.

[25] David Kiyoshi Goldenberg, Jie Lin, and A. Stephen Morse. Towards mobility as a network control primitive. In Jun Murai, Charles E. Perkins, and Leandros Tassiulas, editors, *MobiHoc*, pages 163–174. ACM, 2004.

[26] Mohamed Hefeeda and Majid Bagheri. Wireless sensor networks for early detection of forest fires. In *MASS*, pages 1–6, 2007.

[27] Sushant Jain, Rahul Shah, Waylon Brunette, Gaetano Borriello, and Sumit Roy. Exploiting mobility for energy efficient data collection in wireless sensor networks. *MONET*, 11(3):327–339, 2006.

[28] Zhen Jiang, Jie Wu, and Robert Kline. Mobility control for achieving optimal configuration in wireless sensor networks. *Wireless Networks*, 16(1):141–152, 2010.

[29] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li-Shiuan Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. In *ASPLOS*, pages 96–107, 2002.

[30] Aman Kansal, Mohammad Rahimi, William J Kaiser, Mani Srivastava, Gregory J Pottie, and Deborah Estrin. Controlled mobility for sustainable wireless networks. In *IEEE SECON*, 2004.

[31] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack problems*, pages 412–413. Springer Verlag, 2004.

[32] Hyung Seok Kim, Tarek F. Abdelzaher, and Wook Hyun Kwon. Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks. In Ian F. Akyildiz, Deborah Estrin, David E. Culler, and Mani B. Srivastava, editors, *SenSys*, pages 193–204. ACM, 2003.

[33] Sukun Kim, Shamim Pakzad, David Culler, James Demmel, Gregory Fenves, Steven Glaser, and Martin Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *Proceedings of the 6th international conference on Information processing in sensor networks*, IPSN '07, pages 254–263, New York, NY, USA, 2007. ACM.

[34] 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.

[35] Minming Li. Private Communication.

[36] Minming Li, Xianwei Sun, and Yingchao Zhao. Minimum-cost linear coverage by sensors with adjustable ranges. In *6th WASA*, volume 6843 of *LNCS*, pages 25–35, 2011.

[37] Mona Mehrandish, Lata Narayanan, and Jaroslav Opatrny. Minimizing the number of sensors moved on line barriers. In *WCNC*, pages 653–658, 2011.

[38] Vivek Mhatre, Catherine Rosenberg, Daniel Kofman, Ravi Mazumdar, and Ness B. Shroff. A minimum cost heterogeneous sensor network with a lifetime constraint. *IEEE Trans. Mob. Comput.*, 4(1):4–15, 2005.

[39] Thomas Moscibroda, Roger Wattenhofer, and Aaron Zollinger. Topology control meets sinr: : the scheduling complexity of arbitrary topologies. In Sergio Palazzo, Marco Conti, and Raghupathy Sivakumar, editors, *MobiHoc*, pages 310–321. ACM, 2006.

[40] Brian Phelan, Peter Terlecky, Amotz Bar-Noy, Theodore Brown, and Dror Rawitz. Should I stay or should I go? Maximizing lifetime with relays. In *8th DCOSS*, pages 1–8, 2012.

[41] Rahul C. Shah, Sumit Roy, Sushant Jain, and Waylon Brunette. Data mules: modeling and analysis of a three-tier architecture for sparse sensor networks. *Ad Hoc Networks*, 1(2-3):215–233, 2003.

[42] Xuehou Tan and Gangshan Wu. New algorithms for barrier coverage with mobile sensors. In *4th FAW*, volume 6213 of *LNCS*, pages 327–338, 2010.

[43] Guiling Wang, Mary Jane Irwin, Piotr Berman, Haoying Fu, and Thomas F. La Porta. Optimizing sensor movement planning for energy efficiency. In Kaushik Roy and Vivek Tiwari, editors, *ISLPED*, pages 215–220. ACM, 2005.