Restricted Strip Covering and the Sensor Cover Problem

Adam L. Buchsbaum* Alon Efrat † Shaili Jain ‡ Suresh Venkatasubramanian § Ke Yi ¶

Abstract

Suppose we are given a set of objects that cover a region and a duration associated with each object. Viewing the objects as jobs, can we schedule their beginning times to maximize the length of time that the original region remains covered? We call this problem the SENSOR COVER PROBLEM. It arises in the context of covering a region with sensors. For example, suppose you wish to monitor activity along a fence (interval) by sensors placed at various fixed locations. Each sensor has a range (also an interval) and limited battery life. The problem is then to schedule when to turn on the sensors so that the fence is fully monitored for as long as possible.

This one-dimensional problem involves intervals on the real line. Associating a duration to each yields a set of rectangles in space and time, each specified by a pair of fixed horizontal endpoints and a height. The objective is to assign a bottom position to each rectangle (by moving them up or down) so as to maximize the height at which the spanning interval is fully covered. We call this one-dimensional problem RESTRICTED STRIP COVERING. If we replace the covering constraint by a packing constraint (rectangles may not overlap, and the goal is to minimize the highest point covered), then the problem becomes identical to DYNAMIC STORAGE ALLOCATION, a well-studied scheduling problem, which is in turn a restricted case of the well known problem STRIP PACKING.

We present a collection of algorithms for RESTRICTED STRIP COVERING. We show that the problem is NP-hard and present an $O(\log\log\log n)$-approximation algorithm. We also present better approximation or exact algorithms for some special cases, including when all intervals have equal width. For the general SENSOR COVER PROBLEM, we distinguish between cases in which elements have uniform or variable durations. The results depend on the structure of the region to be covered: We give a polynomial-time, exact algorithm for the uniform-duration case of RESTRICTED STRIP COVERING but prove that the uniform-duration case for higher-dimensional regions is NP-hard. We give some more specific results for two-dimensional regions. Finally, we consider regions that are arbitrary sets, and we present an $O(\log n)$-approximation algorithm for the most general case.

1 Introduction

Sensors are small, low-cost devices that can be placed in a region to monitor local conditions. Distributed sensor networks have become increasingly more popular as advances in MEMS and fabrication allow for such systems that can perform sensing and communication. How sensors communicate is a well-studied problem. Our main interest is: Once a sensor network has been established, how can we maximize the lifetime of the network? It is clear that the limited battery capacities of sensors is a key constraint in maximizing the lifetime of a network. Additionally, research shows that partitioning the sensors into covers and iterating through them in a round-robin fashion increases the lifetime of the network [1, 4, 9, 10].

Definitions. Let $\mathcal{S}$ be a set of n sensors. Each sensor $s \in \mathcal{S}$ can be viewed as a point in some space with an associated region $R(s)$ of coverage. For every point $x \in R(s)$, s is *live at* x. Let U be the region to be covered by $\mathcal{S}$. U is *covered* by some $\mathcal{R} \subseteq \mathcal{S}$ if $U \subseteq \bigcup_{s\in\mathcal{R}} R(s)$. We call $\mathcal{R}$ a *feasible cover*. Every sensor $s \in \mathcal{S}$ can be *active* for a finite *duration* $d(s)$. Let $d_{\min} = \min_{s\in\mathcal{S}} d(s)$, and $d_{\max} = \max_{s\in\mathcal{S}} d(s)$.

PROBLEM 1.1. (SENSOR COVER) *Compute a* schedule S *of maximum* duration T*, in which each sensor* $s \in \mathcal{S}$ *is assigned a* start time $t(s) \geq 0$*, such that any* $x \in U$ *is covered by some active sensor at all times* $0 \leq t < T$*. That is, for all* $x \in U$ *and* $0 \leq t < T$*, there is some* $s \in \mathcal{S}$ *with* $x \in R(s)$ *and* $t(s) \leq t < t(s) + d(s)$.

A sensor is *redundant* in a schedule S if it can be removed without decreasing the duration of S. A schedule with no redundant sensors is *minimal*. It suffices to consider only minimal schedules, which may not utilize all sensors. As a convention, we set $t(s) = \infty$ if s is unused.

Prior work on the SENSOR COVER problem has focused solely on the case where the regions $R(\cdot)$ are arbitrary subsets of U and the durations are all identical. This assumption yields a *packing* constraint, and the problem reduces to partitioning the set of sensors into a maximum number of valid covers. This problem is known as SET COVER PACKING and is $\ln n$-hard to approximate, with a matching upper bound [6].

In practice, these assumptions appear overly constraining. Sensors will have arbitrary durations and typically define geometric regions of coverage: intervals, rectangles, disks, etc. In this paper, we consider such variations. In the RESTRICTED STRIP COVER problem, the $R(\cdot)$'s are one-dimensional intervals, and the problem is equivalent to sliding axis-parallel rectangles vertically to cover a rectangular region of maximum height. In the CUBE COVER problem, the $R(\cdot)$'s are axis-parallel rectangles, and the problem is akin to sliding cubes vertically in the z-dimension. We also

*AT&T Labs, alb@research.att.com.

†Comp. Sci. Dept., University of Arizona, alon@cs.arizona.edu.

‡Eng. and Appl. Sci. Div., Harvard University, shailij@eecs.harvard.edu. Part of the work was done while visiting AT&T Shannon Labs. Supported by the AT&T Labs Fellowship Program.

§AT&T Labs, suresh@research.att.com.

¶AT&T Labs, yike@research.att.com. Part of the work was done while at Duke University.

Table 1: Summary of results.

Shape of sensor	Uniform duration	Variable duration
Intervals	exact in P	NP-hard, $O(\log\log\log n)$-approx., $(2+\epsilon)$-approx. for equal-width
Convex with small cuttings	NP-hard, $O(\log(nd_{\max}/L))$-approx.; $O(\log(L_{\max}/L))$-approx. when congruent & fat	NP-hard, $O(\log n)$-approx.
Arbitrary sets	$\log n$-hard to approx., $O(\log n)$-approx. [6]	$\log n$-hard to approx., $O(\log n)$-approx.

consider SENSOR COVER, when the $R(\cdot)$'s are arbitrary subsets of a set U of size $O(n)$, with varying durations (in contrast to SET COVER PACKING).

In general, a schedule may activate and deactivate a sensor more than once. We call this a *preemptive schedule*. A *non-preemptive schedule* is a schedule in which each sensor is activated at most once. In this paper we only consider the non-preemptive problem. We have some preliminary results for the preemptive case, but more research is needed to gain a better understanding of the differences.

Our results. We show that most variants of SENSOR COVER are NP-hard, and we study approximation algorithms. For any point $x \in U$, let $L(x) = \sum_{s\in\mathcal{S}, s \text{ live at } x} d(s)$ be the *load at* x . Define the overall *load* $L = \min_x L(x)$; let $L_{\max} = \max_x L(x)$. We write L_X (rsp., $L_X(x)$) for the load of any subset X of sensors (rsp., at x). *OPT* denotes the duration of an optimal schedule; a trivial upper bound is $OPT \leq L$. All our approximation ratios are with respect to L. That $OPT \leq L$ allows the assumption that $d_{\max} \leq L$.

Table 1 summarizes our results. Notably, for RESTRICTED STRIP COVER, we give an $O(\log\log\log n)$-approximation algorithm for the general case and $O(1)$-approximations for some special cases, including a $(2+\epsilon)$-approximation when all intervals have equal width. For CUBE COVER, we give approximations that extend to any convex shape with small cuttings: rectangles, disks, ellipses, etc. We discuss RESTRICTED STRIP COVER in Section 2, CUBE COVER in Section 3, and the general SENSOR COVER problem in Section 4.

Related Work. SET COVER PACKING was studied by Feige et al. [6]. They considered the DOMATIC NUMBER problem, where the goal is to maximize the number of disjoint dominating sets on the set of vertices of a graph. A *dominating set* in a graph $G = (V, E)$ is a set $V' \subset V$ of vertices such that every $v \in V$ is either contained in V' or has a neighbor in V'. Feige et al. show that the DOMATIC NUMBER problem is hard to approximate within a factor of $(1-\varepsilon)\ln|V|$ for $\varepsilon > 0$, by first showing the hardness of approximation of the SET COVER PACKING problem. Note that the SET COVER PACKING problem is a combinatorial version of our problem, with each subset being a region of unit duration. Feige et al. also give a randomized $\ln n$-approximation algorithm, which they derandomize. Their work showed the first maximization problem proved to be approximable within polylogarithmic factors but no better.

The practical motivations for studying this problem have inspired the development of numerous heuristics. Slijepcevic and Potkonjak [10] introduce the SET K-COVER problem, where they are given a set of subsets of a base set and an integer k and ask if it is possible to construct at least k disjoint set covers. They prove that SET K-COVER is NP-complete, which is implied by Feige et al.'s result [6], and present a heuristic for constructing disjoint set covers.

Perillo and Heinzelman [9] study a variation of this problem, where they want to maximize the lifetime of a multi-mode sensor network. They compute all possible feasible covers and then translate their problem instance into a graph. Each sensor and feasible cover becomes a node. Sensors are connected to a feasible cover if they are contained in that feasible cover. They use linear programming to model additional energy constraints and solve a maximum flow problem on this graph. Their solution, while optimal, can be exponential in the size of the problem instance. Dasika et al. [4] also compute all possible feasible covers and develop heuristics for switching between these covers in order to maximize the lifetime of their sensor network.

Abrams, Goel, and Plotkin [1] study a variation of the problem where they are given a collection of subsets of a base set and a positive integer $k \geq 2$. Their goal is to partition the subsets into k covers, where the area of coverage, defined as the cardinality of a set, is maximized across all k covers. They give three approximation algorithms for this problem: a randomized algorithm, a distributed greedy algorithm, and a centralized greedy algorithm. Their randomized algorithm partitions sensors within $1 - \frac{1}{e}$ of the optimal solution. Their distributed greedy algorithm gives a $\frac{1}{2}$-approximation ratio. Their centralized greedy algorithm achieves an approximation factor of $1 - \frac{1}{e}$. They also prove a $\frac{15}{16}$-hardness result for their problem.

We are unaware of previous work on the RESTRICTED STRIP COVER problem. Some of the closely related problems are well studied, however. If we replace the covering constraint by a packing constraint (rectangles may not overlap, and the goal is to minimize the height of the highest point covered), then the problem becomes DYNAMIC STORAGE ALLOCATION [7, Problem SR2], for which there is a

$(2+\epsilon)$-approximation [3]. If we further allow rectangles to move both vertically and horizontally, then the problem becomes STRIP PACKING, which has a $(1+\epsilon)$-approximation up to an additive term [8].

2 Restricted Strip Cover

Consider an instance $\mathcal{S}$ of RESTRICTED STRIP COVER (RSC). For ease of presentation, we define $R(s)$ as a semi-closed interval $[\ell(s), r(s))$ for each $s \in \mathcal{S}$; the *width* of s is $r(s) - \ell(s)$. We assume w.l.o.g. that all interval coordinates are integers in $[0, 2n-1]$ and that $U = [0, 2n-1)$, because there are at most $2n$ distinct interval endpoints. It is convenient to view scheduled sensors as semi-closed rectangles in the plane, with intervals along the x-axis and durations along the y-axis. Thus a valid schedule S of duration T is one in which any point (x, y) in the subplane $U \times [0, T)$ is covered by some sensor s; i.e., $\ell(s) \leq x < r(s)$ and $t(s) \leq y < t(s) + d(s)$. The problem is equivalent to sliding axis-parallel rectangles vertically to cover a rectangular region of maximum height. Therefore, in this section we use the terms "sensor" and "rectangle" interchangeably. We say two or more rectangles *overlap* if they cover some common point. When discussing multiple schedules, we write $t_S(s)$ to denote the start time of s in some schedule S.

We assume all durations are positive integers. Let S be some schedule of $\mathcal{S}$. Define *level* j of S to be the horizontal slice of sensors that cover points in the y-range $[j-1, j)$. A *gap* is a point p such that no sensor covers p. For $i \in U$, define $M(S, i)$ to be the greatest y-coordinate j such that no gap exists below j at i; i.e., $M(S, i) = \max\{j : \forall j' < j, \exists s \in S,\ s \text{ covers } (i, j')\}$. Then the *duration* of S is $M(S) = \min_i M(S, i)$.

Our main results are an $O(\log\log\log n)$-approximation for arbitrary intervals and a $(2+\epsilon)$-approximation for intervals whose x-projections are non-nested, which includes the case of uniform width. We use three main components:

1. a simple, exact algorithm if all sensors have the same duration (Section 2.1);
2. an exact, dynamic programming algorithm, which runs in $\text{poly}(n)$ time if $L = O(\log n / \log\log n)$ and yields a PTAS when $L = O(d_{\min} \log n / \log\log n)$ (Section 2.2);
3. $(1+\epsilon)$-approximations when $L = \Omega(d_{\max} \log n \cdot \min\{1/\epsilon, \log(d_{\max}/d_{\min})\}/\epsilon^4)$ (Section 2.3).

2.1 Uniform-Duration Sensors If all sensors have the same duration, a simple greedy algorithm gives an exact solution of duration L. Define $\mathcal{S}_i = \{s \in \mathcal{S} : s \text{ is live at } i\}$. Assume by scaling that all sensors have unit duration. We proceed left-to-right, starting at $i = 0$ and constructing a schedule S while maintaining the following invariants after scheduling sensors in $\mathcal{S}_i$: (i) no sensors overlap at any x-coordinate $\geq i$, and (ii) $M(S, i) = L$.

When $i = 0$, select any L sensors that are live at 0, and schedule them without overlap, establishing the initial invariants. Assuming the invariants are true at i, schedule $\mathcal{S}_{i+1}$ as follows. If there are no gaps at $i+1$, we are done, as the invariants extend to $i+1$. Otherwise, assume there are $k > 0$ unit-duration gaps at $i+1$. At least k sensors in $\mathcal{S}_{i+1}$ must be unscheduled, which can be used to fill the gaps.

2.2 A Dynamic Programming Solution for Small L We give a dynamic program to determine if there is a schedule S such that $M(S) = T$ for a fixed T. The dynamic program is similar to that of Buchsbaum et al. [3], but we need a new analysis, as their analysis would yield an $n^{O(n)}$ time bound here. Below we ignore portions of sensors that extend above level T in any schedule.

Define $\mathcal{S}_{\leq i} = \bigcup_{0 \leq k \leq i} \mathcal{S}_k$. Consider some schedules S_{i-1} of $\mathcal{S}_{i-1}$ and S_i of $\mathcal{S}_i$ such that $M(S_{i-1}, i-1) = M(S_i, i) = T$. We say that S_{i-1} and S_i are *compatible* if (i) for all $s \in S_{i-1} \cap S_i$, $t_{S_{i-1}}(s) = t_{S_i}(s)$; and (ii) for all $j \in [0, T)$, (i, j) is covered by S_{i-1} or S_i. The first condition stipulates that any sensor in both schedules must have the same start time in each; the second requires a sensor in S_i to be scheduled to cover each level at which coverage stops at $i-1$ in S_{i-1}. For each i, we populate an array C_i indexed by possible schedules of $\mathcal{S}_i$. For any S_i, define $C_i[S_i] = 1$ if there is a schedule S of $\mathcal{S}_{\leq i}$ that respects S_i and has $M(S, x) = T$ for $0 \leq x \leq i$; and $C_i[S_i] = 0$ otherwise. Then $C_i[S_i] = 1$ if and only if $M(S_i, i) = T$ and there exists some schedule S_{i-1} of $\mathcal{S}_{i-1}$ such that $C_{i-1}[S_{i-1}] = 1$ and S_{i-1} is compatible with S_i. For $i = 0$, $C_0[S_0] = 1$ for precisely those schedules S_0 of $\mathcal{S}_0$ that have $M(S_0, 0) = T$. The dynamic program then populates the arrays C_i in increasing order of i, by checking all schedules of $\mathcal{S}_i$ for each i. Ultimately we check if there is some schedule S_{2n-1} of $\mathcal{S}_{2n-1}$ such that $C_{2n-1}[S_{2n-1}] = 1$.

First Analysis. For a schedule S_i of $\mathcal{S}_i$, denote by $\partial(S_i)$ the vertical boundaries of the union of the rectangles of S_i. If S_i is part of a minimal schedule S of duration T, then any rectangle of S_i must cover some point on $\partial(S_i)$ that is covered by no other rectangles in S_i. Thus $|S_i| \leq 2T$, because $\partial(S_i)$ has total length $2T$.

Because there are at most $2T$ sensors per schedule, there are at most $\binom{n}{2T} T^{2T}$ possible schedules of $\mathcal{S}_i$. Each schedule of $\mathcal{S}_i$ must be checked for compatibility against each schedule of $\mathcal{S}_{i-1}$, and checking compatibility of a pair of schedules takes $O(T)$ time. Hence the time to run the whole dynamic program is $2n\left(\binom{n}{2T} T^{2T}\right)^2 O(T) = (nT)^{O(T)} = (nL)^{O(L)}$. To determine *OPT*, we run the dynamic program for each of the L possible values of T, which does not affect the overall asymptotics.

Partitioning the Dynamic Program. It suffices to run the dynamic program only on x-coordinates with relatively few live sensors. Let $X = \{i : |\mathcal{S}_i| < 5T\}$. We claim that $\mathcal{S}$ has a schedule of duration T iff $\mathcal{S}$ has a schedule S such that $M(S, i) \geq T$ for any $i \in X$. We prove the "if" part; the "only if" part is clear.

Assume that there is a minimal schedule S of duration T that only covers X. We show how to schedule the sensors not used in S to cover all x-coordinates. Consider any maximal interval $\bar{X}$ of x-coordinates not in X. At most $4T$ sensors from S are live at any $i \in \bar{X}$, because any such sensor is also live at either $\min(\bar{X}) - 1$ or $\max(\bar{X}) + 1$, and at most $2T$ are live at either one. By construction, there are at least $5T$ sensors live at any $i \in \bar{X}$, so there are at least $5T - 4T = T$ sensors live at i that are not used by S and hence are available, which suffice to cover all the levels at i. If such a sensor s should also be live at another $i' \in \bar{X}$ (or another i' in another $\bar{X}'$), it reduces by one both the number of potential uncovered levels and the number of available sensors live at i', so enough sensors will remain at i'.

Therefore we need only run the dynamic program on the x-coordinates in X. This takes only $2n \cdot T^{O(T)}$ time, because there are fewer than $5T$ sensors live at any $i \in X$.

THEOREM 2.1. RSC *can be solved in time* $2n \cdot L^{O(L)}$.

COROLLARY 2.1. RSC *can be solved in* $\mathrm{poly}(n)$ *time if* $L < c \cdot \log n / \log\log n$ *for any constant* c.

Using a standard trick, a PTAS follows directly by appropriately truncating durations.

COROLLARY 2.2. *There is a PTAS for* RSC *if* $L < c \cdot d_{\min} \log n / \log\log n$ *for any constant* c.

2.3 Algorithms for Small Durations We now have algorithms for the cases of uniform duration and large durations (relative to load). Here we consider the case when all durations are small relative to load. To do so, we develop a *grouping* technique, which builds on the *boxing* technique of Buchsbaum et al. [3]. Although we follow the rough outline of their technique, the covering (as opposed to packing) nature of our problem necessitates new ideas.

The basic idea of grouping is to group shorter sensors into longer, virtual sensors until all the sensors have equal duration, at which point the greedy algorithm is invoked. Ensuring that the load does not decrease too much during the process is the key to our algorithms.

Grouping Sensors. A *grouping* is a partition of a set Y of sensors into a set G of groups, each of which is then replaced by a rectangle that can be covered by the sensors in the group. The *duration* of a group is that of the rectangle that replaces it. These rectangles form a modified instance. L_G (rsp., $L_G(i)$) is defined to be the *load of the groups* (rsp., *at* i). Note that $L_G(i) \leq L_Y(i)$, since portions of the sensors in a group that are overlapped or outside the rectangle are not counted in $L_G(i)$. We give polynomial-time procedures to group a set Y of sensors of unit duration into G such that $L_G(i)$ is not much smaller than $L_Y(i)$ for any i.

First, we give a grouping of a set of sensors that are all live at a fixed x-coordinate. The following adapts Lemma 2.1 of Buchsbaum et al. [3], and the proof is similar, although simpler; details are in Appendix A.1.

LEMMA 2.1. *Given a set* Y *of unit-duration sensors, all live at some fixed* x*-coordinate* x_0*, an integer* group-duration parameter D*, and a sufficiently small positive* ϵ*, there is a set* G *of groups, each of duration* D*, such that for any* i*,* $L_G(i) > L_Y(i)/(1+\epsilon) - 4D\lceil 1/\epsilon \rceil$.

We now partition the input so that we can apply Lemma 2.1 individually to the parts.

DEFINITION 2.1. *A* γ-grouping *is a partition of sensors into a set of groups such that: (1) In each group, there is an* anchor *(*x*-coordinate)* i *at which all sensors in the group are live; and (2) for any* x*-coordinate* i*, the set of sensors live at* i *are drawn from no more than* γ *groups.*

Notes. Sensors in a group may share many anchors in common; the anchor of the group is one distinguished from this set. Not all sensors live at an anchor will be in its group. Also, the existence of a γ-grouping is a purely combinatorial property of a family of ranges, like the canonical subsets used in range searching [2].

LEMMA 2.2. *Given a set* Z *of unit-duration sensors that admits a* γ*-grouping, an integer* group-duration parameter D*, and a sufficiently small positive* ϵ*, there is a set* G *of groups, each of duration* D*, such that at any* x*-coordinate* i*,* $L_G(i) > L_Z(i)/(1+\epsilon) - O(\gamma D/\epsilon)$.

Proof: Let G be a γ-grouping of Z. Each group in G possesses an anchor and thus satisfies the premises of Lemma 2.1. Apply Lemma 2.1 to each group of G. Let V be the set of anchors, and let Z_v denote the set of all the sensors in the group that has v as an anchor.

Consider any x-coordinate i. By Lemma 2.1 and the fact that the Z_v form a partition, $L_G(i) > \sum_{v \in V}(L_{Z_v}(i)/(1+\epsilon) - 4D\lceil 1/\epsilon \rceil)$. By the γ-grouping property, there are only γ relevant terms in the summation, so $L_G(i) > L_Z(i)/(1+\epsilon) - 4\gamma D\lceil 1/\epsilon \rceil$. □

LEMMA 2.3. *Any set of intervals has an* $O(\log n)$*-grouping.*

Proof: Build an interval tree $\mathcal{T}$ on the intervals. For each node v of $\mathcal{T}$, form group Z_v containing the intervals associated with v. Clearly, the x-coordinate of the dividing line corresponding to v is a valid anchor of the group Z_v.

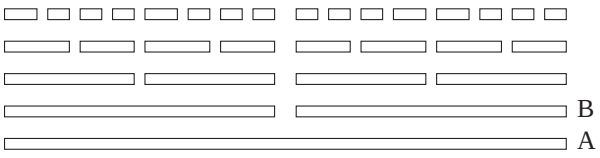

Figure 1: A bad example for grouping.

$\mathcal{T}$ has depth $O(\log n)$. For nodes u, w at the same level of $\mathcal{T}$, sets Z_u and Z_v are disjoint. Thus, the intervals live at any x-coordinate are distributed among $O(\log n)$ groups. □

Remark. This bound is tight in general. Consider any grouping of the example in Figure 1. Let U be $[0, 1]$. We show there must be an x-coordinate i such that the sensors live at i belong to $\Omega(\log n)$ groups. Initially, i might lie anywhere in U. Sensor A is assigned to some group; assume w.l.o.g. that the anchor of this group lies in the left half of A, i.e., $[0, 0.5)$. Restrict the range of candidate x-coordinates for i to $[0.5, 1]$. Note that all groups involving intervals in this range (like B) must be different from A's group. Repeat this process with B. As the range of candidate x-coordinates for i thereby decreases, we maintain an increasing set of intervals that all must belong to different groups. The process terminates after $\Omega(\log n)$ steps.

More structure on the intervals allows for better groupings. Consider families of *non-nested* intervals, in which no interval properly contains another. Uniform-width intervals are a special case.

LEMMA 2.4. *Collections of non-nested intervals admit* 2*-groupings.*

Proof: In non-decreasing order by left endpoint, greedily add intervals into the first group as long as they all share some x-coordinate. When no further progress can be made, create a new group and continue.

Consider three groups A, B, C created in consecutive order. Let s_A, s_B, and s_C be the first intervals picked in each group. By assumption, all intervals in A are live at the right endpoint of s_A. If s_A and s_B are live at some common x-coordinate, then s_B is also live at the right endpoint of s_A, which is not possible, since s_B started a new group.

Thus, s_A, s_B and s_C are mutually disjoint. Since no interval of C has its left endpoint to the left of that of s_C, no interval of A can be live at an x-coordinate of an interval of C, or s_B would be nested. Hence, all intervals active at any x-coordinate between the left endpoints of s_B and s_C must be from A or B. □

The Algorithm. Henceforth, we assume that the input admits a γ-grouping. By Lemma 2.3, $\gamma = O(\log n)$. Let ϵ be a sufficiently small error parameter, and let $D = d_{\max}\lceil 1/\epsilon \rceil$.

THEOREM 2.2. *For any sufficiently small positive ϵ, Algorithm 1 runs in $\mathrm{poly}(n, 1/\epsilon)$ time and gives a schedule of the* RSC *problem with duration at least $L/(1+\epsilon) - O\left(\gamma d_{\max} \log(d_{\max}/d_{\min})/\epsilon^3\right)$.*

Algorithm 1 Approximation algorithm via grouping

(1) Truncate each sensor of duration d to $\lceil (1+\epsilon)^k \rceil$, where $(1+\epsilon)^k \le d < (1+\epsilon)^{k+1}$ for some integer k. Let X be the set of truncated sensors.
(2) For each $d = \lceil (1+\epsilon)^k \rceil$, $k = \lfloor \log_{1+\epsilon} d_{\min} \rfloor, \ldots, \lceil \log_{1+\epsilon} d_{\max} - 1 \rceil$, do the following. Let X_d denote the set of truncated sensors of duration d. Scale each sensor in X_d down by a factor of d, apply Lemma 2.2 with group-duration parameter $\lceil D/d \rceil$ and the given ϵ, and then scale the obtained groups back up by d.
(3) Let G be the set of rectangles obtained from Step (2). Truncate them so that they all have duration exactly D. Call the resulting set of rectangles G'.
(4) Apply the greedy algorithm to G'.

Proof: We will show that truncating and grouping do not decrease the load at any i excessively.

By Lemma 2.2, Step 2 produces a grouping G_d of X_d of duration $\lceil D/d \rceil d$ such that at any i, $L_{G_d}(i) > L_{X_d}(i)/(1+\epsilon) - O(\gamma D/\epsilon)$. Summing over all d, we have

$$\begin{aligned} L_G(i) &> \frac{1}{1+\epsilon} L_X(i) - O\left(\frac{\gamma D \log(d_{\max}/d_{\min})}{\epsilon \log(1+\epsilon)}\right) \\ &= \frac{1}{1+\epsilon} L_X(i) - O\left(\frac{\gamma d_{\max} \log(d_{\max}/d_{\min})}{\epsilon^3}\right). \end{aligned}$$

Truncating the sensors in Step (1) decreases their durations by a factor of at most $1+\epsilon$, so $L_X(i) \ge \frac{1}{1+\epsilon} L(i)$. Truncating the groups in Step (3) also decreases their durations by a factor of at most $\frac{\lceil D/d \rceil d}{D} \le \frac{D+d}{D} \le 1+\epsilon$. Since $\frac{1}{(1+\epsilon)^3} \ge \frac{1}{1+7\epsilon}$, we have $L_{G'}(i) > L(i)/(1+7\epsilon) - O\left(\gamma d_{\max} \log(d_{\max}/d_{\min})/\epsilon^3\right)$. Finally, applying the greedy algorithm in Step (4) yields a schedule of duration $\min_i L_{G'}(i) > L/(1+7\epsilon) - O(\gamma d_{\max} \log(d_{\max}/d_{\min})/\epsilon^3)$. Replacing ϵ with $\epsilon/7$ gives the desired result. □

By bootstrapping Steps (1)–(3) of Algorithm 1, we can replace the $O(\log(d_{\max}/d_{\min}))$ factor with $O(1/\epsilon)$, yielding the following result, the proof of which is in Appendix A.2.

THEOREM 2.3. *For any sufficiently small positive ϵ, there is an algorithm that runs in $\mathrm{poly}(n, 1/\epsilon)$ time and gives a schedule to the* RSC *problem with duration at least $L/(1+\epsilon) - O\left(\gamma d_{\max}/\epsilon^4\right)$.*

COROLLARY 2.3. *There is a constant c, such that for any small enough positive real ϵ, the algorithm gives a schedule of duration at least $L/(1+\epsilon)$ for any $L \ge \gamma c d_{\max}/\epsilon^5$.*

2.4 Putting the Pieces Together Theorem 2.3 yields a good approximation only when $d_{\max}$ is small. On the other

hand, Corollary 2.2 yields a good approximation when $d_{\min}$ is large. We need the following technical lemma.

LEMMA 2.5. *For any partition $\{\mathcal{R}_1, \ldots, \mathcal{R}_k\}$ of $\mathcal{S}$ and any x-coordinate i, some $\mathcal{R}_j$ has load at least L/k at i; define $m(i)$ to be any such j.*

Proof: By contradiction, if there were some i such that $L_{\mathcal{R}_j}(i) < L/k$ for $1 \le j \le k$, then $L_{\mathcal{S}}(i) < L$. Set $m(i) = \text{argmax}_j\{L_{\mathcal{R}_j}(i)\}$. □

Consider the case when $\gamma = O(1)$. Fix a parameter β, and partition $\mathcal{S}$ into two subsets: $\mathcal{R}_0$ consisting of all sensors with duration at least βL (the large sensors), and $\mathcal{R}_1$ containing the remaining (small) sensors. Invoking Lemma 2.5, for each j, we use $\mathcal{R}_j$ to cover all the x-coordinates i where $m(i) = j$. Then by setting $\beta = \epsilon'^5/(\gamma c)$: for $\mathcal{R}_0$, Corollary 2.2 yields a solution of duration at least $L/(2(1+\epsilon'))$; and for $\mathcal{R}_1$, Corollary 2.3 yields a solution of duration at least $L/(2(1+\epsilon'))$. Combining the two solutions, we obtain a solution of duration $L/(2+2\epsilon')$. Setting $\epsilon = \epsilon'/2$ gives us our first result:

THEOREM 2.4. *There exists a $(2+\epsilon)$-approximation for* RSC *for inputs that admit $O(1)$-groupings.*

COROLLARY 2.4. *If all sensors have equal width, we can obtain a $(2+\epsilon)$-approximation for* RSC.

Proof: This follows from the above and Lemma 2.4. □

If γ is an increasing function of n, we must introduce a middle layer in the decomposition of sensors. Place all sensors of duration at least $h = L \log\log n / \log n$ into set $\mathcal{R}_0$. Fix a parameter ℓ, and for $1 \le i \le \ell$, place all sensors of duration between $h/2^i$ and $h/2^{i-1}$ into set $\mathcal{R}_i$, truncating all their durations to $h/2^i$; this at worst doubles the overall approximation ratio. Place all sensors of duration at most $h/2^\ell$ in $\mathcal{R}_{\ell+1}$.

We call $\mathcal{R}_0$ the *large subset*; $\mathcal{R}_i, 1 \le i \le \ell$, the *middle subsets*; and $\mathcal{R}_{\ell+1}$ the *small subset*. Invoking Lemma 2.5, for each j, we use $\mathcal{R}_j$ to cover all the x-coordinates i where $m(i) = j$. For the large subset, we use Corollary 2.2 to find a schedule of duration at least $L/((1+\epsilon)(\ell+2))$. For a middle subset $\mathcal{R}_j, j \ge 1$, because all its sensors have the same duration, we can use the greedy algorithm to find a schedule of duration at least $L/(\ell+2)$. For the small subset $\mathcal{R}_{\ell+1}$, we use Theorem 2.3 with $\epsilon = 1$. Since the sensors in $\mathcal{R}_{\ell+1}$ have maximum duration $h/2^\ell = L \log\log n/(2^\ell \log n)$, Theorem 2.3 yields a schedule of duration at least

$$\frac{L}{2(\ell+2)} - O\left(\gamma \frac{L \log\log n}{2^\ell \log n}\right).$$

If $\gamma = O(\log n/\log\log n)$, then setting $\ell = O(1)$ suffices to make the term $\frac{L}{2(\ell+2)}$ dominate, yielding a constant-factor approximation. For arbitrary intervals, we know from Lemma 2.3 that $\gamma = O(\log n)$, requiring $\ell \ge 3 \log\log\log n$, thereby yielding an $O(\log\log\log n)$-approximation.

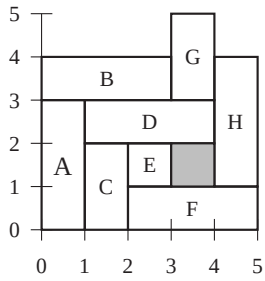

Figure 2: [3] A set of sensors (in the form $(\ell(\cdot), r(\cdot), d(\cdot))$) $A = (0,1,3)$, $B = (0,3,1)$, $C = (1,2,2)$, $D = (1,4,1)$, $E = (2,3,1)$, $F = (2,5,1)$, $G = (3,4,2)$, and $H = (4,5,3)$. The shaded region is a gap. In this example, $L = 4$ but *OPT* $= 3$, which can be realized by sliding G down so that $t(G) = 1$.

THEOREM 2.5. *There exists a polynomial-time constant-factor approximation for the* RSC*-problem on collections of intervals that admit $O(\log n/\log\log n)$-groupings. For general collections of intervals, there exists a polynomial-time $O(\log\log\log n)$-approximation algorithm.*

2.5 Hardness To prove that RSC with variable durations is NP-hard, we exploit an identity to DYNAMIC STORAGE ALLOCATION in a special case. An instance of DSA is like one of RSC, except that the instance load is defined as the maximum load at any x-coordinate, and the goal is to schedule the sensors (*jobs*, in DSA parlance) without overlap to minimize the makespan. If the load is equal for all x-coordinates, then *OPT* $= L$ for either problem implies a schedule that is a solid rectangle of height *OPT* $= L$.

Stockmeyer proves that determining if there exists a solution to DSA of a given makespan is NP-complete [7, Problem SR2]. In fact, he proves that given a DSA instance with uniform load, determining if *OPT* $= L$ is NP-complete, even if durations are restricted to the set $\{1,2\}$, and by the above identity, the same is true for RSC.

To establish a gap between *OPT* and L, consider the example in Figure 2, in which $L = 4$ but *OPT* $= 3$. Scaling shows that no approximation algorithm can guarantee a ratio of better than $4/3$ with respect to L.

3 Cube Cover

3.1 Hardness Results When the $R(\cdot)$'s are axis-aligned rectangles and U is a two-dimensional region, the problem is NP-hard even when the sensors have uniform duration, in contrast to the uniform-duration case for RSC. We use a reduction from an instance of NAE-3SAT with n variables and m clauses to an instance of CUBE COVER.

An instance I of NAE-3SAT is a set U of variables and a collection $C = \{C_1, C_2, ..., C_m\}$ of clauses over U, such that $|C_i| = 3$ for each i. The problem is to determine if there a truth assignment for U such that each clause in C has at least one true literal and at least one false literal [7]. A key property is that if X is a satisfying assignment for an instance I of NAE-3SAT, $\bar{X}$ is also a satisfying assignment of I.

Given I, we construct an associated graph $G(I)$, with vertices for each variable and each clause. We draw an edge between a clause vertex and a variable vertex if the variable appears in the clause. The graph is drawn on a planar grid within a bounding box U. From $G(I)$, we construct an instance $\mathcal{S}(I)$ of CUBE COVER that has a schedule of duration 2 if I is satisfiable but only 1 if I is unsatisfiable. Details will appear in the full paper.

The construction eliminates the possibility of a PTAS as well. Assume we have a PTAS $\mathcal{P}$ for CUBE COVER. On input $(\mathcal{S}(I), \epsilon)$, where $\epsilon > 0$ and $\mathcal{S}(I)$ is an instance of CUBE COVER induced by the above construction, $\mathcal{P}$ would output a solution with duration T, where $T \geq (1-\epsilon) \cdot OPT$. Setting $\epsilon = 0.25$, $T \geq 1.5$ when $OPT = 2$, and $0.75 \leq T \leq 1$ when $OPT = 1$. Thus we can use $\mathcal{P}$ to solve NAE-3SAT.

THEOREM 3.1. *CUBE COVER is NP-hard and does not admit a PTAS, even with uniform duration.*

3.2 Rectangles With Uniform Duration We consider approximation algorithms if all sensors have unit duration and scale to get the uniform-duration results in Table 1. First we prove a technical lemma, which actually holds for arbitrary sets.

LEMMA 3.1. *Let U be an m-element set. For $s \in \mathcal{S}$, $R(s)$ is an arbitrary subset of U with unit duration. There exists some constant c large enough, such that if $L > c \ln m$, then in polynomial time we can find a subset $\mathcal{R} \subseteq \mathcal{S}$ and a schedule of $\mathcal{R}$ with duration at least $L/\ln m$ so that the remaining load $L_{\mathcal{S}\setminus\mathcal{R}} \geq L/2$.*

Proof: We take covers from $\mathcal{S}$ one by one. Let L_i be the load of the remaining sensors after taking the i^{th} cover. For the $(i+1)^{\text{th}}$ cover $\mathcal{X}$, we take each remaining sensor into $\mathcal{X}$ with probability $p = c \ln m / L_i$. Then we check if (1) $\mathcal{X}$ is a valid cover, and (2) $L_{i+1} > L_i - \frac{1}{2} \ln m$. For any $x \in U$, the probability that x is not covered is at most $(1-p)^{L_i} < m^{-c}$, so (1) occurs with probability at least $1 - m^{1-c}$ (probability of union of events). For any $x \in U$, the probability that $L_i(x) \leq L_i - \frac{1}{2} \ln m$ is at most $m^{-(2c-1)^2/8c}$ (Chernoff bound), so (2) occurs with probability at least $1 - m^{1-(2c-1)^2/8c}$. Thus we can choose c large enough that both (1) and (2) occur with high probability (e.g., $> 1/2$). We repeatedly take $\mathcal{X}$ until this happens and then proceed to the next cover. We repeat this procedure until L_{i+1} drops below $L/2$, and the lemma follows. □

The basic idea of our algorithms is the following. Take a partition of U with a small number of cells, and then crop $R(\cdot)$ so that each sensor fully covers a number of cells but is completely disjoint from the rest. We ensure that the load does not decrease by more than a constant factor and then apply Lemma 3.1.

THEOREM 3.2. *If each sensor $s \in \mathcal{S}$ has unit duration, then there is a polynomial-time $O(\log(n/L))$-approximation algorithm for* CUBE COVER.

Proof: We assume $L > c \ln n$ for some large constant c; otherwise we just take one cover, and the theorem follows. It is well known that sets of rectangles in the plane admit $(1/r)$-cuttings: there exists a subset $\mathcal{R} \subset \mathcal{S}$ of $r \log r$ rectangles such that in the partition $\mathcal{A}_{\mathcal{R}}$ determined by the rectangles of $\mathcal{R}$, each face is intersected by the boundaries of at most cn/r rectangles of $\mathcal{S}$ [5]. We choose $r = \lceil 2cn/L \rceil$, so $cn/r \leq L/2$.

Let f be a face of $\mathcal{A}_{\mathcal{R}}$, and let $\mathcal{S}_f \subseteq \mathcal{S}$ denote the subset of rectangles that fully contain f. Since the load at every point in f is at least L and only $L/2$ rectangles partially cover f, we derive $|\mathcal{S}_f| \geq L/2$. Now replace each rectangle $R(s)$ by a cropped region that consists of all faces of $\mathcal{A}_{\mathcal{R}}$ that s fully covers. This yields an instance of SENSOR COVER, with a universe of size $r^2 \log^2 r$ and load $L' \geq L/2$. Applying Lemma 3.1 yields the desired result. □

When all the $R(\cdot)$'s have the same size, a more careful cropping scheme yields an improved bound.

THEOREM 3.3. *If each sensor $s \in \mathcal{S}$ has unit duration and each $R(s)$ is a unit square, then there is a polynomial-time $O(\log(L_{\max}/L))$-approximation algorithm for* CUBE COVER.

Proof: We assume $L > c \ln L_{\max}$ for some large constant c; otherwise we just take one cover, and the theorem follows. We draw a unit-coordinate grid Γ inside U. There are only $O(n/L)$ cells in Γ. For a cell $\gamma \in \Gamma$, let $\mathcal{S}(\gamma) \subseteq \mathcal{S}$ denote the set of squares of $\mathcal{S}$ that intersect γ. Let $n_{max} = \max_\gamma |\mathcal{S}(\gamma)|$. Packing arguments imply $n_{max} \leq 4L_{\max}$.

Two cells in Γ are *independent* if they are at least two grid cells apart from each other in both dimensions. It is easy to see that we can partition Γ into 9 *independent sets* $\Gamma_1, \ldots, \Gamma_9$, where all cells in any one set are mutually independent. In the following, we will show how to make $\Omega(L/\ln(n_{max}/L))$ covers for Γ_1, such that the remaining load is at least $L/8$. Then we repeat the process for $\Gamma_2, \ldots, \Gamma_9$, and ultimately we derive a schedule that covers all cells with duration $\Omega(L/\ln(n_{max}/L)) = \Omega(L/\ln(L_{\max}/L))$.

By the definition of independence, we can isolate the cells in Γ_1 and need only show that for any $\gamma \in \Gamma_1$, we can make $\Omega(L/\ln(n_{max}/L))$ covers from $\mathcal{S}(\gamma)$ without

decreasing the load of any of its neighboring 8 cells by more than a factor of 8. The load of $\mathcal{S}(\gamma)$ inside γ is at least L, so following the approach used in the proof of Theorem 3.2, we build a partition $\mathcal{A}$ in γ and its neighboring cells such that each face of $\mathcal{A}$ intersects the boundaries of at most $L/2$ squares from $\mathcal{S}(\gamma)$. $\mathcal{A}$ has size $r^2 \log^2 r$, where $r = \lceil 2cn_{max}/L \rceil$. We further partition the faces of $\mathcal{A}$ that intersect the boundary of γ, such that each face of $\mathcal{A}$ is either inside γ or outside. This at most doubles the size of $\mathcal{A}$. Let $\mathcal{F}$ be the set of faces of $\mathcal{A}$ that are fully covered by at least $L/4$ sensors from $\mathcal{S}(\gamma)$. $\mathcal{F}$ includes all faces inside γ and some faces outside. For any face not in $\mathcal{F}$, the load of $\mathcal{S} \setminus \mathcal{S}(\gamma)$ must be at least $L - L/2 - L/4 = L/4$, so we can ignore it. Consider the faces in $\mathcal{F}$. Crop the squares of $\mathcal{S}(\gamma)$ according to $\mathcal{F}$ as in the proof of Theorem 3.2. After cropping, by construction the load at each face of $\mathcal{F}$ remains at least $L/4$. Apply Lemma 3.1 with $U = \mathcal{F}$ and $\mathcal{S}(\gamma)$, which yields $\Omega(L/\ln(n_{max}/L))$ covers while the remaining sensors have load at least $L/8$ for any face of $\mathcal{A}$. □

Remark. These results extend to convex shapes that admit small cuttings: disks, ellipses, etc.

4 Sensor Cover

Now consider the general SENSOR COVER problem, in which each $R(\cdot)$ is an arbitrary subset of a finite set U of size $|U| = O(n)$. We show that a random schedule of the sensors yields an $O(\log n)$-approximation with high probability. This result extends that of Feige et al. [6], which deals with the unit-duration case.

Let $T = cL/\ln n$, where c is some constant to be determined later. We show that if we choose the start time of each sensor randomly between 0 and T, then we will have a valid schedule with high probability. In order to avoid fringe effects, we must choose positions near 0 or T judiciously. More precisely, for a sensor s of duration $d(s) < T$, we choose its start time $t(s)$ uniformly at random between $-d(s)$ and T; if $t(s) < 0$, we reset it to 0. If $d(s) \geq T$, we simply set $t(s) = 0$. Divide T evenly into $2n$ time intervals $[t_0 = 0, t_1], [t_1, t_2], \ldots, [t_{2n-1}, t_{2n} = T]$, each of length $T/2n$. If $d(s) \geq T/n$, then for any $x \in R(s)$ and in any time interval, x is covered by s with probability at least $(d(s) - T/2n)/(T + d(s)) \geq \frac{1}{4} \cdot d(s)/T$.

Consider any $x \in U$, and let $\{s_1, \ldots, s_k\}$ be the set of sensors live at x with durations at least T/n. We know that $\sum_{i=1}^{k} d(s_i) \geq L - T/n \cdot n \geq L/2$. In any time interval $[t_i, t_{i+1}]$, the probability that x is not covered is at most

$$\prod_{i=1}^{k} \left(1 - \frac{d(s_i)}{4T}\right) \leq \prod_{i=1}^{k} \exp\left(-\frac{d(s_i)}{4T}\right)$$
$$\leq \exp\left(-\frac{L}{8T}\right) = \exp\left(-\frac{\ln n}{8c}\right) = n^{-\frac{1}{8c}}.$$

There are $O(n^2)$ different $(x, [t_i, t_{i+1}])$ pairs, so the probability that some $x \in U$ is not covered at some time is at most $O(n^2) \cdot n^{-\frac{1}{8c}} = O\left(n^{2-\frac{1}{8c}}\right)$. With $c < 1/16$, we obtain a valid schedule with high probability.

The algorithm can be de-randomized using the method of conditional probability. We omit the details.

SET COVER PACKING reduces to SENSOR COVER. Given an optimal schedule for an instance of SENSOR COVER, we can "snap" each starting time $t(s)$ to the integer $\lceil t(s) \rceil$ without introducing gaps or decreasing the total duration. Hence, the lower bound of Feige et al. [6] applies.

THEOREM 4.1. *There exists a polynomial-time $O(\log n)$-approximation algorithm for the* SENSOR COVER *problem. This bound is tight up to constant factors.*

5 Open Problems

Ideally we would like to prove stronger hardness results or find better approximation algorithms in order to narrow the gap between our lower and upper bounds. In fact, we have not ruled out the possibility of a PTAS for the RESTRICTED STRIP COVER problem, although it cannot be in terms of L. It would be interesting to see if other techniques for geometric optimization problems could be applied to our problem as well.

We are also interested in understanding preemptive schedules better. For RESTRICTED STRIP COVER, a simple algorithm based on maximum flow yields an optimal preemptive schedule in polynomial time. In higher dimensions, however, it is not fully understood in which situations non-preemptive schedules are sub-optimal when compared with the best preemptive schedules. For example, using essentially the same argument as in the NP-hardness proof for CUBE COVER, we can show that its preemptive variant remains NP-hard. In general, we would like to uncover the relationship between the load of the problem instance, the duration of the optimal preemptive schedule, and the duration of the optimal non-preemptive schedule.

Acknowledgements. We thank Nikhil Bansal for pointing us to the paper by Kenyon and Remila [8] and Piotr Indyk for fruitful discussions.

References

[1] Z. Abrams, A. Goel, and S. Plotkin. Set K-cover algorithms for energy efficient monitoring in wireless sensor networks. In *Proc. 3rd Int'l. Symp. Information Processing in Sensor Networks (IPSN)*, pages 424–432, 2004.

[2] P. K. Agarwal and J. Erickson. Geometric range searching and its relatives. In B. Chazelle, J. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, pages 1–56. American Math. Soc., 1999.

[3] A. L. Buchsbaum, H. Karloff, C. Kenyon, N. Reingold, and M. Thorup. *OPT* versus *LOAD* in dynamic storage allocation. *SIAM J. Computing*, 33(3):632–46, 2004.

[4] S. Dasika, S. Vrudhula, K. Chopra, and R. Srinivasan. A framework for battery-aware sensor management. In *Proc. Design, Automation and Test in Europe Conf. and Expos. (DATE)*, pages 1–6, 2004.

[5] M. de Berg and O. Schwarzkopf. Cuttings and applications. *Int'l. J. Comp. Geom. & Appl.*, 5(4):343–355, 1995.

[6] U. Feige, M. M. Halldórsson, G. Kortsarz, and A. Srinivasan. Approximating the domatic number. *SIAM J. Computing*, 32(1):172–195, 2002.

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

[8] C. Kenyon and E. Remila. A near-optimal solution to a two-dimensional cutting stock problem. *Math. of Op. Res.*, 25(4):645–656, 2000.

[9] M. Perillo and W. Heinzelman. Optimal sensor management under energy and reliability constraints. In *Proc. IEEE Wireless Communications and Networking Conf. (WCNC)*, pages 1621–1626, 2003.

[10] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In *Proc. IEEE Int'l. Conf. on Communications (ICC)*, pages 472–476, June 2001.

A Details for Section 2

A.1 Proof of Lemma 2.1 *Given a set Y of unit-duration sensors, all live at some fixed x-coordinate x_0, an integer* group-duration parameter *D, and a sufficiently small positive ϵ, there is a set G of groups, each of duration D, such that for any i, $L_G(i) > L_Y(i)/(1+\epsilon) - 4D\lceil 1/\epsilon \rceil$.*

Proof: It is convenient to view a sensor s as a point $(\ell(s), r(s))$ in the plane. Note that all sensors live at x_0 are inside the rectangle $R_{x_0} = \{(x,y) : x \le x_0 \le y\}$. (See Figure 3.) First we partition the sensors of Y into strips by repeating the following as long as sensors remain.

(1) Create a vertical strip containing the at most $D\lceil 1/\epsilon \rceil$ sensors that remain with the smallest $\ell(\cdot)$ values.

(2) Create a horizontal strip containing the at most $D\lceil 1/\epsilon \rceil$ sensors that remain with the largest $r(\cdot)$ values.

Now for every vertical strip of Y, take the sensors in order of decreasing $r(\cdot)$ value in groups of size D. (We may discard the last $< D$ sensors in the last strip.) Similarly, for every horizontal strip, take the sensors in order of increasing $\ell(\cdot)$ value in groups of size D. (We may discard the last $< D$ sensors in the last strip.) Replace each group X with a larger rectangle s_X with $\ell(s_X) = \max_{s\in X} \ell(s)$, $r(s_X) = \min_{s\in X} r(s)$, and $d(s_X) = \sum_{s\in X} d(s) = D$.

Consider any $i \le x_0$ (the case $i > x_0$ is symmetric), and examine Figure 3(c). All sensors live at i are inside the rectangle $R_i = \{(x,y) : x \le i \le y\}$. Assume that the line $x = i$ intersects k horizontal strips; then R_i entirely contains at least $k-1$ vertical strips, so $L_Y(i) \ge (k-1)D\lceil 1/\epsilon \rceil$. For any group completely inside R_i, it contributes D to both $L_Y(i)$ and $L_G(i)$; for any group completely outside R_i, it does not contribute anything to either $L_Y(i)$ or $L_G(i)$. So only the groups in the k horizontal strips and the single vertical strip intersected by the line $x = i$ contribute to the difference, that is, $L_Y(i) - L_G(i) < kD + D\lceil 1/\epsilon \rceil + D$, where the last term accounts for the fewer than D sensors that we did not group in the last strip. Therefore,

$$\begin{aligned} L_G(i) &> L_Y(i) - (k-1)D - (2 + \lceil 1/\epsilon \rceil)D \\ &\ge (1-\epsilon)L_Y(i) - 2D\lceil 1/\epsilon \rceil \\ &\ge \frac{1}{1+2\epsilon} L_Y(i) - 2D\lceil 1/\epsilon \rceil, \end{aligned}$$

for any $\epsilon \le 1/2$. Replacing ϵ with $\epsilon/2$ gives the desired result. □

A.2 Proof of Theorem 2.3 *For any sufficiently small positive ϵ, there is an algorithm that runs in $\mathrm{poly}(n, 1/\epsilon)$ time and gives a schedule to the* RSC *problem with duration at least $L/(1+\epsilon) - O\left(\gamma d_{\max}/\epsilon^4\right)$.*

Proof: We are going to apply Steps (1)–(3) of Algorithm 1 repeatedly, grouping the smaller sensors so as to increase $d_{\min}$ until $\log(d_{\max}/d_{\min})$ becomes small enough that we can apply Theorem 2.2 to the resulting rectangles.

For ease of presentation, we assume that $1/\epsilon$ is an integer. Let r denote the ratio $d_{\max}/d_{\min}$. Assume first that $\log r \ge 1/\epsilon$, and set $\mu = \epsilon/\log r$ and $D = \lceil \mu^4 d_{\max} \rceil$. Apply Steps (1)–(3) of Algorithm 1 to $\mathcal{S}_s$, the set of sensors of duration at most $d'_{\max} = \lceil \mu D \rceil$, with group duration D and error parameter μ. This yields a set of rectangles G_s of duration D such that for any i and some constant c_1

$$\begin{aligned} L_{G_s}(i) &> \frac{1}{1+\mu} L_{\mathcal{S}_s}(i) - O\left(\frac{\gamma d'_{\max} \log(d'_{\max}/d_{\min})}{\mu^3}\right) \\ &> \frac{1}{1+\mu} L_{\mathcal{S}_s}(i) - O(\gamma \mu^2 d_{\max} \log(\mu^5 r)) \\ &> \frac{1}{1+\mu} L_{\mathcal{S}_s}(i) - \frac{c_1 \gamma \epsilon^2 d_{\max}}{\log r}. \end{aligned}$$

Now consider G_s as a set of sensors and the new problem instance $\mathcal{S}' = G_s \cup (\mathcal{S} \setminus \mathcal{S}_s)$. Its load at i is

$$L_{\mathcal{S}'}(i) > \frac{L(i)}{1+\mu} - \frac{c_1 \gamma \epsilon^2 d_{\max}}{\log r}.$$

Moreover, the new minimum duration of this problem instance is at least $d'_{\max}$, and the maximum duration remains $d_{\max}$, so the new ratio is $r' \le \frac{d_{\max}}{d'_{\max}} \le \frac{1}{\mu^5} = \frac{\log^5 r}{\epsilon^5} \le \log^{10} r$, since $\log r \ge 1/\epsilon$. For ϵ sufficiently small, we have $r' \le \sqrt{r}$; hence $\log r' \le \frac{1}{2} \log r$.

Iterate the above procedure, each time using new error parameter $\mu' = \epsilon/\log r'$, until it yields a problem instance $\mathcal{S}^*$ with minimum duration $d^*_{\min}$ for which $r^* = d_{\max}/d^*_{\min}$ is such that $\log r^* < 1/\epsilon$. Let $r_0, \ldots, r_k = r^*$ be the sequence of ratios and $L_0(i) = L(i), L_1(i), \ldots, L_k(i) =$

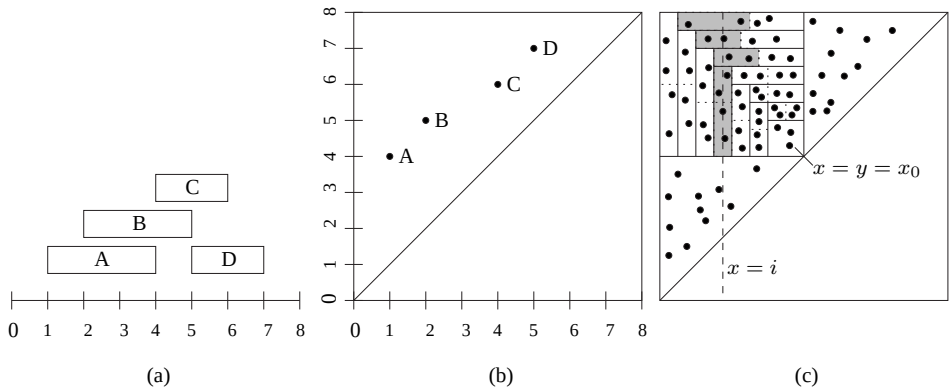

Figure 3: [3] (a) Four sensors. (b) The sensors of (a) viewed as (x, y) points. (c) Grouping a set of sensors with $D = 2$ and $\epsilon = 1/2$. The rectangle R_{x_0} contains the set Y of sensors. The sensors are first partitioned into alternating vertical and horizontal strips of $D\lceil 1/\epsilon \rceil = 4$ each. Within each strip, the sensors are grouped (dotted lines) into groups of $D = 2$. The groups that intersect the line $x = i$ are shaded.

$L^*(i)$ be the sequence of loads. For some constant c_2

$$
\begin{aligned}
L^*(i) &> \frac{1}{1+\epsilon/\log r_{k-1}} L_{k-1}(i) - \frac{c_1\gamma\epsilon^2 d_{\max}}{\log r_{k-1}} \\
&> \frac{1}{1+\epsilon/\log r_{k-1}} \times \\
&\quad \left(\frac{1}{1+\epsilon/\log r_{k-2}} L_{k-2}(i) - \frac{c_1\gamma\epsilon^2 d_{\max}}{\log r_{k-2}} \right) - \\
&\quad \frac{c_1\gamma\epsilon^2 d_{\max}}{\log r_{k-1}} \\
&\quad \vdots \\
&> \left(\prod_{i=0}^{k-1} \left(1 + \frac{\epsilon}{\log r_i} \right) \right)^{-1} L(i) - \\
&\quad \sum_{i=0}^{k-1} \left(\frac{1}{\log r_i} \right) c_1\gamma\epsilon^2 d_{\max} \\
&\geq \frac{1}{1+c_2\epsilon/\log r^*} L(i) - \frac{2}{\log r^*} c_1\gamma\epsilon^2 d_{\max} \\
&> L(i)/(1+2c_2\epsilon^2) - 4c_1\gamma\epsilon^3 d_{\max}.
\end{aligned}
$$

Let $L^* = \min_p L_{\mathcal{S}^*}(i)$. Finally, apply Theorem 2.2 to $\mathcal{S}^*$, which yields a schedule of duration at least

$$
\frac{1}{1+\epsilon} L^* - O\left(\frac{\gamma d_{\max} \log r^*}{\epsilon^3} \right) \geq \frac{1}{1+c_4\epsilon} L - O\left(\frac{\gamma d_{\max}}{\epsilon^4} \right),
$$

for some constant c_4. Replacing ϵ with ϵ/c_4 gives the desired result. □