Lifetime Maximization of Monitoring Sensor Networks

Peter Sanders and Dennis Schieferdecker[*]

Karlsruhe Institute of Technology (KIT), 76128 Karlsruhe, Germany
{sanders,schieferdecker}@kit.edu

Abstract. We study the problem of maximizing the lifetime of a sensor network assigned to monitor a given area. Our main result is a linear time dual approximation algorithm that comes arbitrarily close to the optimal solution if we additionally allow the sensing ranges to increase by a small factor. The best previous result is superlinear and has a logarithmic approximation ratio. We also provide the first proof of the NP completeness of this specific problem.

1 Introduction

Wireless sensor networks have become a prominent research topic in recent years. Their unique structure and limitations provide new and fascinating challenges. A network consists of a union of small nodes that are equipped with sensing, communication and processing capabilities. The nodes are usually only battery powered with no means of recharging. Therefore, energy is a highly limited resource and energy consumption becomes a critical factor in this context. On the other hand, the sensor nodes themselves are cheap and available in abundance. This fact is exploited to counter their inherent limitations. Wireless sensor networks can be used for a multitude of monitoring tasks. Thus, there has been a lot of research on monitoring problems.

In this paper, we consider the question of how to maximize the lifetime of a monitoring sensor network when each node has a nonrechargeable battery with limited capacity. The basic idea is to switch on only subsets of nodes at a time while the remaining nodes are in an energy saving sleep-mode. Our results apply to several variants of this question. In the *target monitoring problem*, we are given a set of points in the plane that need to be monitored, i.e., during the entire lifetime of the network and for every target t, there must be an active node in its sensing range. In the *area monitoring problem*, every point in a designated area must be monitored. During most of this paper we will assume a uniform fixed sensing range for all the nodes. While this assumption is common in theoretical papers, it is unrealistic in practice. However, we believe that our techniques can be adapted for more general shapes of sensing areas. Moreover, sensing ranges do not come into play at all when we reinterpret the area monitoring problem as the problem to regularly take samples of point-measurements at a given minimum *resolution*, i.e., every point of the monitoring area is at most a given distance away from a sample point. This combination of a simple geometric model and a realistic interpretation was one of the main motivations for us to look at this particular problem. Area monitoring was first studied by Berman et al. [1,2] and a

[*] Partially supported by the German Research Foundation (DFG) within the Research Training Group GRK 1194 "Self-organizing Sensor-Actuator-Networks".

superlinear algorithm for finding efficient schedules with a logarithmic approximation ratio was presented. Our main result is a linear time dual approximation algorithm that comes within a factor $(1 - \epsilon)$ of the optimum if we additionally allow the sensing range to grow by a factor $(1 + \delta)$ for arbitrary constants ϵ and δ. Our focus is the approximability of this problem. Thus, we are initially content with a sequential approximation algorithm which can be later used to evaluate distributed variants.

We give more related work in Section 2, define the model in Section 3 and then present the approximation algorithm in Section 4 and discuss an adaptation to target monitoring in Section 5. Since previous NP-hardness results only apply to a variant of the target monitoring problem without geometric structure, we also provide the missing NP-hardness proof in Section 6. A conclusion and outlook completes the paper (Section 7).

2 Related Work

There has been a lot of activity in the field of wireless sensor networks over the last years and, thus, there already exist many contributions dealing with the optimization of energy-constraint networks designed for specific tasks. Monitoring tasks are a particularly large field of applications for wireless sensor networks. Thus, determining schedules that maximize the lifetime of these networks has been the focus of a lot of work.

In [3], Cardei and Wu first classify this problem as area monitoring and target monitoring. Later in [1], Berman et al. argue that area monitoring can be reduced to target monitoring of $O(n^2)$ targets for a class of sensor models, given a network of n sensors. Thus, it is sufficient to regard only the latter problem.

Slijepcevic and Potkonjak study the problem of monitoring a set of targets in [4]. Their approach is based on disjoint sets of sensor nodes each of which covers all of the targets. They further simplify the problem by assuming that each node has the same initial energy. Thus, in an optimal solution given these constraints, each node is used in at most one set and each set is active for the same amount of time. They state that set-k-cover is a generalization of their problem and provide a heuristic for set-k-cover which they use to solve their problem. Unfortunately, they do not provide asymptotic runtimes or approximation guarantees.

A more general model is analyzed by Cardei et al. in [5]. Here, sensor nodes are allowed to be part of more than one set but their initial battery capacity is still assumed to be equal. They propose two heuristics, one using a linear programming (LP) approach and the other being a greedy algorithm. The former has a complexity of $O(n^3p^3)$ with n the number of sensor nodes and p an upper bound on the number of sets. The latter takes $O(dm^2n)$ with n the number of sensor nodes, m the number of targets and d the minimum number of nodes covering a target. They do not give approximation guarantees. Cardei et al. also provide a proof of NP-completeness, reducing from 3-SAT. This proof is subsequently cited by most works on this subject even though it only applies to the more general case with no geometric structure.

In [1,2] Berman et al. discuss the problem of area monitoring without enforcing constraints on the node sets or battery capacities as the previous authors did. They outline an efficient data structure and algorithm to transform an area monitoring task to a target monitoring task. After giving a LP formulation of the problem, they provide an approximation algorithm using the Garg-Könemann algorithm [6] as basis. Their proposed algorithm has a runtime complexity of $O(1/\epsilon^2 n \log n \cdot T)$ with n the number of sensor nodes and T the runtime of an auxiliary min set cover algorithm. The approximation guarantee is $(1+\epsilon)(1+2\log n)$. They also provide LP formulations for two additional models, one only requiring partial coverage of the area and the other taking into account communication costs. Both are solvable with the same setup as their initial model.

A variation of the previous area monitoring problem is considered by Dhawan et al. in [7]. They generalize upon the initial model of Berman et al. by introducing variable sensor radii that are directly linked to their energy consumption over time. They provide an approximation algorithm along the same line as Berman et al., introducing a new heuristic for the Garg-Könemann subroutine. Their algorithm has an $(1+\epsilon)(1+\log m)$ approximation guarantee, m being the number of targets to cover and runs in $O(1/\epsilon^2 n \log n \cdot T)$ with n the number of sensor nodes and T the unspecified runtime of the subroutine.

The area monitoring problem introduced by Berman et al. was picked up by Gu et al. in 2007 [8]. Even though they only model target monitoring, this approach leads to the same LP formulation. They extend the model by demanding the existence of a data gathering tree with appropriate energy costs for communication. They propose a column generation approach to exactly solve the LP problem and provide an algorithm to generate good initial solutions. This is a well-known and adequate technique to manage large-scale LP problems in practice. No runtime guarantees are provided since the problem is still NP-hard.

In 2009, Luo et al. revisited the problem as one example to demonstrate their technique for solving large LPs [9]. As Gu et al. , they apply a column generation technique as exact solver. Their main contribution is a new way to generate new columns that yields a speed-up of more than one order of magnitude. Their model also offers rudimental support for incorporating connectivity costs into the energy consumption of the sensor nodes. As before, no theoretical runtime guarantees are given, only simulational values.

3 Model and Problem Definition

Sensor Network Model. This paper considers a sensor network consisting of n nodes $\{S_1, \dots, S_n\} = \mathcal{S}$. Each node $S_i,\ i \in \{1, \dots, n\}$ consists of a triple (x_i, y_i, b_i), denoting its position in the plane (x_i, y_i) and its battery capacity b_i. Nodes are placed arbitrarily in the plane. We assume that nodes only consume energy while actively monitoring their surroundings and that energy consumption is constant over time and independent of small changes in the sensing range. We further assume that communication and processing costs are proportional to monitoring costs[1]. All quantities are normalized, i.e. sensor nodes consume one unit of energy per unit of time and battery capacity is given in units of energy.

Covers. Let A be a connected area with a description complexity linear in $|\mathcal{S}|$. Given this area A and a sensing range R, a set of sensor nodes $C \subseteq \mathcal{S}$ is called a cover, if area A is

[1] Thus, we can incorporate communication and processing costs implicitly by using effective monitoring costs that are a linear combination of all three types of costs.

contained in the union of disks of radius R centered at each sensor node. In particular, the set of all sensor nodes $\mathcal{S}$ is a cover if any cover exists. The set of all covers is called $\mathcal{C}$.

Problem Definition. Consider a sensor network $\mathcal{S}$, an area A and a sensing range R. What is the maximum time T area A can be monitored by these sensors before this becomes impossible due to node failure? In particular, we want to determine a set of m covers $\underline{C} = \{C_1, \ldots, C_m\} \subseteq \mathcal{C}$ and a corresponding set of durations $\underline{t} = \{t_1, \ldots, t_m\}$ that maximizes the lifetime $T = \sum_{j=1}^{m} t_j$ while respecting the limited battery capacities, i.e.

$$\sum_{i:S_j \in C_i} t_i \leq b_j \; \forall \; S_j \in \mathcal{S}. \tag{1}$$

We refer to this problem as *sensor network lifetime problem (SNLP)* [1].

A tuple $(\underline{C}, \underline{t})$ is also called a *schedule*. Applying a schedule implies activating each cover iteratively for the corresponding duration. If a cover is active, all of its sensor nodes are active and all other ones are sleeping.

4 Approximation Algorithm

We introduce two techniques for solving simpler variants of the original problem. First, we consider discretizing sensor positions by snapping them to a grid[2]. Secondly, we consider solving subproblems restrained to small areas of the plane. Subsequently, we combine both variants in our approximation algorithm and prove its approximation guarantee and asymptotic runtime.

A problem instance is denoted by a triple $(\mathcal{S}, A, R)$ with $\mathcal{S}$ the set of sensor nodes, A the area to be monitored and R the sensing range of each node. A solution of problem $(\mathcal{S}, A, R)$ consists of a tuple $(\underline{C}, \underline{t})$ with $\underline{C}$ a sequence of covers and $\underline{t}$ a sequence of corresponding durations, i.e. t_i denotes the duration of cover C_i. A solution is called feasible if condition (1) is fulfilled. We abbreviate the lifetime of a solution of problem $(\mathcal{S}, A, R)$ by $T\langle \mathcal{S}, A, R\rangle$. The lifetime of an optimal solution is denoted by $T_{\text{opt}}\langle \mathcal{S}, A, R\rangle$, respectively.

Subsequently, we usually omit mentioning area A for brevity. We also normalize sensing ranges to 1 w.l.o.g. .

4.1 Discretizing Positions

Consider a modified problem with sensor nodes restricted to positions on a grid. Given an algorithm $\mathcal{A}$ that computes f-approximate solutions for this special problem, we can find an approximation of the original problem with a small computational overhead, if we also allow the sensing radii to increase by a small amount. Lemma 1 summarizes this claim.

[2] A grid is a set of points $\{(\alpha \cdot i, \alpha \cdot j) | i, j \in \mathbb{Z}\}$ with $\alpha \in \mathbb{R}$ the width of the grid.

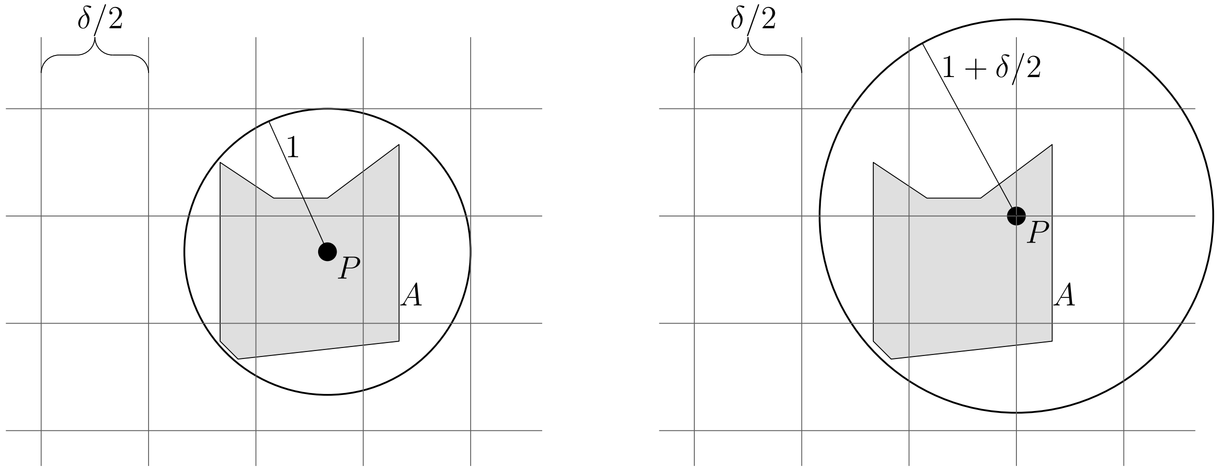

Fig. 1. *Left.* Sensor node P with sensing range 1 covers area A. A grid of width $\delta/2$ is plotted. *Right.* Sensor node P has been moved to a grid position and its sensing range increased to $1+\delta/2$. It still covers area A.

Algorithm 4.1: Grid-Relaxation

```
input:  parameter $\delta \in [0,1]$, sensor nodes $\mathcal{S}$, area $A$, sensing range $R$
output:  set of covers $\underline{C}$, set of corresponding durations $\underline{t}$

1. Define a grid of width $\delta/2$.
2. Move every node in $\mathcal{S}$ to the closest point on the grid $\rightarrow \tilde{\mathcal{S}}$.
3. Use algorithm $\mathcal{A}$ to solve $(\tilde{\mathcal{S}}, 1+\delta/2) \rightarrow (\underline{C}, \underline{t})$.
4. Return $(\underline{C}, \underline{t})$.
```

Lemma 1. *Let $\delta \in [0,1]$. Algorithm 4.1 yields a feasible solution of problem $(\mathcal{S}, 1+\delta)$ with lifetime $T\langle\mathcal{S}, 1+\delta\rangle \geq f \cdot T_{\text{opt}}\langle\mathcal{S}, 1\rangle$. The running time outside algorithm $\mathcal{A}$ is $O(|\mathcal{S}|)$.*

Proof. Correctness: Consider the original problem $(S, 1)$. Moving all nodes in $\mathcal{S}$ to the closest point on a grid of width $\delta/2$ yields $\tilde{\mathcal{S}}$. Each node is shifted by at most $\frac{\sqrt{2}}{2} \cdot \delta/2 < \delta/2$. Note that if we additionally increase the sensing range by a factor $1+\delta/2$, a cover with respect to $(S, 1)$ is also a cover with respect to $(\tilde{\mathcal{S}}, 1+\delta/2)$ as depicted in Figure 1. Thus, $T_{\text{opt}}\langle\tilde{\mathcal{S}}, 1+\delta/2\rangle \geq T_{\text{opt}}\langle\mathcal{S}, 1\rangle$ and algorithm $\mathcal{A}$ computes a solution of $(\tilde{\mathcal{S}}, 1+\delta/2)$ with lifetime $T\langle\tilde{\mathcal{S}}, 1+\delta/2\rangle \geq f \cdot T_{\text{opt}}\langle\tilde{\mathcal{S}}, 1+\delta/2\rangle \geq f \cdot T_{\text{opt}}\langle\mathcal{S}, 1\rangle$. A solution to $(\tilde{\mathcal{S}}, 1+\delta/2)$ is also a solution to $(\mathcal{S}, 1+\delta)$ by the same argument as above for $(S, 1)$ and $(\tilde{\mathcal{S}}, 1+\delta/2)$. Thus the solution provided by algorithm $\mathcal{A}$ is also a feasible solution for $(\mathcal{S}, 1+\delta)$ with lifetime $T\langle\mathcal{S}, 1+\delta\rangle \geq f \cdot T_{\text{opt}}\langle\mathcal{S}, 1\rangle$.

Computational Overhead: $O(|\mathcal{S}|)$ is the time required to relocate all sensor nodes to their closest grid point. □

Note that this approach assumes that small changes in sensing range R have no impact on the energy consumption of the sensor nodes. Also note that the applied approach of rounding continuous values to some discrete numbers is a common technique found in many other approximation schemes.

4.2 Area Partitioning

Consider the original problem $(\mathcal{S}, A, 1)$ and a partition of the plane into axis-aligned squares of width k. If we confine our problem to a single square T of this partition, we only have to consider covering area $A \cap T$ with the subset of sensor nodes in $\mathcal{S}$ that lie within T or less than one sensing range outside.

Given an algorithm $\mathcal{A}$ that computes f-approximate solutions for problems restricted to small squared areas, we can compute a solution for each square of the partition and combine them to a solution of the whole problem[3]. Unfortunately, such a solution does not have to be feasible. It is possible that a sensor node has to be considered for the coverage of more than one square and, thus, the node could require more than its available capacity to fulfill its assignments.

Now, consider a set of k partitions $\underline{\mathcal{T}} = \{\mathcal{T}^i\}$ with $i \in \{0, \ldots, k-1\} = \mathbb{Z}_k$. All partitions consist of axis-aligned squares of width k. Partition $\mathcal{T}^{i+1}$ is generated from partition $\mathcal{T}^i$ by translation to the top and to the right by 1 for $i \in \mathbb{Z}_k$ (see Figure 2). Observe that a sensor node has to be considered for the coverage of at most 4 squares. But the case of more than 1 square only occurs in at most two partitions as shown in Figure 2.

A solution $(\underline{C}^i, \underline{t}^i)$ of problem $(\mathcal{S}, 1)$ constructed from the solutions of each square of partition $\mathcal{T}^i,\ i \in \mathbb{Z}_k$ as described above, satisfies the following two conditions

$$T\langle \mathcal{S}, 1\rangle^i = \sum_{j : C^i_j \in \underline{C}^i} t^i_j \geq f \cdot T_{\text{opt}}\langle \mathcal{S}, 1\rangle, \tag{2}$$

$$\forall S_n \in \mathcal{S} : \sum_{j : S_n \in C^i_j} t^i_j \leq \begin{cases} 4 \cdot b_n & \text{node } S_n \text{ needed by more than one square,} \\ 1 \cdot b_n & \text{otherwise.} \end{cases} \tag{3}$$

The lifetime $T\langle \mathcal{S}, 1\rangle^i$ of each solution is the sum over the durations $t^i_j \in \underline{t}^i$ of its covers $C^i_j \in \underline{C}^i$. It is optimal up to a factor of f. Similarly, the active time of each sensor node $S_n \in S$ is the sum over the durations $t^i_j \in \underline{t}^i$ of each cover $C^i_j \in \underline{C}^i$ with $S_n \in C^i_j$, i.e. of the covers containing node S_n. Its active time is at most 4 times its battery capacity.

Now, we can compute k (infeasible) solutions $(\underline{C}^i, \underline{t}^i)$ of problem $(\mathcal{S}, 1)$ with respect to each partition $\mathcal{T}^i,\ i \in \mathbb{Z}_k$ by combining the solutions of algorithm $\mathcal{A}$ for each square of the respective partition as described above. We obtain a feasible solution by concatenating these solutions and scaling the durations of each cover, i.e. $(\cup_{i \in \mathbb{Z}_k} \underline{C}^i, c \cdot \cup_{i \in \mathbb{Z}_k} \underline{t}^i)$. Here, $c \cdot \cup_{i \in \mathbb{Z}_k} \underline{t}^i$ denotes a union of all $\underline{t}^i$ with each element $t^i_j \in \underline{t}^i$ multiplied by c. Lemma 2 summarizes this claim.

Lemma 2. *Let $k = \lceil 10/\epsilon \rceil$ with $\epsilon \in (0, 1]$. Given an (infeasible) solution $(\underline{C}^i, \underline{t}^i)$ of problem $(\mathcal{S}, 1)$ with respect to each partition $\mathcal{T}^i, i \in \mathbb{Z}_k$ that has lifetime $T\langle \mathcal{S}, 1\rangle^i \geq f \cdot T_{\text{opt}}\langle \mathcal{S}, 1\rangle$, a union of the solutions of all partitions as defined above and with $c = (1-\epsilon)/k$ is a feasible solution of problem $(\mathcal{S}, 1)$ with lifetime $T\langle \mathcal{S}, 1\rangle \geq f \cdot (1-\epsilon) \cdot T_{\text{opt}}\langle \mathcal{S}, 1\rangle$.*

[3] The schedules of all squares can be run concurrently and independently. A node is active if the schedule of any square requires it to be active.

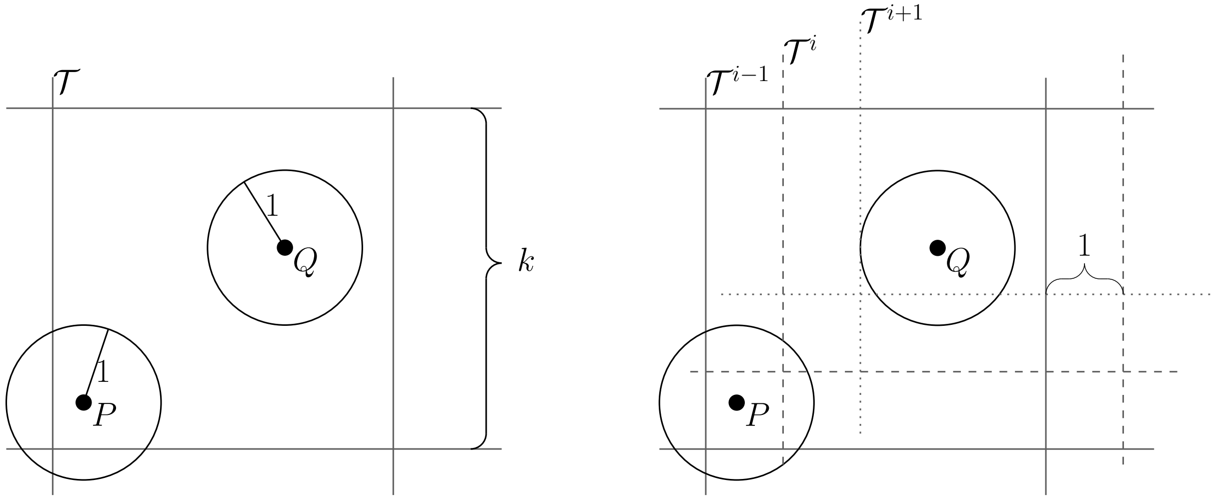

Fig. 2. *Left.* One partition $\mathcal{T}$ is depicted. Sensor node P has to be considered for the coverage of 4 squares, i.e. the circle around P with radius equal to one sensing range overlaps 4 squares. *Right.* Three subsequent partitions $\mathcal{T}^{i-1}$, $\mathcal{T}^{i}$ and $\mathcal{T}^{i+1}$ are shown. In partitions $\mathcal{T}^{i-1}$ and $\mathcal{T}^{i}$ node P overlaps 4 squares and Q only one. In partition $\mathcal{T}^{i+1}$ P overlaps only one square and Q two squares.

Proof. Lifetime: The lifetime $T\langle\mathcal{S},1\rangle$ of the union is the sum of the lifetimes $T\langle\mathcal{S},1\rangle^i$ for each partition $\mathcal{T}^i$, scaled by $(1-\epsilon)/k$. This sum is bounded by

$$T\langle\mathcal{S},1\rangle = \frac{1-\epsilon}{k}\sum_{i\in\mathbb{Z}_k} T\langle\mathcal{S},1\rangle^i \geq \frac{1-\epsilon}{k}\sum_{i\in\mathbb{Z}_k} f\cdot T_{\text{opt}}\langle\mathcal{S},1\rangle = f\cdot(1-\epsilon)\cdot T_{\text{opt}}\langle\mathcal{S},1\rangle$$

as claimed in the lemma. The inequality follows by Equation (2).

Feasibility: Each sensor node $S_n \in \mathcal{S}$ is active for the sum of the durations t_j^i of all covers $C_j^i \in \underline{C}^i$ with $S_n \in C_j^i$ over all partitions $\mathcal{T}^i$ with $i \in \mathbb{Z}_k$. This sum is bounded by

$$\frac{1-\epsilon}{k}\sum_{i\in\mathbb{Z}_k}\sum_{j:S_n\in C_j^i} t_j^i \leq \frac{1-\epsilon}{k}\Big((k-2)\cdot 1 + 2\cdot 4\Big)\cdot b_n \leq b_n .$$

The first inequality follows by Equation (3) and the fact that a sensor node is only required for the coverage of more than one square in at most two partitions. The second inequality follows due to our choice of k. Thus, the active time of each sensor nodes S_n is bound by b_n and, therefore, the solution is feasible. □

A similar approach, using shifting partitions to devise a polynomial approximation scheme for numerous NP-complete geometric covering and packing problems, has first been proposed by Hochbaum and Maas in [10].

4.3 Algorithm

After introducing these two techniques, we present our approximation algorithm. The general approach is also depicted by Algorithm 4.3. We assume the availability of an algorithm $\mathcal{A}$ that computes f-approximate solutions for problem instances $(\mathcal{S}, 1)$ confined to a small squared area and with sensor nodes restricted to positions on a grid with a runtime complexity of $g_{\mathcal{A}}(|\mathcal{S}|)$. Note that this algorithm combines the restrictions of both algorithms assumed in the previous sections.

Now, consider a general problem $(\mathcal{S}, 1)$. We construct a feasible solution $(\underline{C}, \underline{t})$ similar to Section 4.2 by using algorithm $\mathcal{A}$ to compute solutions for squares of partitions of the plane and combining them. But since algorithm $\mathcal{A}$ also requires the sensor nodes to lie on a grid, we can only compute these solutions, if we allow the sensing radii to grow by a small amount, similar to Section 4.1. The whole algorithm yields a solution that comes arbitrarily close to the optimal solution if we additionally allow the sensing ranges to increase by a small factor and runs in pseudo-linear time. Theorem 1 summarizes these claims.

```
Algorithm 4.3: Approximation Algorithm
input:  parameter $\delta \in [0, 1], \epsilon \in (0, 1]$, sensor nodes $\mathcal{S}$, area $A$, sensing range $R$
output:  set of covers $\underline{C}$, set of corresponding durations $\underline{t}$

1. Define a grid of width $\delta/2$.
2. Move every node in $\mathcal{S}$ to the closest point on the grid $\to \tilde{\mathcal{S}}$.
3. Define $k$ partitions $\mathcal{T}^i$ of the plane into axis-aligned squares of width $k$ with $\mathcal{T}^{i+1}$
   generated from partition $\mathcal{T}^i$ by translation to the top and to the right by 1 for $i \in \mathbb{Z}_k$.
4. For each partition $\mathcal{T}^i$,
   – use algorithm $\mathcal{A}$ to solve $(\tilde{\mathcal{S}}, 1 + \delta/2)$ confined to each square of $\mathcal{T}^i$,
   – combine these partial solutions $\to (\underline{C}^i, \underline{t}^i)$
5. Unite the $k$ solutions to $(\underline{C}, \underline{t}) = (\cup_{i \in \mathbb{Z}_k} \underline{C}^i, (1 - \epsilon)/k \cdot \cup_{i \in \mathbb{Z}_k} \underline{t}^i)$.
6. Return $(\underline{C}, \underline{t})$.
```

Theorem 1. *Let $\delta \in [0, 1]$ and $k = \lceil 10/\epsilon \rceil$ with $\epsilon \in (0, 1]$. Algorithm 4.3 computes a feasible solution $(\underline{C}, \underline{t})$ of problem $(\mathcal{S}, 1 + \delta)$ with lifetime*

$$T\langle \mathcal{S}, 1 + \delta \rangle \geq (1 - \epsilon) \cdot f \cdot T_{\text{opt}}\langle \mathcal{S}, 1 \rangle.$$

The runtime complexity of Algorithm 4.3 is pseudo-polynomially bounded by

$$O\Big(|\mathcal{S}| + \epsilon|\mathcal{S}| \cdot g_{\mathcal{A}}(O({}^1\!/_{\delta^2\epsilon^2}))\Big) = O(|\mathcal{S}|)$$

if area A can be fully contained in a number of squares of width k linear in $|\mathcal{S}|$.

Proof. Feasibility: The feasibility of solution $(\underline{C}, \underline{t})$ follows directly from the proof of correctness in Lemma 1 and the proof of feasibility in Lemma 2.

Approximation Guarantee: By applying a grid discretization, we obtain an approximation guarantee $T\langle \mathcal{S}, 1 + \delta \rangle \geq f \cdot T_{\text{opt}}\langle \mathcal{S}, 1 \rangle$ for the solution of each square according to Lemma 1. As stated by Equation (2), the same approximation guarantee holds for each solution $(\underline{C}^i, \underline{t}^i)$ of $(\mathcal{S}, 1 + \delta)$ with $(\underline{C}^i, \underline{t}^i)$ obtained by combining the solutions of all squares of partition

$\mathcal{T}^i$, $i \in \mathbb{Z}_k$. Combining these k solutions as described above, yields an additional factor $(1-\epsilon)$ in the approximation guarantee according to Lemma 2. The claimed lifetime follows.

Runtime: According to Lemma 1, there is an additive overhead of $O(|\mathcal{S}|)$ when using a grid discretization. The solution for each square can be found in $g_{\mathcal{A}}(O(1/\delta^2\epsilon^2))$ since each square only contains $O(1/\delta^2\epsilon^2)$ distinct grid points and, thus, at most as many sensor nodes. The number of sensor nodes to be considered for covering each square is higher by at most a constant factor since only nodes closer to the square than one sensing range have to be considered in addition. There are k partitions and area A is contained in at most $O(\epsilon^2|\mathcal{S}|)$ squares. Thus, a solution for $k \cdot O(\epsilon^2|\mathcal{S}|) = O(\epsilon|\mathcal{S}|)$ squares has to be found.

The maximum number of squares can be assessed by regarding the minimum requirements for covering area A. A disk cover requires at least $2/\sqrt{27}$ disks of radius 1 to cover the unit square [11]. Thus, we need $2k^2/\sqrt{27}$ sensor nodes to cover a square of width k. If a problem contains $|\mathcal{S}|$ nodes, the number of squares that can be covered is bounded by $2|\mathcal{S}|k^2/\sqrt{27} = O(\epsilon^2|\mathcal{S}|)$. □

Note that if δ, ϵ are constant, the input of algorithm $\mathcal{A}$ is of constant size. Thus, any implementation of algorithm $\mathcal{A}$ would only contribute a constant factor to the runtime of Algorithm 4.3 with respect to $|\mathcal{S}|$. Naturally, one would try to use an efficient implementation or one with a good approximation ratio f. For example, even an exact solver could be used. Also note that we did not optimize the numerical constants of the grid width and the number of partitions.

5 Adaptation to Target Monitoring

Berman et al. stated in [1,2] that the area monitoring problem can be reduced to a target monitoring problem for convex sensing ranges. Thus, many subsequent studies only concern themselves with target monitoring. Our approximation algorithm was originally designed for the problem of monitoring an arbitrary area, but it can be easily adapted to monitor a set of discrete targets.

In principle, Algorithm 4.3 is already sufficient for this task. We only have to replace algorithm $\mathcal{A}$ by another one that solves the target monitoring problem on a squared area with sensor nodes placed on a grid. However, the runtime of this algorithm also depends on the number of targets. This can be resolved by also discretizing target positions. Multiple targets occupying the same position can be regarded as one for the purpose of covering. If we relocate them to the same grid positions as the sensor nodes, there are at most $O(1/\delta^2\epsilon^2)$ targets in each square. Thus, the runtime no longer depends on the number of targets.

Note that for the proofs to remain correct, $T_{\text{opt}}\langle\mathcal{S}, 1-\delta\rangle$ has to be replaced by $T_{\text{opt}}\langle\mathcal{S}, 1-2\delta\rangle$ since now both, sensor nodes and targets, are relocated.

6 Proof of NP-Completeness

In this section, we provide a proof of NP-completeness of SNLP by exploiting certain properties of linear programs (LPs). We are convinced that this novel proof is necessary since the currently accepted proof by Cardei et al. in [5] does not take into account the geometric structure of the problem. By this omission a much more difficult problem was considered.

We first introduce a geometric problem and prove its NP-completeness. Then, we formulate SNLP as linear program and prove that it is equally hard to solve than a related LP problem. Finally, we show that the related problem and the geometric problem are equivalent.

Theorem 2. *The Minimum Dominating Set (MDS) problem on unit disk graphs is NP-hard [12]. The same is true for weighted MDS on unit disk graphs.*

Lemma 3. *Given a unit disk graph $G = (V, E)$ and a unit disk embedding of V, a set $D \subseteq V$ is a dominating set iff the the set of unit disks centered at D is a cover of V.*

Proof. First, let the set of unit disks centered at D be a cover of V. Thus, there is at least one node $d \in D$ for each node $v \in V$ with a distance of at most 1. By definition of the unit disk graph, there exist an edge (d, v) or $d \equiv v$. Therefore, D is a dominating set of G.

Now, let D be a dominating set of G. Thus, every node $v \in V$ is either in D or neighboring to a node $d \in D$. By definition of the unit disk graph, the distance between v and d is at most 1. Therefore, the set of unit disks centered at D covers N. □

Definition 1. Minimum-Cost Geometric Disc Coverage (MCGDC). *Given a set of points P in the plane and a set of unit disks U with associated costs $c_{1,\ldots,|U|}$. Can all points in P be covered by a subset $D \subseteq U$ with total cost $\sum_{i \in D} c_i \le C$?*

Theorem 3. *MCGDC is NP-complete.*

Proof. NP-hardness is proved by reduction of the decision variant of weighted MDS. Nodes V of the MDS input graph $G = (V, E)$ become centers of the unit disks U in MCGDC and are also used as point set P. Then, the geometric cover of U computed by MCGDC is also a dominating set of V, as shown by Lemma 3. Since costs in MCGDC correspond to weights in MDS, a solution of MCGDC is also a solution of the decision variant of weighted MDS. NP-completeness follows trivially. □

Definition 2. Linear Programs. *SNLP can be formulated as linear program $T_{\mathrm{opt}}\langle \mathcal{S}, 1\rangle = \max\{\underline{1} \cdot \underline{t} | \mathbf{M}\underline{t} \le \underline{b}\}$ with $\underline{1} = (1, \ldots, 1) \in \mathbb{R}^{|\mathcal{S}|}$ and $\mathbf{M}_{i,j} = 1$ iff $S_i \in C_j$, $\mathbf{M}_{i,j} = 0$ otherwise. Note that each column of matrix $\mathbf{M}$ corresponds to a cover $C_j \in \mathcal{C}$.*

The dual linear program $\min\{\underline{b}\ \underline{w} | \mathbf{M}^{\mathrm{T}}\underline{w} \ge \underline{1}\}$ can be read as finding minimal costs for all sensor nodes so that the cost of each cover is at least 1. We interpret the elements of $\underline{w} = (w_1, \ldots, w_{|\mathcal{S}|})$ as costs for using the respective nodes, i.e. w_i is the cost of node $S_i \in \mathcal{S}$.

Definition 3. Separation Problem. *Given an LP, verify whether a candidate solution meets all constraints of the LP, and if not, provide a counter-example.*

Theorem 4. *The separation problem associated with an LP is polynomially solvable iff the corresponding LP is polynomially solvable [13].*

Lemma 4. *The separation problem associated with the dual of SNLP is equivalent to MCGDC.*

Proof. The separation problem associated with the dual of SNLP can be formulated as follows: Given a candidate solution consisting of sensor nodes $\mathcal{S}$ and associated weights $\underline{w}$, decide whether there exists a cover of area A using nodes with a total cost of less than 1. If true, the candidate solution does not fulfill all constraints. Since covering an area is equal to covering a set of points [1,2], this problem is equal to MCGDC with costs $c_i = w_i$ and $C = 1$. □

Theorem 5. *The Sensor Network Lifetime Problem (SNLP) is NP-complete.*

Proof. Theorem 3 states that MCGDC is NP-complete. Since MCGDC and the separation problem associated with the dual of SNLP are equivalent (Lemma 4), the latter is also NP-complete. Thus, the dual of SNLP is also NP-complete by Theorem 4. Since a solution of the dual problem of an LP can be transformed into a solution of the LP in polynomial time [14], SNLP is NP-complete. □

7 Conclusion

Even though our theoretical results are very strong, they only mark a first step. There are still many ways to enhance the underlying model, e.g. considering non-uniform sensing ranges is of great interest to us. Thus, we are currently looking into basing our algorithm more firmly in the field of computational geometry. By proving its approximation and runtime guarantees for general low-dimensional metrics, the inclusion of obstacles and variable sensing ranges into our model should become easy. Also, a generalization to higher dimensions and angular dependent sensor ranges would be possible, as well removing of the dependence on squared partitions which would provide more flexibility.

Furthermore, we intend to implement and to experimentally evaluate our approximation algorithm using an exact solver based on column generation as algorithm $\mathcal{A}$ for computing the small subproblems. This task will most likely encompass several interesting aspects of the field of algorithm engineering.

Since we are considering a sensor network, a distributed implementation of our algorithm is a natural extension. Its structure is already well-suited for a parallel implementation with the partition of the considered area into independent squares. This could be taken even further, e.g. with all sensor nodes organizing themselves independently into these partitions. Thus, we are also interested in looking into distributed variants of our approximation algorithm.

Finally, we hope that our novel proof of NP-completeness will be spread in the community and, henceforth, be referenced as correct proof for the area monitoring problem (SNLP).

Acknowledgements. We would like to thank Sanjeev Arora and David Steurer for providing key ideas of this work and the enlightening insight into approximation algorithms and computational geometry they provided.

References

1. Berman, P., Calinescu, G., Shah, C., Zelikovsky, A.: Power efficient monitoring management in sensor networks. In: Proceedings of the Wireless Communications and Networking Conference 2004. Volume 4., IEEE Computer Society Press (March 2004) 2329–2334
2. Berman, P., Calinescu, G., Shah, C., Zelikovsky, A.: Efficient energy management in sensor networks. In Pan, Y., Xiao, Y., eds.: Ad Hoc and Sensor Networks. Volume 2 of Wireless Networks and Mobile Computing. Nova Science Publisher (2005) 71–90
3. Cardei, M., Wu, W.: Energy-efficient coverage problems in wireless ad-hoc sensor networks. Computer Communications **29**(4) (2006) 413–420
4. Slijepcevic, S., Potkonjak, M.: Power efficient organization of wireless sensor networks. In: Proceedings of the IEEE International Conference on Communications 2001. (June 2001) 472–476
5. Cardei, M., Thai, M.T., Li, Y., Wu, W.: Energy-Efficient Target Coverage in Wireless Sensor Networks. In: Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom). Volume 4., IEEE Computer Society Press (March 2005) 1976–1984
6. Garg, N., Könemann, J.: Faster and Simpler Algorithms for Multicommodity Flow and other Fractional Packing Problems. In: Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS'98), IEEE Computer Society (1998) 300–309
7. Dhawan, A., Vu, C.T., Zelikovsky, A., Li, Y., Prasad, S.K.: Maximum Lifetime of Sensor Networks with Adjustable Sensing Range. In: Proceedings of the International Workshop on Self-Assembling Wireless Networks, 2006, IEEE Computer Society (2006)
8. Gu, Y., Liu, H., Zhao, B.: Joint Scheduling and Routing for Lifetime Elongation in Surveillance Sensor Networks. In: Proceedings of the 2nd IEEE Asia-Pacific Service Computing Conference (APSCC). (2007) 304–311
9. Luo, J., Girard, A., Rosenberg, C.: Efficient Algorithms to Solve a Class of Resource Allocation Problems in Large Wireless Networks. In: 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt). (June 2009)
10. Hochbaum, D.S., Maass, W.: Approximation schemes for covering and packing problems in image processing and VLSI. Journal of the ACM **32**(1) (January 1985) 130–136
11. Williams, R.: The Geometrical Foundation of Natural Structure: A Source Book of Design. The Dover Publications (June 1979)
12. Masuyama, S., Ibaraki, T., Hasegawa, T.: The Computational Complexity of the m-Center Problems on the Plane. IEICE Transactions Japan **E64**(2) (1981) 57–64
13. Grötschel, M., Lovász, L., Schrijver, A.: The ellipsoid method and its consequences in combinatorial optimization. Combinatorica **1**(2) (June 1981) 169–197
14. Dantzig, G.B.: Linear Programming and Extensions. Princeton University Press (1963)