“Green” Barrier Coverage with Mobile Sensors[*]

Amotz Bar-Noy[†]
amotz@sci.brooklyn.cuny.edu

Thomas Erlebach[‡]
t.erlebach@leicester.ac.uk

Dror Rawitz[§]
dror.rawitz@biu.ac.il

Peter Terlecky[¶]
pterlecky@gc.cuny.edu

January 30, 2021

Abstract

Mobile sensors are located on a barrier represented by a line segment. Each sensor has a single energy source that can be used for both moving and sensing. 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. We address the problem of energy efficient coverage. The input consists of the initial locations of the sensors and a coverage time requirement t. A feasible solution consists of an assignment of destinations and coverage radii to all sensors such that the barrier is covered. We consider two variants of the problem that are distinguished by whether the radii are given as part of the input. In the *fixed* radii case, we are also given a radii vector ρ, and the radii assignment r must satisfy $r_i \in \{0, \rho_i\}$, for every i, while in the *variable* radii case the radii assignment is unrestricted. The goal is to cover the barrier for t time in an energy efficient manner. More specifically, we consider two objective functions. In the first the goal is to minimize the sum of the energy spent by all sensors and in the second the goal is to minimize the maximum energy used by any sensor.

We present fully polynomial time approximation schemes for the problem of minimizing the energy sum with variable radii and for the problem of minimizing the maximum energy with variable radii. We also show that the latter can be approximated within any additive constant $\varepsilon > 0$. We present a 2-approximation algorithm for the problem of minimizing the maximum energy with fixed radii which also is shown to be strongly NP-hard. We show that the problem of minimizing the energy sum with fixed radii cannot be approximated within a factor of $O(n^c)$, for any constant c, unless P=NP. Additional results are given for three special cases: (i) sensors are stationary, (ii) free movement, and (iii) uniform fixed radii.

Keywords: Approximation algorithms, Barrier coverage, Energy conservation, Mobile sensors, Sensor deployment, Sensor networks.

---

[*]A preliminary version was presented at the 9th International Conference on Algorithms and Complexity (CIAC), 2015.

[†]Graduate Center, City University of New York, NY 10016, USA.

[‡]Department of Informatics, University of Leicester, University Road, Leicester, LE1 7RH, UK.

[§]Faculty of Engineering, Bar-Ilan University, Ramat Gan 52900, Israel.

[¶]Graduate Center, City University of New York, NY 10016, USA.

1 Introduction

Battery lifetime is a significant bottleneck on wireless sensor network performance. Thus, one of the fundamental problems in sensor networks is optimizing battery usage when accomplishing tasks such as covering, monitoring, tracking and communicating. We study the problem of covering a boundary or a barrier by mobile sensors, e.g., covering borders, coastlines, railroads, etc. Also, often covering region boundaries is the cost efficient way of protecting the interior. The focus of this paper is to determine what is 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. Prior work tried to optimize either covering costs or mobility costs but not a combination of both costs. We consider a model where energy is consumed by sensing and movement from a single battery source as is most commonly the architecture [2].

1.1 Model

We consider a setting where there are n mobile sensors initially located on a barrier represented by the interval $[0, 1]$. (It is convenient, but not essential, to assume that the sensors are located on the barrier.) Let $x = (x_1, \dots, x_n)$ be the initial position vector, where x_i is the initial position of sensor i. We assume that the sensor positions are given in sorted order, i.e., $x_1 \le x_2 \le \cdots \le x_n$. We consider the *set-up and sense* model [12, 11, 13, 23, 7], where sensors first move to their desired destinations and then begin sensing. Let $y = (y_1, \dots, y_n)$ be the deployment vector, where y_i is the destination position of sensor i. The system works in two phases. In the *deployment phase*, sensor i moves from its initial position x_i to its destination y_i. Without loss of generality, this phase ends 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 $r = (r_1, \dots, r_n)$ be the radii vector. We call this interval the *covering interval* of sensor i. An example of movement and coverage by one sensor is given in Fig. 1. It is required that the sensors collectively cover the unit interval, i.e., $[0, 1] \subseteq \cup_i [y_i - r_i, y_i + r_i]$. 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 consumption per time unit of sensor i with a covering radius of r_i is r_i^α, where $\alpha \ge 1$ is the *path-loss exponent* [16]. Given 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.)

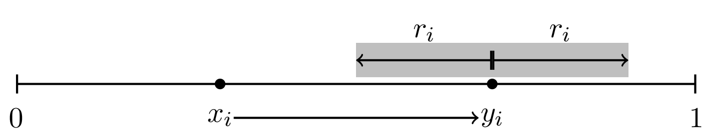


Figure 1: Sensor i moves from x_i to y_i and covers the interval $[y_i - r_i, y_i + r_i]$.

1.2 The 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 minimizing $\sum_i E_i(y, r)$; and, (ii) minimizing the maximum amount of energy expended, i.e., minimizing $\max_i E_i(y, r)$. 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 pair of 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 $y_i \leq y_j$.

1.3 Related Work

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.

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. [12] 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 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 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. [11] gave a polynomial time algorithm for the more general case in which the sensing radii are non-uniform, namely for MaxFix_1 with $t = 0$, and improved upon the running time for MaxFix_1 with uniform radii and $t = 0$.

Czyzowicz et al. [13] 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 SumFix_1 in which $t = 0$ (without loss of generality $a = 1$). They presented a polynomial time algorithm for 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 problems in which movement is optimized. We list several examples. Mehrandish et al. [20] considered the same model with the objective of minimizing the number of sensors which must move to cover the barrier. Dobrev et al. [15] studied the problem of covering a set of barriers attempting to optimize movement costs. Tan and Wu [22] 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. [8]. Demaine et al. [14] studied minimizing movement of pebbles in a graph in order to obtain a property (such as connectivity, independence, and perfect matchability), and the goal is to minimize maximum movement, total movement, or number of movements.

In many papers it is assumed that sensors are static, and the goal is to minimize sensing energy. Li et al. [16] presented a polynomial time algorithm for $\textsc{SumFix}_\infty$ and an FPTAS for $\textsc{SumVar}_\infty$ with $\alpha = 1$. They also showed that $\textsc{SumVar}_\infty$ with $\alpha = 1$ is NP-hard. Agnetis et al. [1] considered an extension of $\textsc{SumVar}_\infty$ 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., [19, 6]).

Another common research direction is to consider the dual problem which is to maximize the lifetime of the network where the battery sizes are given. See, e.g., [9, 17, 3, 4, 5, 23, 7].

Chambers et al. [10] looked at the problem of finding, for given points in the plane, 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 was also considered. Buchsbaum et al. [9] and Gibson and Varadarajan [17] 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. [3, 4, 5] and Poss and Rawitz [21] studied stationary sensors with variable radii that may start covering at any time.

To the best of our knowledge our earlier papers [23, 7] and this work are the first to consider energy consumption from moving and sensing from a single battery source. In [23] we attempted to maximize the transmission lifetime of mobile battery-powered relays on a direct line from source to sink, and in [7] we considered maximizing barrier coverage lifetime of a network of mobile battery-powered sensors.

1.4 Our Results

SumVar is studied in Section 2, where we present an $O(n)$ time algorithm for $\textsc{SumVar}_0$ and an FPTAS for SumVar, for any a. The latter is based on the FPTAS for $\textsc{SumVar}_\infty$ with $\alpha = 1$ by Li et al. [16]. However, we introduce several new ideas in order to cope with sensor mobility and with $\alpha > 1$. In particular we show that there exists a non-swapping optimal solution and use the optimal value for $a = 0$ as a lower bound for the case where $a > 0$.

Section 3 deals with MaxVar. We present an FPTAS for MaxVar that is similar to the SumVar FPTAS. However, while the SumVar non-swapping property is reminiscent of previous non-swapping results for uniform radii (see, e.g., [12, 13]), proving MaxVar non-swapping is more challenging and requires a rigorous case analysis. We present $O(n)$ time algorithms for $\textsc{MaxVar}_0$ and for $\textsc{MaxVar}_\infty$, and we also show that MaxVar can be approximated to within an additive approximation $\varepsilon > 0$, for any constant $\varepsilon > 0$, assuming efficient infinite precision computations. This result is based on the non-swapping property and on [7].

The results for variable radii are given in Table 1.

In Section 4 we study MaxFix. We provide an $O(n \log n)$ time algorithm for $\textsc{MaxFix}_0$. 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 $x = (\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2})$, and this result

Problem	$a = 0$	$a = \infty$	$a \in (0, \infty)$
SumVar	OPT	$\alpha = 1$: NP-hard, FPTAS [16] FPTAS	FPTAS
MaxVar	OPT	OPT	FPTAS, OPT+ε

Table 1: Summary of results for variable radii.

Problem	$a = 0$	$a = \infty$	$a \in (0, \infty)$	
			uniform radii	general radii
MaxFix	OPT	OPT	$t = 0$: OPT [12, 11]	$t = 0$: OPT [11] 2-approximation strongly NP-hard
SumFix	FPTAS $\alpha = 1$: NP-hard	OPT [16]	$t = 0$: OPT [13] OPT+ε $a = 0$: OPT	$\omega(1)$ [13] $\Omega(n^c)$, for any c

Table 2: Summary of results fixed radii.

implies that it is NP-hard to find an optimal ordering. On the other hand, we provide a polynomial time 2-approximation algorithm for MaxFix.

We study SumFix in Section 5. We show that SumFix cannot be approximated within a factor of $O(n^c)$, for any constant c, unless P=NP. We show that SumFix$_0$ is NP-hard for $\alpha = 1$, and provide an FPTAS for SumFix$_0$ for any α. We also prove that SumFix with uniform radii can be approximated to within an additive approximation $\varepsilon > 0$, for any constant $\varepsilon > 0$.

The results for fixed radii are given in Table 2.

2 Minimum Sum Energy 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. [16] who gave an FPTAS for SumVar$_\infty$ with $\alpha = 1$. We note that several new and non-trivial ideas were introduced in order to cope with mobility and with $\alpha > 1$.

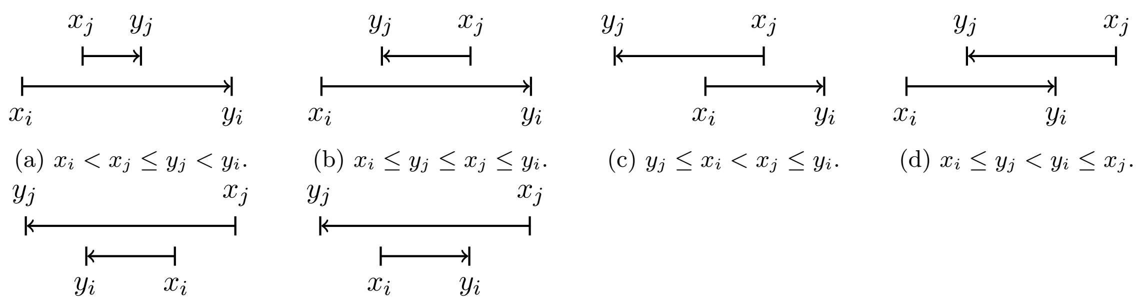
2.1 Zero Friction

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

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

Proof. Given a SumVar$_0$ instance (x, t), let $r_i = \frac{1}{2n}$, for all i, and let $y_i = \frac{2i-1}{2n}$, 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

$$\sum_i E_i(y, r) = nt\left(\frac{1}{2n}\right)^\alpha \leq \sum_i t(r'_i)^\alpha = \sum_i E_i(y', r') \ .$$



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

Figure 2: Six configurations of a swapping pair.

(i) Original deployment. (ii) New deployment.

Figure 3: Case (a): Swapping where $x_i \le x_j \le y_j \le y_i$.

Thus, (y, r) is optimal as well. Finally, notice that (y, r) can be computed in linear time. $\square$

Observe that the cost of the optimal solution may only increase as a increases. 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 Non-Zero Friction

Our FPTAS for the non-zero friction case 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 1. *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. There are six possible configurations for such a pair of sensors as shown in Figure 2.

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 $k \ne i, j$. Clearly, the barrier $[0, 1]$ remains covered. We show that the energy sum does not increase, since the total distance traveled by the sensors does not increase.

Case (a): $x_i < x_j \le y_j < y_i$. In this case both sensors redeploy to the right, as shown in Figure 3. Hence, $(y'_i - x_i) + (y'_j - x_j) = (y_i - x_i) + (y_j - x_j)$.

Case (e): $y_j < y_i \le x_i < x_j$. Symmetric to Case (a).

(i) Original deployment. (ii) New deployment.

Figure 4: Case (b): Swapping where $x_i \le y_j \le x_j \le y_i$.

(i) Original deployment. (ii) New deployment.

Figure 5: Case (c): Swapping where $x_i \le y_j < y_i \le x_j$.

Case (b): $x_i \le y_j \le x_j \le y_i$. In this case, the sensors pass each other, and j's path is contained in i's, as shown in Figure 4. Hence, $(y_i - x_i) + (x_j - y_j) \ge y_j - x_i + y_i - x_j = (y'_i - x_i) + (y'_j - x_j)$.

Case (f): $y_j \le x_i \le y_i \le x_j$. Symmetric to Case (b).

Case (c): $y_j \le x_i < x_j \le y_i$. In this case the sensors pass the original locations of each other, as shown in Figure 5. Hence $(y_i - x_i) + (x_j - y_j) \ge y_i - x_j + x_i - y_j = (y'_j - x_j) + (x_i - y'_i)$.

Case (d): $x_i \le y_j < y_i \le x_j$. In this case the sensors do not pass the original locations of each other, as shown in Figure 6. Hence, $(y_i - x_i) + (x_j - y_j) \ge y_j - x_i + x_j - y_i = (y'_i - x_i) + (x_j - y'_j)$.

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

$$\mathcal{G} = \{x_i : i \in \{1, \ldots, n\}\} \cup \left\{ \frac{j}{m} : j \in \{0, \ldots, m\} \right\} .$$

The points in $\mathcal{G}$ are called *grid points*. Let $g_0, \ldots, g_{n+m}$ be an ordering of grid points such that $g_i \le g_{i+1}$. Given a point $p \in [0, 1]$, let p^+ be the left-most grid point to the right of p, namely $p^+ = \min \{g \in \mathcal{G} : g \ge p\}$. Similarly, $p^- = \max \{g \in \mathcal{G} : g \le p\}$ 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, and a 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. *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*

$$\sum_i E_i(y', r') \le (1 + 2\varepsilon) \sum_i E_i(y, r) .$$

(i) Original deployment. (ii) New deployment.

Figure 6: Case (d): Swapping where $x_i \le y_j < y_i \le x_j$.

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:

$$y'_i = \begin{cases} y_i^+ & y_i \le x_i, \\ y_i^- & y_i > x_i. \end{cases}$$

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 $(g_j, g_{j+1}) \cap [y_i - r_i, y_i + r_i] \ne \emptyset$, 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

$$\sum_i E_i(y', r') \le (1 + 2\varepsilon) \sum_i E_i(y, r) \ .$$

Since y'_i can only be closer than y_i to x_i, we have that $|y'_i - x_i| \le |y_i - x_i|$. 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, $r'_i \le r_i + \frac{2}{m}$.

If $r_i \ge \frac{1}{2\mu}$, then

$$r'_i \le r_i + \frac{2}{m} \le r_i + \frac{\varepsilon}{4\alpha\mu} \le r_i \left(1 + \frac{\varepsilon}{2\alpha}\right) \ .$$

Hence,

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

Otherwise, if $r_i < \frac{1}{2\mu}$, then

$$r'_i \le r_i + \frac{2}{m} \le \frac{1}{2\mu} + \frac{\varepsilon}{4\alpha\mu} \le \frac{1}{\mu} \ .$$

Hence,

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

Putting it all together we get

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

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 after Theorem 1. $\square$

Lemma 2 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 3. *There exists an $O(nm^4)$ time algorithm that finds the optimal discrete non-swapping solution.*

Proof. The dynamic programming table is denoted by Π, and it is constructed as follows. The entry $\Pi(i,\ell,k)$, where i is a sensor number, $\ell \in \{0,\ldots,n+m\}$, 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 ith sensor is located anywhere in $[0,g_\ell]$, 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

$$\Pi(i,\ell,k) = \min_{\ell' \leq \ell} \left\{ a|g_{\ell'} - x_i| + \min\left\{ \Pi(i-1,\ell',k), \min_{k'<k}\left\{ \Pi(i-1,\ell',k') + tr_i^\alpha \right\} \right\} \right\} \tag{2}$$

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) is the energy 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. $\square$

Lemma 2 and the above algorithm lead to an FPTAS for SumVar.

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

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 sensors to cover the interval $[0,g_k]$. Also, (2) is changed to

$$\Pi(i,k) = \min\left\{ \Pi(i-1,k), \min_{k'<k}\left\{ \Pi(i-1,k') + tr_i^\alpha \right\} \right\}, \tag{3}$$

where $r_i = \max\{x_i - g_{k'}, g_k - x_i\}$. An entry can be computed in $O(m)$, and the total running time is $O(nm^2)$. We get the following result.

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

3 Minimum Max Energy with Variable Radii

In this section we consider MaxVar. We present a linear time algorithm for MaxVar$_0$ and an FPTAS for the case where $a>0$. We also show a linear time algorithm for the case where $a=\infty$. For the non-zero finite friction case, along with providing an FPTAS, we also provide an algorithm that computes solutions within additive factor ε, for any constant $\varepsilon > 0$.

3.1 Zero Friction

Theorem 2. $\textsc{MaxVar}_0$ *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 j for which $r'_j > r_j$. Thus, we have

$$\max_i E_i(y', r') \geq E_j(y', r') > E_j(y, r) = \max_i E_i(y, r) ,$$

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 $a > 0$.

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 4. *Let* (x, T) *be a* $\textsc{MaxVar}_\infty$ *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 $R = \sqrt[\alpha]{E/t}$. The solution $(x, (R, R, \ldots, R))$ is feasible, since $R \geq r_i$, for every i. Also, $\max_i E_i(x, (R, R, \ldots, R)) = \max_i E_i(x, r)$. We prove the lemma by showing that $\max_i E_i(x, (R, R, \ldots, R)) \geq t\Delta^{\alpha}$.

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 3. $\textsc{MaxVar}_\infty$ *can be solved in* $O(n)$ *time.*

Proof. It is not hard to verify that the pair $(x, (\Delta, \Delta, \ldots, \Delta))$ covers $[0, 1]$. Furthermore, $\max_i E_i(x, (\Delta, \Delta, \ldots, \Delta)) = t\Delta^{\alpha}$, which means that it is optimal due to Lemma 4. Finally, Δ can be computed in $O(n)$ time. □

3.3 Non-zero Finite Friction: FPTAS

In this section we provide an FPTAS for MaxVar with non-zero finite friction that is based on the same approach that was 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.

As mentioned above, we prove that there is no need to consider solutions which swap sensors, but as opposed to the proof of Lemma 1, the proof for MaxVar is more involved and requires case analysis. We will need the following lemma that was proven in [7] and whose proof is provided for completeness.

Lemma 5 ([7]). *Let $\beta_1, \beta_2, \gamma_1, \gamma_2 \geq 0$ such that (i) $\gamma_1 < \beta_1 \leq \beta_2$, and (ii) $\beta_1 + \beta_2 \geq \gamma_1 + \gamma_2$. Also let $\alpha \geq 1$. Then, $\sqrt[\alpha]{\beta_1} + \sqrt[\alpha]{\beta_2} \geq \sqrt[\alpha]{\gamma_1} + \sqrt[\alpha]{\gamma_2}$.*

Proof. The proof for $\alpha = 1$ is straightforwards, hence we assume that $\alpha > 1$.

Let $s = \beta_1 + \beta_2$, and define $\gamma_2' \stackrel{\text{def}}{=} s - \gamma_1$. We prove that $\gamma_1^{1/\alpha} + (s-\gamma_1)^{1/\alpha} < \beta_1^{1/\alpha} + (s - \beta_1)^{1/\alpha}$. The lemma follows, since $\gamma_2' > \gamma_2$. To prove the above, define $f(x) = x^{1/\alpha} + (s - x)^{1/\alpha}$. The derivative of f is:

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

$f'(x) = 0$ 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 $f(\beta_1) > f(\gamma_1)$. □

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 and a position p, 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 $E - a|p - x_i| > 0$.

Lemma 6. *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 $k \neq i, j$. As shown in the proof of Lemma 1, there are six possible configurations for such a pair of sensors as shown in Figure 2.

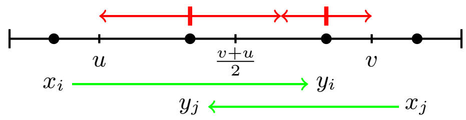
If the barrier can be covered without i, then i is moved to y_j. Sensor i has enough energy for moving to y_j, since either $|y_j - x_i| \leq |y_i - x_i|$ or $|y_j - x_i| \leq |y_j - x_j|$. Similarly, if the barrier is covered without j, then j is moved to y_i. Sensor j has enough energy to move to y_i, since either $|y_i - x_j| \leq |y_j - x_j|$ or $|y_i - x_j| \leq |y_i - x_i|$. In both cases we get a solution with fewer 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 (that are shown in Figure 2) we provide a solution (y', r') such that $y_k' = y_k$ and $r_k' = r_k$, for $k \neq i, j$, $y_i' \leq y_j'$, and the interval $[u, v]$ is covered by i and j. Moreover, $E_k(y', r') \leq E_k(y, r) = E$, for every k. Then, we eliminate any new swaps that may have been created by moving i and j. The resulting solution has fewer swaps than (y, r), and we get a contradiction.

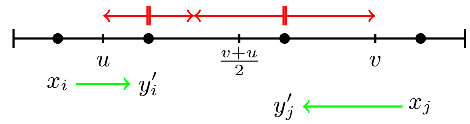
We start with cases (c) and (d), since they are easier. Then, we move to deal with the other cases. Newly created swaps will be considered later on.

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 $r_j' = r_i$ (as shown in Figure 5). 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 $E_j(y', r') \leq E_i(y, r)$.



(i) Original deployment.



(ii) New deployment.

Figure 7: Case (d): Swapping where $x_i \le y_j < y_i \le x_j$.

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

First, notice that since both i and j participate in the cover, we have that $y_j \le \frac{u+v}{2} \le y_i$. 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 $r'_j = r_j$. See example in Figure 7. Observe that $[u, v]$ remains covered as $y'_j - y'_i = y_i - y_j$. Also, we have that $y'_i \le \frac{u+v}{2} \le y_i$ and $y'_j \ge \frac{u+v}{2} \ge y_j$. If $y'_i \ge x_i$, then $d'_i \le d_i$. Otherwise, if $y'_i < x_i$, then

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

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

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

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 $x_i \le y_j$ 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 $z \ge x_i$, 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 $y'_j \le y_i$ and $y'_j - r'_j(y'_j, E) \le y'_i + r'_i$. 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 $z \ge x_j$, 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,

$$\beta_i(y_i, E) \le \beta_i(y'_i, E), \beta_j(y'_j, E) \ ,$$

and

$$\beta_i(y_i, E) + \beta_j(y_j, E) \le \beta_i(y'_i, E) + \beta_j(y'_j, E) \ .$$

By Lemma 5 we have that $r'_i + r'_j > r_i + r_j$, and thus

$$y'_j + r'_j = u + 2r'_i + 2r'_j > u + 2r_i + 2r_j \ge v \ .$$

Case (e): $y_j \le y_i \le x_i < x_j$. Symmetric to case (a).

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

In this case we have two options. First, if $d_j = x_j - y_j \ge y_j - x_i$, switch places and radii between i and j (see Figure 6).

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 $y'_j \le x_j$, then $d'_j < d_j$, and we have $d'_i + d'_j \le d_i + d_j$. 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 5 to show that $r'_i + r'_j > r_i + r_j$, and it follows that $y'_j + r'_j > v$.

Case (f): $y_j \le x_i \le y_i \le x_j$. Symmetric to case (b).

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 $S_R = \{k : x_k \ge x_i \wedge y_k \in (y'_i, y_i]\}$. By moving left to y'_i, i creates new swaps with sensors in S_L, but eliminates swaps with sensors in S_R. Let $\ell = \operatorname{argmin}_{k \in S_L}(y_k - r_k)$. If $y_\ell - r_\ell \ge u$, 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 $x_k \le y'_i$, 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.

Thus there is a solution with minimum maximum energy E with fewer swaps. A contradiction. □

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

Lemma 7. *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 $\max_i E_i(y', r') \le (1 + 2\varepsilon)\max_i E_i(y, r)$.*

Proof. The proof is almost the same as the proof of Lemma 2. The only difference is that Equation (1) should be replaced by

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

where the second inequality is due to $e^{\varepsilon/2} \le 1 + \varepsilon$, for any $\varepsilon \in (0, 1)$, and $\text{OPT} \ge t\frac{1}{(2n)^\alpha}$. □

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

Lemma 8. *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 3. The main difference is that Equation 2 should be replaced by

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

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

3.4 Non-zero Finite Friction: Additive Approximation

In this section we show that MaxVar can be approximated to within an additive factor ε, for any constant $\varepsilon > 0$. This is done using an algorithm that, given (x, t) and an energy level E, computes a solution with value at most $E+\varepsilon$, if E is feasible, and outputs "No Solution", if $E+\varepsilon$ is infeasible. Binary search is used to approximate the minimum energy level.

Our decision algorithm is based on an algorithm from [7] that solves a sort of a promise problem.

Theorem 4 ([7])**.** *Let $\varepsilon > 0$ and assume efficient infinite precision computations. Then there exists a polynomial time algorithm that, given x, $t \geq 0$, and an energy bound E, satisfies the following:*

1. *If there is a non-swapping* MaxVar *solution (y, r), such that $\max_i E_i^{t+\varepsilon}(y, r) \leq E$, it computes a solution (y', r'), such that $\max_i E_i^t(y', r') \leq E$.*
2. *If there is no non-swapping* MaxVar *solution (y, r), such that $\max_i E_i^t(y, r) \leq E$, it terminates with "No Solution".*

We note that the result in [7] is actually more general, in two ways: it copes with non-uniform energy levels and it works for any given final deployment order. (Here the initial order is the final order.)

Next we show that, given a solution, increasing the lifetime by ε amounts to an increase of at most ε in the energy consumption.

Observation 9. *Given x and $\varepsilon > 0$, if (y, r) is a solution such that $\max_i E_i^t(y, r) \leq E$, then $\max_i E_i^{t+\varepsilon}(y, r) \leq E + \varepsilon$.*

Proof. Since $r_i \leq 1$ for every i, we have that keeping the network alive for an additional ε time costs at most $\varepsilon \cdot 1^\alpha = \varepsilon$. □

The above observation allows us to "translate" the lifetime promise problem into an energy promise problem.

Theorem 5. *Let $\varepsilon > 0$ and assume efficient infinite precision computations. There exists a polynomial time algorithm that, given x, $t \geq 0$, and an energy bound E, satisfies the following:*

1. *If there is a non-swapping* MaxVar *solution (y, r), such that $\max_i E_i^t(y, r) \leq E - \varepsilon$, it computes a solution (y', r'), such that $\max_i E_i^t(y', r') \leq E$.*
2. *If $\max_i E_i^t(y, r) > E$ for any non-swapping* MaxVar *solution (y, r), it terminates with "No Solution".*

Proof. We call the algorithm from Theorem 4 with x, t and E. First, assume that there exists a non-swapping MaxVar solution (y, r), such that $\max_i E_i^t(y, r) \leq E - \varepsilon$. In this case, by Observation 9 we have that $\max_i E_i^{t+\varepsilon}(y, r) \leq E$. Hence, the algorithm computes a solution (y', r'), such that $\max_i E_i^t(y', r') \leq E$ as required. On the other hand, suppose that $\max_i E_i^t(y, r) > E$ for any non-swapping MaxVar solution (y, r). Hence, the algorithm terminates with "No Solution" as required. □

Due to Lemma 6 and Theorem 5 we can perform a binary search on E. An upper bound on E is t. Hence, the running time of the parametric search algorithm is polynomial in the input size and in $\log \frac{1}{\varepsilon}$, where ε is the accuracy parameter. Hence we get the following result.

Theorem 6. *There exists a polynomial time algorithm that, given a* MaxVar *instance and an accuracy parameter $\varepsilon > 0$, computes a solution whose value is within an additive constant ε of the optimum, assuming efficient infinite precision computations.*

4 Minimum Max Energy with Fixed Radii

In this section we study MaxFix. Recall that 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.

First, we provide $O(n \log n)$ time algorithms for both MaxFix_0 and MaxFix_∞. Czyzowicz et al. [12] presented an algorithm for MaxFix with uniform radii and $t = 0$. Chen et al. [11] improved upon the running time of the above problem 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 $x = (\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2})$, and that it is strongly NP-hard when radii are non-uniform. We note that our reductions are based on the fact that $t > 0$. We also present a 2-approximation algorithm for MaxFix.

4.1 Zero Friction

We describe a simple algorithm for solving MaxFix_0.

Theorem 7. MaxFix_0 *can be solved in $O(n \log n)$ time.*

Proof. First, observe that if $\sum_i 2\rho_i < 1$, the problem has no solution. 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. □

4.2 Infinite Friction

Next, we show that the static version of MaxFix can be solved in $O(n \log n)$ time.

Theorem 8. MaxFix_∞ *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 $i = \text{argmin}_{i \notin S} \rho_i$ 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 sensor 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 $[0,1]$. The list contains a set of non-intersecting segments ordered by their left end-point. Notice that the addition of a sensor may increase the size of the list by at most one due to splitting a segment. Hence, the list contains $O(n)$ segments, and therefore the insertion of a new sensor can be done in $O(n)$ time. Consequently, the total running time is $O(n^2)$.

A more efficient implementation can be obtained by storing the list in a balanced search tree. Given a new sensor i whose covering interval is $I_i = [x_i - \rho_i, x_i + \rho_i]$, search the left-most segment L_i and the right-most segment R_i in the list which intersect I_i. (Notice that it may be that $L_i = R_i$.) Replace L_i and R_i by $L_i \setminus I_i$ and $R_i \setminus I_i$. (An empty segment means removal.) In addition, remove all segments in the list that are located in between L_i and R_i, if such segments exist. Observe that the covering interval I_i of a sensor i may shorten one or two segments, and it may also split a segment. If I_i splits a segment S, we treat the left segment as a short version of S and the right segment as a new segment which is *owned* by i. (Only the segment $[0,1]$ has no owner.) Ownership still applies if a segment is shortened. We charge the removal of a segment to its owner. Hence each sensor is charged $O(\log n)$ time for searching and updating L_i and R_i and $O(\log n)$ time for the removal of its segment. It follows that the total running time is $O(n \log n)$. $\square$

4.3 Non-zero Finite Friction

As mentioned earlier, Czyzowicz et al. [12] presented a polynomial time algorithm for MaxFix with uniform radii and $t = 0$. Their result is based on showing that there exists a non-swapping optimal solution for the special case of uniform radii. We show that MaxFix is NP-hard, even if $x = (\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2})$, using a reduction from Partition. This implies that it is NP-hard to find an optimal ordering of a MaxFix instance. On the other hand, we present a 2-approximation algorithm for MaxFix that uses the algorithm for minimizing maximum movement, or for MaxFix with $t = 0$, by Chen et al. [11].

The description of the algorithm is as follows. For each $i \in \{1, \ldots, n\}$, execute the algorithm from [11] for minimizing maximum movement only on the sensors in the set $S_i = \{j : \rho_j \leq \rho_i\}$. The sensors that do not belong to S_i remain in their original locations. Let (y^i, r^i) be the computed solution for index i. Then, output the solution that minimizes the maximum energy, namely output (y^k, r^k), where $k = \operatorname{argmin}_i \max_j E_j(y^i, r^i)$.

Lemma 10. *Given a* MaxFix *instance (x, ρ, t), let (y^*, r^*) be an optimal solution. Then, $\max_j E_j(y^k, r^k) \leq 2 \max_j E_j(y^*, r^*)$.*

Proof. Let $E_M = a \cdot \max_i |y_i^* - x_i|$ be the maximum movement energy of the optimal solution and let $E_S = t \cdot \max_i (r_i^*)^\alpha$ be its maximum sensing energy. Clearly, $\max\{E_M, E_S\} \leq \max_i E_i(y^*, r^*)$.

Let ℓ be the sensor with the largest radius among active sensors in (y^*, r^*), i.e., $\ell = \operatorname{argmax}_i r_i^*$. It follows that $E_S = t \cdot \rho_\ell^\alpha$. By construction the solution (y^ℓ, r^ℓ) minimizes the maximum movement with respect to the set S_ℓ. Hence it also minimizes the maximum energy invested in movement with respect to S_ℓ. Since (y^*, r^*) is a solution with respect to S_ℓ, we have that $E_M \geq a \cdot \max_j \left| y_j^\ell - x_j^\ell \right|$. If follows that

$$\max_j E_j(y^\ell, r^\ell) \leq a \cdot \max_{j \in S_\ell} \left| y_j^\ell - x_j^\ell \right| + t \cdot \max_i (r_i^\ell)^\alpha \leq E_M + E_S \ .$$

The lemma follows, since $\max_j E_j(y^k, r^k) \leq \max_j E_j(y^\ell, r^\ell)$. $\square$

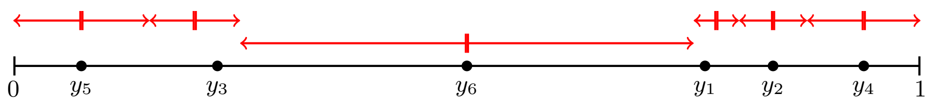


Figure 8: An example of the instance $(2, 3, 4, 5, 6)$.

Lemma 10 and the fact that the running time of the algorithm from [11] is $O(n^2 \log n)$ leads to the following result:

Theorem 9. *There exists an $O(n^3 \log n)$ time 2-approximation algorithm for* MaxFix*.*

Next we show that MaxFix is computationally hard.

Theorem 10. MaxFix *is NP-hard even if $x = (\frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2})$, for every $a \in (0, \infty)$ and $\alpha \geq 1$.*

Proof. Given a Partition instance $(s_1, \dots, s_n)$, 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 $(s_1, \dots, s_n) \in$ Partition if and only if there is a solution (y, r) such that $\max_i E_i(y, r) = a$.

Suppose that $(s_1, \dots, s_n) \in$ Partition, and let $I \subseteq \{1, \dots, n\}$ 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 $[\frac{3}{4}, 1]$. (See example in Figure 8.) This is possible, since $\sum_{i \in I} 2\rho_i = \sum_{i \in \{1,\dots,n\} \setminus I} 2\rho_i = \frac{1}{4}$. A sensor i, where $i \leq n$, needs less than $\frac{a}{2}$ energy to move, and at most $a4^\alpha \cdot \frac{1}{8}^\alpha = \frac{a}{2^\alpha} \leq \frac{a}{2}$ 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 sensor $n+1$ 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 $(s_1, \dots, s_n) \in$ Partition. □

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

Theorem 11. MaxFix *is strongly NP-hard, for every $a \in (0, \infty)$ and $\alpha \geq 1$.*

Proof. Given a 3-Partition instance $(s_1, \dots, s_n)$, 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 $i \leq n$, 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 $(s_1, \dots, s_n) \in$ 3-Partition if and only if there exists a solution (y, r) such that $\max_i E_i(y, r) = a$.

Now suppose that $(s_1, \dots, s_n) \in$ 3-Partition, and let $I_1, \dots, I_m \subseteq \{1, \dots, n\}$, 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 $k \in \{1, \dots, m-1\}$ 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 $k \in \{1, \dots, m\}$ to cover the interval $[\frac{2k-2}{2m-1}, \frac{2k-1}{2m-1}]$. This is possible, since

$$\sum_{i \in I_k} 2\rho_i = \sum_{i \in I_k} \frac{s_i}{(2m-1)Q} = \frac{1}{2m-1} .$$

Sensor i, where $i \le n$, needs less than $\frac{a}{2}$ energy to move, and

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

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.

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 $\sum_{i \in I_k} 2\rho_i \ge \frac{1}{2m-1}$. Hence $\sum_{i \in I_k} s_i \ge Q$, which means that $(s_1, \ldots, s_n) \in$ 3-PARTITION. □

5 Minimum Sum Energy with Fixed Radii

In this section we consider SUMFIX. Li et al. [16] 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. We also prove that SUMFIX with uniform radii can be approximated to within an additive approximation $\varepsilon > 0$, for any constant $\varepsilon > 0$. This algorithm is based on the non-swapping property and on placing the sensors on grid points. However, as opposed to the variable case, we cannot change radii, only locations, which is problematic when there is very little excess coverage. We cope with this issue by considering two solution types, small excess and large excess.

Czyzowicz et al. [13] showed that it is NP-hard to approximate the special case of SUMFIX_1 where $t = 0$ to within any constant c. We extend their approach and obtain 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 $\alpha > 1$, 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 $i \le n-1$, 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 $i \le n-1$, and cover the barrier without the help of sensor n. In this case the optimal value is $\frac{1}{2} + \frac{t}{2^\alpha(n-1)^{\alpha-1}}$.

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 12. 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 $\sum_i E_i(y, r) \le \frac{1}{2}$.

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 $\sum_i r_i = \sum_{i \in I} \frac{s_i}{S} = \frac{1}{2}$. Hence, (y, r) covers $[0, 1]$ and $\sum_i E_i(y, r) = \sum_i t r_i = \frac{1}{2}$ 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 13. 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 [18], SumFix$_0$ also has an FPTAS, for any $\alpha \geq 1$. □

5.2 Non-Zero Finite Friction & 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.

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 1. One only needs to notice that a switch can be made since $\rho_i = \rho_j$, for every $i \neq j$.

Lemma 11. *Any uniform* SumFix *instance has a non-swapping optimal solution.*

Given a SumFix instance with uniform radii, assume that $\rho = (R, R, \ldots, R)$, 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 12. *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*

1. *(y', r) is feasible,*
2. *$y'_i \in \mathcal{G}$, if i is active, and*
3. *$\sum_i E_i(y', r) \leq \sum_i E_i(y, r) + \varepsilon$.*

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 $y'_i \leq y_i + \frac{n}{m}$ and $y'_i \leq y'_{i-1} + 2R$, for $i > 1$, or $y'_1 \leq R$.

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,

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

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

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

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

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

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

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 12, 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 each arc leaving a vertex (i, j) is $a|x_i - \frac{j}{m}| + tR^\alpha$, and the length of each arc 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 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 following predetermined positions:

$$\mathcal{P} = \{R(2\ell - 1) : \ell \in \{1, \ldots, \chi\}\} \ ,$$

where $\chi = \lceil \frac{1}{2R} \rceil$.

Lemma 13. *Let $\varepsilon > 0$, and let (x, ρ, t) be a uniform* SumFix *instance that has a non-swapping optimal solution (y, r) with $X(r) \le \frac{\varepsilon}{an}$. There is a weakly non-swapping feasible solution (y', r') such that (i) there are χ active sensors deployed at $\mathcal{P}$, and (ii) $\sum_i E_i(y', r') \le \sum_i E_i(y, r) + \varepsilon$.*

Proof. Let $i_1, \ldots, i_k$ 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 $j \in \{2, \ldots, k\}$, 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 $i_q \ge i_j$ by δ_j rightwards. That is, $y'_{i_q} = y_{i_q} + \sum_{j \le q} \delta_j$. Inactive sensors simply remain at their initial positions. Next, we deactivate 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 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, $|y'_i - y_i| \le X(r)$, for every i, and therefore the additional movement cost is bounded by $anX(r) \le \varepsilon$. Hence $\sum_i E_i(y', r) \le \sum_i E_i(y, r) + \varepsilon$. $\square$

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

$$E(H) = \left\{((i, \ell), (i', \ell + 1)) : i < i'\right\} \ .$$

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 $(n + 1, \chi + 1)$ in G to solutions for the SumFix instance induced by Lemma 13. 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 14. *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 costs nothing.

Theorem 15. *There exists a polynomial time algorithm for* SumFix$_0$ *with uniform radii.*

(i) Long non swapping deployment.

(ii) Short non swapping deployment.

Figure 9: Non Swapping deployments.

5.3 Non-zero Finite Friction & General Radii

Next, we show that non-swapping does not hold in general for non-uniform instances.

Lemma 14. *There are* SumFix *instances in which an optimal solution must be swapping, for any $a > 0$ and $\alpha \geq 1$. Moreover, the ratio between the value of best non-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 was deployed at $y_2 \geq \frac{1}{2}$. In this case $y_i \geq y_2 \geq \frac{1}{2}$, for $i \geq 3$. See Figure 9i. 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. See Figure 9ii. 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. [13] proved that the special case of SumFix in which $t = 0$ cannot be approximated within any constant. We show that 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 very similar to the reduction from [13].

Theorem 16. 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 $B \in \mathbb{N}$.

Given a 3-Partition instance $(s_1, \ldots, s_n)$, where $n = 3m$ and $\sum_i s_i = mQ$, we construct a SumFix instance with $n + m\ell$ sensors, where $\ell = 8Bm(Q+1)n^{c+1}$, 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. □

6 Open Problems

We briefly mention some research directions and open problems.

We presented FPTASs for SumVar and MaxVar with non-zero finite friction. While it would not be surprising if the hardness result for SumVar with infinite friction [16] can be extended to SumVar with non-zero finite friction, the complexity of MaxVar remains open. Another open question is to come up with an approximation algorithm whose ratio is better than 2 or with a lower bound for MaxFix, for $a \in (0, \infty)$.

Another possible research direction is to consider a model in which sensors are allowed to move and to change their covering radii at any given time. Constant factor approximation algorithms were given for the case where radii can change at any time but moving is disallowed (i.e., $a = \infty$) for both variable and fixed radii [5, 21], but determining the complexity even for this special case is an open question.

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. Note that the hardness results of the one-dimentional setting still apply.

Acknowledgements

Amotz Bar-Noy was partially supported by the CUNY Institute for Computer Simulation, Stochastic Modeling and Optimization (CoSSMO). Dror Rawitz was partially supported by the Israel Science Foundation (grant no. 497/14).

References

[1] A. Agnetis, E. Grande, P. B. Mirchandani, and A. Pacifici. Covering a line segment with variable radius discs. *Computers & OR*, 36(5):1423–1436, 2009.

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

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

[4] A. Bar-Noy, B. Baumer, and D. Rawitz. Changing of the guards: Strip cover with duty cycling. *Theor. Comput. Sci.*, 610:135–148, 2016.

[5] A. Bar-Noy, B. Baumer, and D. Rawitz. Set it and forget it: Approximating the set once strip cover problem. *Algorithmica*, 79(2):368–386, 2017.

[6] A. Bar-Noy, T. Brown, M. P. Johnson, and O. Liu. Cheap or flexible sensor coverage. In *5th IEEE DCOSS*, volume 5516 of *LNCS*, pages 245–258, 2009.

[7] A. Bar-Noy, D. Rawitz, and P. Terlecky. Maximizing barrier coverage lifetime with mobile sensors. In *21st ESA*, volume 8125 of *LNCS*, pages 97–108, 2013.

[8] B. K. Bhattacharya, M. Burmester, Y. Hu, E. Kranakis, Q. Shi, and A. Wiese. Optimal movement of mobile sensors for barrier coverage of a planar region. *Theor. Comput. Sci.*, 410(52):5515–5528, 2009.

[9] A. L. Buchsbaum, A. Efrat, S. Jain, S. Venkatasubramanian, and K. Yi. Restricted strip covering and the sensor cover problem. In *18th SODA*, pages 1056–1063, 2007.

[10] E. W. Chambers, S. P. Fekete, H.-F. Hoffmann, D. Marinakis, J. S. B. Mitchell, S. Venkatesh, U. Stege, and S. Whitesides. Connecting a set of circles with minimum sum of radii. In *12th WADS*, volume 6844 of *LNCS*, pages 183–194, 2011.

[11] D. Z. Chen, Y. Gu, J. Li, and H. Wang. Algorithms on minimizing the maximum sensor movement for barrier coverage of a linear domain. *Discrete Comput. Geom.*, 50(2):374–408, 2013.

[12] J. Czyzowicz, E. Kranakis, D. Krizanc, I. Lambadaris, L. Narayanan, J. Opatrny, L. Stacho, J. Urrutia, and M. 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.

[13] J. Czyzowicz, E. Kranakis, D. Krizanc, I. Lambadaris, L. Narayanan, J. Opatrny, L. Stacho, J. Urrutia, and M. 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.

[14] E. D. Demaine, M. T. Hajiaghayi, H. Mahini, A. S. Sayedi-Roshkhar, S. O. Gharan, and M. Zadimoghaddam. Minimizing movement. *ACM Transactions on Algorithms*, 5(3), 2009.

[15] S. Dobrev, S. Durocher, M. E. Hesari, K. Georgiou, E. Kranakis, D. Krizanc, L. Narayanan, J. Opatrny, S. M. Shende, and J. Urrutia. Complexity of barrier coverage with relocatable sensors in the plane. *Theor. Comput. Sci.*, 579:64–73, 2015.

[16] H. Fan, M. Li, X. Sun, P. Wan, and Y. Zhao. Barrier coverage by sensors with adjustable ranges. *ACM Transactions on Sensor Networks*, 11(1):14:1–14:20, 2014.

[17] M. Gibson and K. Varadarajan. Decomposing coverings and the planar sensor cover problem. In *50th FOCS*, pages 159–168, 2009.

[18] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack problems*, pages 412–413. Springer Verlag, 2004.

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

[20] M. Mehrandish, L. Narayanan, and J. Opatrny. Minimizing the number of sensors moved on line barriers. In *WCNC*, pages 653–658, 2011.

[21] M. Poss and D. Rawitz. Maximizing barrier coverage lifetime with static sensors. In *13th ALGOSENSORS*, volume 10718 of *LNCS*, pages 198–210, 2017.

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

[23] P. Terlecky, B. Phelan, A. Bar-Noy, T. Brown, and D. Rawitz. Should I stay or should I go? Maximizing lifetime with relays. *Computer Networks*, 70:210–224, 2014.