The Maximum Exposure Problem

Neeraj Kumar*[1], Stavros Sintos†[2], and Subhash Suri‡[1]

[1]Department of Computer Science, University of California, Santa Barbara, USA
[2]Duke University, Durham, NC, USA

Abstract

Given a set of points P and axis-aligned rectangles $\mathcal{R}$ in the plane, a point $p \in P$ is called *exposed* if it lies outside all rectangles in $\mathcal{R}$. In the *max-exposure problem*, given an integer parameter k, we want to delete k rectangles from $\mathcal{R}$ so as to maximize the number of exposed points. We show that the problem is NP-hard and assuming plausible complexity conjectures is also hard to approximate even when rectangles in $\mathcal{R}$ are translates of two fixed rectangles. However, if $\mathcal{R}$ only consists of translates of a single rectangle, we present a polynomial-time approximation scheme. For range space defined by general rectangles, we present a simple $O(k)$ bicriteria approximation algorithm; that is by deleting $O(k^2)$ rectangles, we can expose at least $\Omega(1/k)$ of the optimal number of points.

1 Introduction

Let $S = (P, \mathcal{R})$ be a geometric set system, also called a *range space*, where P is a set of points and each $R \in \mathcal{R}$ is a subset of P, also called a range. We are primarily interested in range spaces defined by a set of points in two dimensions and ranges defined by axis-aligned rectangles. We say that a point $p \in P$ is *exposed* if no range in $\mathcal{R}$ contains p. The *max-exposure* problem is defined as follows: given a range space $(P, \mathcal{R})$ and an integer parameter $k \geq 1$, remove k ranges from $\mathcal{R}$ so that a maximum number of points are exposed. That is, we want to find a subfamily $\mathcal{R}^* \subseteq \mathcal{R}$ with $|\mathcal{R}^*| = k$, so that the number of exposed points in the (reduced) range space $(P, \mathcal{R} \setminus \mathcal{R}^*)$ is maximized.

The max-exposure problem arises naturally in many geometric coverage settings. For instance, if points are the location of clients in the two-dimensional plane, and ranges correspond to coverage areas of facilities, then exposed points are those not covered by any facility. The max-exposure problem in this case gives a worst-case bound on the number of clients that can be exposed if an adversary disables k facilities. Similarly, in distributed sensor networks, ranges correspond to *sensing zones*, points correspond to physical assets being monitored by the network, and the max-exposure problem computes the number of assets exposed when k sensors are compromised.

More broadly, the max-exposure problem is related to the densest k-subgraph problem in *hypergraphs*. In the *densest k-subhypergraph* problem, we are given a hypergraph $H = (X, E)$, and we want to find a set of k vertices with a maximum number of induced hyperedges. In general hypergraphs, finding k-densest subgraphs is known to be (conditionally) hard to approximate within

*neeraj@cs.ucsb.edu †ssintos@cs.duke.edu ‡suri@cs.ucsb.edu

a factor of $n^{1-\epsilon}$, where n is the number of vertices. The max-exposure problem is equivalent to the densest k-subhypergraph problem on a *dual* hypergraph, whose vertices X corresponds to the ranges $\mathcal{R}$, and whose hyperedges correspond to the set of points P. Specifically, each point $p \in P$ corresponds to a hyperedge adjacent to the set of ranges containing the point p. In the rest of the paper, we will use $n = |\mathcal{R}|$ for the number of ranges in $\mathcal{R}$ and $m = |P|$ for the number of points. We show that if the range space is defined by *convex polygons*, then the max-exposure problem is just as hard as the densest k-subhypergraph problem. However, for ranges defined by *axis-aligned rectangles*, one can achieve better approximation. In particular, we obtain the following results.

- We show that the max-exposure problem is NP-hard and assuming the *dense vs random* conjecture [1], it is also hard to approximate better than a factor of $O(n^{1/4})$ even if the range space is defined by *only two types of rectangles* in the plane. For range space defined by convex polygons, we show that max-exposure is equivalent to densest k-subhypergraph problem, which is hard to approximate within a factor of $O(n^{1-\epsilon})$.

- When ranges are defined by translates of a *single* rectangle, we give a polynomial-time approximation scheme (PTAS) for max-exposure. The PTAS stands in sharp contrast to the inapproximability of ranges defined by *two* types of rectangles. Moreover, as an easy consequence of this result, we obtain a constant approximation when the ratio of longest and smallest side of rectangles in $\mathcal{R}$ is bounded by a constant. However, we do not know if max-exposure with translates of a single rectangle can be solved in polynomial time or is NP-hard.

- For ranges defined by arbitrary rectangles, we present a simple greedy algorithm that achieves a bicriteria $O(k)$-approximation. That is, if the optimal number of points exposed is m^*, the algorithm picks a subset of k^2 rectangles such that the number of points exposed is at least m^*/ck, for some constant c. No such approximation is possible for general hypergraphs. If rectangles in $\mathcal{R}$ have a bounded aspect ratio, the approximation improves to $O(\sqrt{k})$. For pseudodisks with *bounded-ply* (no point in the plane is contained in more than a constant number of ranges), this algorithm achieves a constant approximation.

The PTAS is obtained by first optimally solving a restricted max-exposure instance where all points are contained in a unit square using dynamic programming in polynomial time. Next, we carefully combine them to obtain an optimal solution in $(nm)^{O(h^2)}$ time for the case when input points lie in a $h \times h$ square. Applying well known shifting techniques on this gives us the PTAS. Both bicriteria algorithms are obtained by carefully assigning the points to ranges and applying greedy strategies.

Related Work Coverage and exposure problems have been widely studied in geometry and graphs. In the classical *set cover* problem, we want to select a subfamily of k sets that cover the maximum number of items (points) [2, 3]. For the set cover problem, the classical greedy algorithm achieves a factor $\log n$ approximation for the number of sets needed to cover all the items, or factor $(1 - 1/e)$ approximation for the number of items covered by using exactly k sets. Similarly, in geometry, the art gallery problems explore coverage of polygons using a minimum number of guards. Unlike coverage problems where greedy algorithms deliver reasonably good approximation, the exposure problems turn out to be much harder. Specifically, choosing k sets whose union is of *minimum size* is much harder to approximate with a conditional inapproximability of $O(n^{1-\epsilon})$ where n is the number of elements, or $O(m^{1/4-\epsilon})$ where m is the number of sets [1]. This so-called *min-union* problem is essentially the complement of the densest k-subhypergraph problem on hypergraphs [4]. The densest k-subgraph problem for graphs has a long history [5–8]. The paper [4] also studies the special case of an *interval hypergraph* $H = (V, E)$, whose vertices V is a finite subset of $\mathbb{N}$ and for each edge $e \in E$ there are values $a_e, b_e \in \mathbb{N}$ such that $e = \{i \in V : a_e \le i \le b_e\}$. That is, vertices are integer points and edges are intervals containing them. They show that this restricted case can be solved in polynomial time. The corresponding max exposure instance is when ranges $\mathcal{R}$ are intervals $R_i = (a_i, b_i)$ on the real line. As discussed later, this 1-D case can also be solved in polynomial time. Moreover, we show that good approximations can also be obtained for some geometric objects in two dimensions.

The coverage problems have also been studied for geometric set systems where improved approximation bounds are possible using the VC dimension [9–11]. Multi-cover variants, where each input point must be covered by more than one set, are studied in [12, 13]. The geometric constraint removal problem [14, 15], where given a set of ranges, the goal is to *expose* a path between two given points by deleting at most k ranges (a path is exposed if it lies in the exterior of all ranges), is also closely related to the max-exposure problem. Even for simple shapes such as unit disks (or unit squares) [16, 17], no PTAS is known for this problem.

The remainder of the paper is organized as follows. In Section 2, we discuss our hardness results followed by the bicriteria $O(k)$-approximation in Section 3. In Section 4, we study the case when $\mathcal{R}$ consists of translates of a fixed rectangle and describe a PTAS for it. Finally, in Section 5, we use these ideas to obtain a bicriteria $O(\sqrt{k})$-approximation when the aspect ratio of rectangles in $\mathcal{R}$ is bounded by a constant.

2 Hardness of Max-Exposure

We show that the max-exposure problem for geometric ranges is both NP-hard, and inapproximable. We begin by reducing the densest k-subgraph on bipartite graphs (*bipartite-DkS*) to the max-exposure problem; the known NP-hardness of biparite-DkS then implies the hardness for max-exposure. Moreover, we show that bipartite-DkS is hard to approximate assuming the *dense vs random* conjecture, thereby establishing the inapproximability of max-exposure.

In the *bipartite-DkS* problem, we are given a bipartite graph $G = (A, B, E)$, an integer k, and we want to compute a set of k vertices such that the induced subgraph on those k vertices has the maximum number of edges. Given an instance $G = (A, B, E)$ of bipartite-DkS, we construct a max-exposure instance as follows.

Let $R_1 = [0, \epsilon] \times [0, n]$ be a thin vertical rectangle and $R_2 = [0, n] \times [0, \epsilon]$ be a thin horizontal rectangle. For each vertex $v_i \in A$, we create a copy R_i of R_1, and place it such that its lower-left corner is at $(i, 0)$. Similarly, for each vertex $v_j \in B$, we create a copy R_j of R_2, and place it such that its lower-left corner is at $(0, j)$. These $|A| + |B|$ rectangles create a checkerboard arrangement, with $|A| \times |B|$ cells of intersection. For each edge $(v_i, v_j) \in E$, we place a single point in the cell corresponding to intersection of R_i and R_j. It is now easy to see that G has a k-subgraph with m^* edges if and only if we can expose m^* points in this instance by removing k rectangles: the removed rectangles are exactly the k vertices chosen in the graph, and each exposed point corresponds to the edge included in the output subgraph. (See also Figure 1.) We will later make use of this reduction, and therefore state it as the following lemma.

Lemma 1. *The max-exposure problem is at least as hard as bipartite-DkS.*

Since bipartite-DkS is known to be NP-hard [18], we have the following.

Theorem 1. *Max-exposure problem with axis-aligned rectangles is* NP*-hard.*

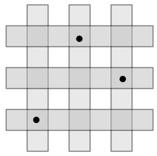

Figure 1: Reducing bipartite-DkS to max-exposure with axis-aligned rectangles.

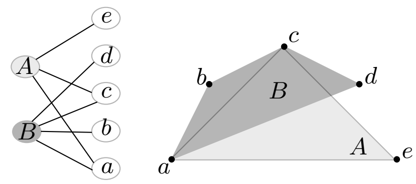

Figure 2: Reducing densest k-subhypergraph problem to max-exposure. Hypergraph vertices A, B shown as convex ranges.

2.1 Hardness of Approximation

The construction in the preceding proof shows that max-exposure with rectangles is at least as hard as bipartite-DkS problem. Moreover, the geometric construction uses translates of *only two rectangles* R_1, R_2. In the following, we show that even with such a restricted range space, the problem is also hard to approximate. To that end we prove that bipartite-DkS cannot be approximated better than a factor $O(n^{1/4})$, where n is the number of vertices in this graph. More precisely, if the densest subgraph over k vertices has m^* edges, it is hard to find a subgraph over k vertices that contains $\Omega(m^*/n^{\frac{1}{4}-\epsilon})$ edges in polynomial time. This hardness of approximation is conditioned on the so-called *dense vs random* conjecture [1] stated as follows.

Given a graph G, constants $0 < \alpha, \beta < 1$, and a parameter k, we want to distinguish between the following two cases.

1. (Random) $G = G(n, p)$ where $p = n^{\alpha-1}$, that is, G has average degree approximately n^α.
2. (Dense) G is adversarially chosen so that the densest k-subgraph of G has average degree k^β.

The conjecture states that for all $0 < \alpha < 1$, sufficiently small $\epsilon > 0$, and for all $k \leq \sqrt{n}$, one cannot distinguish between the *dense* and *random* cases in polynomial time (w.h.p), when $\beta \leq \alpha - \epsilon$.

In order to obtain hardness guarantees using the above conjecture, one needs to find the 'distinguishing ratio' r, that is the least multiplicative gap between the optimum solution for the problem on the dense and random instances. If there exists an algorithm with an approximation factor significantly smaller than r, then we would be able to use it to distinguish between the dense and random instances, thereby refuting the conjecture. We obtain the following result for densest k-subgraph problem on bipartite graphs.

Lemma 2. *Assuming that* dense vs random *conjecture is true, the densest k-subgraph problem on bipartite graphs is hard to approximate better than a factor $O(n^{1/4})$ of optimum.*

Proof. Let $G' = (V', E')$ be a graph sampled either from the dense or from the random instances. We construct a bipartite graph $G = (A, B, E)$ as follows. For every vertex $v \in V'$, add a vertex v_a to A and v_b to B. For every edge $e = (u, v) \in E'$, we add the pair of edges $e_1 = (u_a, v_b)$ and $e_2 = (v_a, u_b)$ to E. That is, every edge $e \in E'$ is mapped to two copies $e_1, e_2 \in E$ and we define e to be their *parent* edge as $par(e_1) = par(e_2) = e$. Similarly, for a vertex $u \in V'$ and its two copies $u_a, u_b \in V$, we define $par(u_a) = par(u_b) = u$. We say that G is *dense* if the underlying graph G' was sampled from the dense case, otherwise we say that G is *random*.

Consider a set of $k^* = 2k$ vertices in G. If G came from the dense case, there must be a set of $2k$ vertices that have $2k^{\beta+1}$ edges between them. So the number of edges in dense case $m^*_d \geq 2k^{\beta+1}$. Otherwise, we are in the random case. Consider the optimal set of $k^* = 2k$ vertices V^* that maximizes the set E^* of edges in the induced subgraph $G[V^*]$. Now consider the corresponding set of vertices $V_p = \{par(v) \mid v \in V^*\}$ of the original graph G' and the set of edges E_p in the induced subgraph $G'[V_p]$). We have that $|V_p| \leq |V^*| = 2k$ and $|E_p| \geq |E^*|/2$ because for each edge $e = (u, v) \in E^*$, we will have the edge $par(e) = (par(u), par(v)) \in E_p$. Since $|V_p| \leq 2k$ and we are in the random case, we can upperbound the number of edges in E_p as the number of edges in the densest subgraph of $G(n, n^{\alpha-1})$ over $2k$ vertices. This is $\tilde{O}(\max(2k, 4k^2 n^{\alpha-1}))$ w.h.p. where $\tilde{O}$ ignores logarithmic factors. Therefore the optimum number of edges in the random case is $m_r^* = |E^*| \leq 2|E_p| = \tilde{O}(\max(k, k^2 n^{\alpha-1}))$ w.h.p.

Choosing $k = n^{1/2}$, $\alpha = \frac{1}{2}$, $\beta = \frac{1}{2} - \epsilon$, gives us $m_r^* = \tilde{O}(n^{1/2})$ w.h.p. and $m_d^* = \tilde{\Omega}(n^{\frac{3-2\epsilon}{4}})$. If we could approximate this problem within a factor $O(n^{1/4-\epsilon})$, then in the dense case, the number of edges computed by this approximation algorithm is $\tilde{\Omega}(n^{\frac{1+\epsilon}{2}})$ which is strictly more than the maximum possible edges in the random case. Therefore, we would be able to distinguish between dense and random cases, and thereby refute the conjecture for these values of α, β and k. □

Using the same construction as in Lemma 1, we obtain the following.

Corollary 2. *Assuming the dense vs random conjecture, max-exposure with axis-aligned rectangles is hard to approximate better than factor $O(n^{1/4})$ of optimum.*

2.2 Hardness of Max-exposure with Convex Polygons

We now show that the max-exposure problem is *equivalent* to the densest k-subhypergraph problem for *general hypergraphs* when the range space $(P, \mathcal{R})$ is defined by convex polygons. In one direction, the max-exposure instance $(P, \mathcal{R})$ naturally corresponds to a hypergraph $H = (\mathcal{R}, P)$ whose vertices are the ranges and the edges correspond to points and are defined by the containment relationship. Clearly, the densest k-subhypergraph corresponds to the set of k ranges deleting which exposes maximum number of points. For the other direction, we have the following lemma. (See also Figure 2.)

Lemma 3. *Given a hypergraph $H = (X, E)$, one can construct a max-exposure instance with convex ranges $\mathcal{R}$ and points P such that the densest k-subhypergraph of H corresponds to a solution of max-exposure.*

Proof. For each edge $e \in E$ of the hypergraph, add a point $p_e \in P$. We place all the points of P in convex position. Let $v \in X$ be a vertex and E_v be the set of hyperedges adjacent to v. Since points in P are in convex position, any subset of P forms a convex polygon. Therefore, for every $v \in V$, we can draw a convex polygon $R_v \in \mathcal{R}$ whose corners are the point set corresponding to the hyperedges E_v. The polygons will likely overlap in the convex region but for every point $p_e \in P$, the polygons containing p_e are precisely the ones that have p_e as its corner. Therefore, p_e is exposed if and only if all vertices of the hyperedge e are selected. □

3 A Bicriteria $O(k)$-approximation Algorithm

In this section, we present a simple approximation algorithm for the max-exposure problem that achieves bicriteria $O(k)$-approximation for range spaces defined by arbitrary axis-aligned rectangles. Specifically, if the optimal number of points exposed is m^*, the algorithm picks a subset of k^2 rectangles such that the number of points exposed is at least m^*/ck, for some constant c. In fact, the results hold for any polygonal range with $O(1)$ complexity.

This bicriteria approximation should be contrasted with the fact that no such approximation is possible for the densest k-subhypergraph problem: that is, one cannot compute a set of $O(k^b)$ vertices for any constant b such that the number of edges in the induced subhypergraph is at least optimal. Thus the geometric properties of the range space have a significant impact on the problem complexity. In particular, if $\mathcal{R}$ consists of rectangle ranges, we show that the following strategy picks a subset of αk ranges such that the number of points exposed is at least $\alpha m^*/(ck^2)$, for a parameter $1 \leq \alpha \leq k$ and constant c that will be fixed later. Choosing $\alpha = k$ gives us the claimed bound.

Our algorithm is essentially greedy. We divide the points into maximal equivalence classes, where each class is the maximal subset of points belonging to the same subset of ranges. We define $\mathcal{R}(p)$ as the set of ranges that contain a point $p \in P$, and remove all points that are contained in more than k ranges, since they can be never exposed in the optimal solution. Therefore, without loss of generality, we can assume that $|\mathcal{R}(p)| \leq k$ for all points $p \in P$. The rest of the algorithms is as follows.

Algorithm 1 Greedy-Bicriteria

---

1. Partition P into a set $\mathcal{G}$ of groups where each group $G_i \in \mathcal{G}$ is an equivalence class of points that are contained in the same set of ranges. That is, for any $p \in G_i$, $p' \in G_j$, we have $\mathcal{R}(p) = \mathcal{R}(p')$ if $i = j$ and $\mathcal{R}(p) \neq \mathcal{R}(p')$, otherwise.
2. Sort the groups in $\mathcal{G}$ by decreasing order of their size $|G_i|$ and select the ranges in first α groups for deletion.
3. Return $m' = \sum_{1 \leq i \leq \alpha} |G_i|$ as the number of points exposed.

In Algorithm 1, observe that every point in the ith group G_i is contained in the same set of ranges, which we denote by $\mathcal{R}(G_i)$. Moreover, we have $|\mathcal{R}(G_i)| \leq k$. Therefore, the total number of ranges that we delete in Step 2 is at most αk. It remains to show that the number of points exposed m' is at least $\alpha m^*/ck^2$.

Lemma 4. *Let m' be the number of points exposed by the algorithm* Greedy-Bicriteria*, and let m^* be the optimal number of exposed points, Then, $m' \geq \alpha m^*/ck^2$.*

Proof. Consider the optimal set $\mathcal{R}^*$ of k ranges that are deleted, and let P^* be the set of exposed points. We partition the set of points P^* into groups $\mathcal{G}^*$ as before, such that each group $G_i^* \in \mathcal{G}^*$ is identified by the range set $\mathcal{R}(G_i^*) = \mathcal{R}(p)$, for any $p \in G_i^*$. Since $P^* \subseteq P$, we must have that $\mathcal{G}^* \subseteq \mathcal{G}$. This holds because for every group $G_i^* \in \mathcal{G}^*$ there must be a group $G_j \in \mathcal{G}$ such that $\mathcal{R}(G_i^*) = \mathcal{R}(G_j)$. Moreover since P^* is the maximum set of points that can be exposed, we must have that $G_i^* = G_j$. Finally, we note that the number of groups $|\mathcal{G}^*|$ is bounded by the number of cells in the arrangement of ranges in $\mathcal{R}^*$ which is at most ck^2 for some fixed constant c, for all $O(1)$-complexity ranges. If the groups in $\mathcal{G}$ are arranged by decreasing order of their sizes, we have that

$$
\begin{aligned}
m^* &= \sum_{1 \leq i \leq |\mathcal{G}^*|} |G_i^*| \leq \sum_{1 \leq i \leq |\mathcal{G}^*|} |G_i| \leq \sum_{1 \leq i \leq ck^2} |G_i| \\
&\leq \frac{ck^2}{\alpha} \sum_{1 \leq i \leq \alpha} |G_i| = \frac{ck^2}{\alpha} \cdot m'
\end{aligned}
$$

□

The parameter α can be tuned to improve the approximation guarantee with respect to one criterion (say the number of exposed points) at the cost of other. With $\alpha = k$, the algorithm exposes at least $\Omega(m^*/k)$ by removing k^2 ranges. As for the running time, a simple implementation of the algorithm can be made to run in $O(mn \log m)$ time: we can build the point-range containment relation in $O(mn)$ time, partitioning the point set into groups takes an additional $O(mn \log m)$ time.

3.1 Constant Approximation for Pseudodisks with Bounded-ply

If the range space $\mathcal{R}$ consists of pseudodisks of *bounded-ply* (no point in the plane is contained in more than a constant number ρ pseudodisks), then the algorithm Greedy-Bicriteria achieves a constant approximation. Due to the bounded-ply restriction, we have that the number of pseudodisks containing the points of group G_i is $|\mathcal{R}(G_i)| \leq \rho$, and therefore number of pseudodisks that are removed in Step 2 of the algorithm is also at most $\alpha\rho$. Moreover, the number of cells in an arrangement of k pseudodisks with depth at most ρ is $O(\rho k)$ [19]. Therefore, we can bound the number of groups of the optimal solution $|\mathcal{G}^*|$ in the proof for Lemma 4 to be at most $c\rho k$. This gives us that the number of points exposed $m' \geq \alpha m^*/c\rho k$, where m^* is the number of points exposed by the optimal solution.

Lemma 5. *If the range space $\mathcal{R}$ consists of pseudodisks of* bounded-ply ρ, *then algorithm* Greedy-Bicriteria *exposes at least $\alpha m^*/c\rho k$ points by deleting at most $\alpha\rho$ pseudodisks, where $1 \leq \alpha \leq k$.*

Choosing $\alpha = k$, the algorithm achieves a bicriteria $O(\rho)$-approximation. With $\alpha = k/\rho$, the algorithm exposes at least $1/c\rho^2$ fraction of the optimal number of points by deleting k ranges.

4 A PTAS for Unit Square Ranges

We have seen that max-exposure is hard to approximate even if the ranges are translates of two types of rectangles. We now describe an approximation scheme when the ranges are translates of a *single* rectangle. In this case, we can scale the axes so that the rectangle becomes a *unit square* without changing any point-rectangle containment. Therefore, we can assume that our ranges are all unit squares. The problem is non-trivial even for unit square ranges, and as a warmup we first solve the following special case: *all the points lie inside a unit square.* We develop a dynamic programming algorithm to solve this case exactly, and then use it to design an approximation for the general set of points.

4.1 Exact Solution in a Unit Square

We are given a max-exposure instance consisting of unit square ranges $\mathcal{R}$ and a set of points P in a unit square C. Without loss of generality, we can assume that the lower left corner of C lies at origin $(0, 0)$ and all ranges in $\mathcal{R}$ intersect C. We classify the ranges in $\mathcal{R}$ to be one of the two types: (See also Figure 3).

Type-0: Unit square ranges that intersect $x = 0$.

Type-1: Unit square ranges that intersect $x = 1$.

(A unit square range coincident with both $x = 0$ and $x = 1$ is assumed to be Type-0). We draw two parallel horizontal lines $\ell_0 : y = 0$ and $\ell_1 : y = 1$ coincident with bottom and top horizontal sides of C respectively. We say that a range $R \in \mathcal{R}$ is *anchored* to a line ℓ if it intersects ℓ. Note that every $R \in \mathcal{R}$ is anchored to exactly one of ℓ_0 or ℓ_1. (When R is coincident with both ℓ_0 and ℓ_1, we say that it is anchored to ℓ_0).

Moreover, for the rest of our discussion, let $x = x_i$ be a vertical line and define $P_i \subseteq P$ to be the set of points that have x-coordinate at least x_i. In other words, P_i is the set of *active points* at $x = x_i$. Similarly, define $\mathcal{R}_i \subseteq \mathcal{R}$ to be the set of *active ranges* that have at least one corner to the right of $x = x_i$. That is, $R \in \mathcal{R}_i$ either intersects $x = x_i$ or lies completely to the right of it.

In order to gain some intuition, we will first consider the following two natural dynamic programming formulations for the problem.

DP-template-0 Suppose that the points in P are ordered by their increasing x-coordinates and let x_i be the x-coordinate of the ith point p_i. We define a subproblem as $S(i, k', \mathcal{R}_d)$ which represents the maximum number of points in P_i that can be exposed by removing k' ranges from the range set $\mathcal{R}_i \setminus \mathcal{R}_d$. If we define $x_0 = 0$, then $S(0, k, \emptyset)$ gives the optimal number of exposed points for our problem.

Let $k_i = |\mathcal{R}(p_i) \setminus \mathcal{R}_d|$ be the number of new ranges in $\mathcal{R}_i$ that contain p_i. Then, we can can express the subproblems at i in terms of subproblems at $i+1$ as follows.

$$S(i,\ k', \mathcal{R}_d) = \max \begin{cases} S(i+1,\ k'-k_i,\ \mathcal{R}_d \cup \mathcal{R}(p_i))\ +\ 1 & \text{expose } p_i \\ S\left(i+1,\ k',\ \mathcal{R}_d\right) & p_i \text{ not exposed} \end{cases}$$

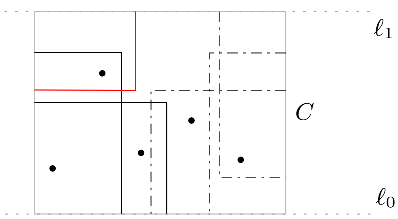

Figure 3: Max-exposure in a unit square C. Type 0 ranges are drawn with solid lines, Type 1 ranges are dash-dotted.

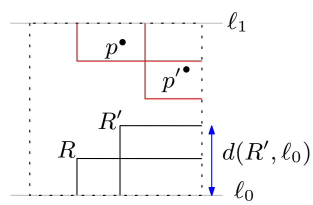

Figure 4: An example of closer *relationship. Point p is closer to ℓ_1 than p'. R is closer to ℓ_0 than R'.*

Roughly speaking, at $x = x_i$ which is the *event* corresponding to a point $p_i \in P$, we have two choices : *expose* p_i or *do not expose* p_i. If we expose p_i, we pay for deleting the ranges in $\mathcal{R}_i \setminus \mathcal{R}_d$ that contain p_i and mark them as deleted by adding to the *deleted range set* $\mathcal{R}_d$. It is easy to see that this correctly computes the optimal number of exposed points since we charge for every deletion exactly once. However, there is one complication: a priori it is not clear how to bound the number of range subset $\mathcal{R}_d$ used by this dynamic program. We later argue that the geometry of range space for Type-0 ranges allows us to use only a polynomial number of choices.

DP-template-1 An alternative approach is to consider both *point* and *begin-range* events. That is, $x = x_i$ is either incident to a point $p_i \in P$ or to the left vertical side of a range $R_i \in \mathcal{R}$. Then, we can define a subproblem by the tuple $S(i, k', P_f)$ which represents the maximum number of points in $(P_i \setminus P_f)$ that can be exposed by removing k' ranges in $\mathcal{R}_i$. If we define $x_0 = 0$, then $S(0, k, \emptyset)$ gives the optimal number of exposed points. Let $P(R_i) \subseteq P$ be the set of points contained in the range R_i, then we have the following recurrence.

$$
\begin{aligned}
S(i,\ k', P_f) \ &= \ \max \begin{cases} S(i+1,\ k'-1,\ P_f) & \text{delete range } R_i \\ S(i+1,\ k',\ P_f \cup P(R_i)) & R_i \text{ not deleted} \end{cases} \\
&\qquad \textit{(event } x = x_i \textit{ was beginning of a range } R_i \in \mathcal{R}_i\textit{)} \\
&= \ \max \begin{cases} S(i+1,\ k', P_f) & \text{if } p_i \in P_f \text{, cannot expose } p_i \\ S(i+1,\ k', P_f) + 1 & \text{otherwise, expose } p_i \end{cases} \\
&\qquad \textit{(otherwise, event } x = x_i \textit{ was a point } p_i \in P_i\textit{)}
\end{aligned}
$$

In the above formulation, at each *begin-range* event for some $R_i \in \mathcal{R}_i$, we have two choices: *delete* R_i or *do not delete* R_i. If R_i was deleted, we reduce the budget k' by one. Otherwise, if R_i was not deleted, we can never expose the points in $P(R_i)$, and therefore we add $P(R_i)$ to the *forbidden point set* P_f. The correctness of the dynamic program follows from the fact that for every point p_i, all the ranges containing it must begin before $x = x_i$, and we expose p_i only if those ranges were *deleted*. Again, it is not obvious how many different subsets P_f are needed by the dynamic program. However, we will later show that by keeping track of polynomial number of sets P_f, we can solve max-exposure with Type-1 ranges.

We note that the Type-0 and Type-1 ranges may superficially seem symmetric but once we fix the order of computing subproblems, they become structurally different. Therefore, we would need slightly different techniques to handle each type. For the ease of exposition, we present dynamic programs for Type-0 and Type-1 ranges separately and finally combine them. Also note that if the ranges in $\mathcal{R}$ are intervals on the real line (max exposure in 1D), then both DP-template-0 and DP-template-1 can be easily applied to obtain a polynomial time algorithm.

We will now define the following ordering relations that will be useful later. Let ℓ be a horizontal line, and let $d(p, \ell)$ denote the orthogonal distance of $p \in P$ from ℓ. If $p, p' \in P$ are two points, we say that p is *closer* to ℓ than p' if $d(p, \ell) < d(p', \ell)$. Similarly, for a range $R \in \mathcal{R}$ that is anchored to ℓ, let $d(R, \ell)$ be the vertical distance *inside the unit square* C between ℓ and the side of R parallel to ℓ. If $R, R' \in \mathcal{R}$ are two ranges, we say that R is *closer* (or equivalently R' is *farther*) from ℓ if both R, R' are anchored to ℓ and $d(R, \ell) < d(R', \ell)$. (See Figure 4.)

4.1.1 Max-exposure with Type-0 Ranges

Recall that Type-0 ranges intersect the vertical lines $x = 0$ and are anchored to either ℓ_0 or ℓ_1. We will apply the formulation discussed in *DP-template-0*. The key challenge here is to bound the number of possible deleted range sets $\mathcal{R}_d$. Towards that end, we make the following claim. Recall that $\mathcal{R}_i$ is the set of active ranges at $x = x_i$.

Lemma 6. *Let q_0, q_1 be the two exposed points strictly to the left of $x = x_i$ that are closest to ℓ_0 and ℓ_1 respectively. Then our dynamic program only needs to consider the set of deleted ranges $\mathcal{R}_d = \mathcal{R}_i \cap (\mathcal{R}(q_0) \cup \mathcal{R}(q_1))$ at $x = x_i$ conditioned on q_0, q_1.*

Proof. Observe that since $\mathcal{R}$ consists of Type-0 ranges, every range in $\mathcal{R}_i$ must intersect the vertical line $x = x_i$. Suppose we partition $\mathcal{R}_i$ into ranges $\mathcal{R}_i^0$ that are anchored to ℓ_0 and $\mathcal{R}_i^1$ that are anchored to ℓ_1. Let $P' \subseteq P$ be the set of all *exposed points* strictly to the left of $x = x_i$. Observe that for all $p \in P'$, any range $R \in \mathcal{R}_i^0$ that contains p must also contain q_0. Therefore, we must have $\mathcal{R}_i^0 \cap \mathcal{R}(p) \subseteq \mathcal{R}_i^0 \cap \mathcal{R}(q_0)$, for all $p \in P'$. Similarly, $\mathcal{R}_i^1 \cap \mathcal{R}(p) \subseteq \mathcal{R}_i^1 \cap \mathcal{R}(q_1)$, for all $p \in P'$. This gives us $\bigcup_{p \in P'} \mathcal{R}_i \cap \mathcal{R}(p) = \mathcal{R}_i \cap (\mathcal{R}(q_0) \cup \mathcal{R}(q_1))$. Therefore, the set $\mathcal{R}_d$ consists of all the active ranges that contain at least one exposed point and were therefore deleted to the left of $x = x_i$. $\square$

Therefore, if our dynamic program remembers the exposed points q_0, q_1, then we can compute the deleted range set $\mathcal{R}_d = \mathcal{R}_i \cap (\mathcal{R}(q_0) \cup \mathcal{R}(q_1))$ at $x = x_i$. There are $O(m^2)$ choices for the pair q_0, q_1, so the number of possible sets $\mathcal{R}_d$ is also $O(m^2)$. We can therefore identify our subproblems by the tuple $S(i,\ k',\ q_0,\ q_1)$ which represents the maximum number of exposed points with x-coordinates x_i or higher using k' rectangles from the set $\mathcal{R}_i \setminus \mathcal{R}_d$. With $k_i = |\mathcal{R}(p_i) \setminus \mathcal{R}_d|$, we obtain the following recurrence:

$$S(i,\ k',\ q_0,\ q_1) \ = $$
$$\max \begin{cases} S\left(i+1,\ k'-k_i,\ \mathit{closer}(q_0,p_i),\ \mathit{closer}(q_1,p_i)\right) \ + \ 1 & \text{expose } p_i \\ S\left(i+1,\ k',\ q_0,\ q_1\right) & p_i \text{ not exposed} \end{cases}$$

where the function $\mathit{closer}(q_0, p_i)$ returns whichever of q_0, p_i is closer to ℓ_0, and $\mathit{closer}(q_1, p_i)$ returns whichever of q_1, p_i is closer to ℓ_1. The optimal solution is given by $S(0,\ k,\ q_0^*,\ q_1^*)$, where $q_0^* = (0,1)$ and $q_1^* = (0,0)$ are two artificial points with $\mathcal{R}(q_0^*) = \mathcal{R}(q_1^*) = \emptyset$ (not contained in any range). The base case is defined by the rightmost event at vertical line $x = 1$ and is initialized with zeroes for all q_0, q_1 and $k' \geq 0$. Any subproblem with $k' < 0$ has value $-\infty$.

4.1.2 Max-exposure with Type-1 Ranges

Next we consider the case when we only have Type-1 ranges in $\mathcal{R}$. Unfortunately in this case, our previous dynamic program does not work and we need to remember a different set of parameters. More precisely, we will apply the formulation discussed in *DP-template-1*, and bound the number of possible forbidden point sets P_f. Recall that P_i is the set of active points at $x = x_i$ (with x-coordinate x_i or higher).

Lemma 7. *Let Q_0, Q_1 be two ranges that begin to the left of $x = x_i$ and were* not deleted*. Moreover, Q_0 is anchored to and is farthest from ℓ_0. Similarly Q_1 is anchored to and is farthest from ℓ_1 (Figure 5). Then the forbidden point set at $x = x_i$ is given by $P_f = P_i \cap (P(Q_0) \cup P(Q_1))$, where $P(Q)$ is the set of points contained in range Q.*

Proof. Recall that the set $\mathcal{R}_i$ consists of ranges that have at least one corner to the right of the vertical line $x = x_i$. Since we are dealing with Type-1 ranges, every range that begins to the left of $x = x_i$ lies in $\mathcal{R}_i$. Now let $\mathcal{R}' \subseteq \mathcal{R}_i$ be the set of ranges that begin to the left of $x = x_i$ and were *not deleted*. Here P_i is the set of points in P that have x-coordinate x_i or higher. Now consider any range $R \in \mathcal{R}'$. Recall that R must be anchored to either ℓ_0 or ℓ_1. If R was anchored to ℓ_0, then every point of P_i that lies in R also lies in Q_0. Otherwise R was anchored to ℓ_1, so every point of P_i that lies in R also lies in Q_1. Therefore, $\bigcup_{R \in \mathcal{R}'} \ (P_i \cap P(R)) = P_i \cap (P(Q_0) \cup P(Q_1))$, which is precisely the forbidden point set P_f. $\square$

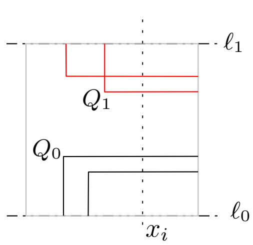

Figure 5: Undeleted ranges Q_0 and Q_1 farthest from ℓ_0 and ℓ_1 respectively.

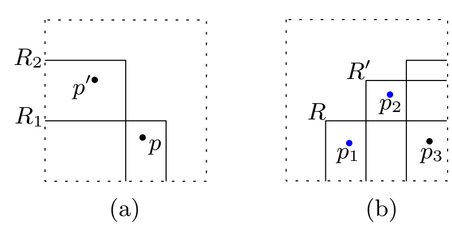

Figure 6: Remembering one of R_1, R_2 in (a) or one of p_1, p_2 in (b) is not sufficient.

Notation	Explanation
$\mathcal{R}(p)$	ranges containing point p
$P(R)$	points contained in range R
q_0, q_1	exposed points closest to ℓ_0, ℓ_1
Q_0, Q_1	undeleted ranges farthest from ℓ_0, ℓ_1
P_i	points with x-coordinate at least x_i (*active points*)
$\mathcal{R}_i$	ranges with at least one corner to the right of $x = x_i$ (*active ranges*)
$\mathcal{R}_{i0}$	subset of $\mathcal{R}_i$ that are Type-0 (*active Type-0 ranges* at $x = x_i$)
P_f	*forbidden point set* given by $P_f = P_i \cap (P(Q_0) \cup P(Q_1))$
$\mathcal{R}_d$	*deleted range set* given by $\mathcal{R}_d = \mathcal{R}_{i0} \cap (\mathcal{R}(q_0) \cup \mathcal{R}(q_1))$

Table 1: A table of commonly used notations and their explanations.

Therefore, if our dynamic program remembers the ranges Q_0 and Q_1, we can compute the forbidden point set $P_f = P_i \cap (P(Q_0) \cup P(Q_1))$ at $x = x_i$. Since there are $O(n^2)$ choices for the pair Q_0, Q_1, the number of possible sets P_f is also $O(n^2)$. We can now identify the subproblems by the tuple $S(i,\ k',\ Q_0,\ Q_1)$ which represents the maximum number of points in $P_i \setminus P_f$ that are exposed by deleting k' ranges that begin on or after $x = x_i$. This gives us the following recurrence.

$$
\begin{aligned}
&S(i,\ k',\ Q_0,\ Q_1)\ = \\
&\max \begin{cases} S(i+1,\ k'-1,\ Q_0,\ Q_1) & \text{delete range } R_i \\ S(i+1,\ k',\ \mathit{farther}(Q_0, R_i),\ \mathit{farther}(Q_1, R_i)) & R_i \text{ not deleted} \end{cases} \\
&\qquad (\textit{event } x = x_i \textit{ was beginning of a range } R_i \in \mathcal{R}) \\
&\max \begin{cases} S(i+1,\ k',\ Q_0,\ Q_1) & \text{if } p_i \in P_f, \text{ cannot expose } p_i \\ S(i+1,\ k',\ Q_0,\ Q_1) + 1 & \text{otherwise, expose } p_i \end{cases} \\
&\qquad (\textit{otherwise, event } x = x_i \textit{ was a point } p_i \in P)
\end{aligned}
$$

Here, the function *farther* simply updates the ranges Q_0, Q_1 with R_i if needed. More precisely, if R_i is anchored to ℓ_0 and is farther from ℓ_0 than Q_0, then *farther*(Q_0, R_i) returns R_i, otherwise it returns Q_0. Similarly, if R_i is anchored to ℓ_1, and is farther from ℓ_1 than Q_1, then *farther*(Q_1, R_i) returns R_i, otherwise it returns Q_1.

The optimal solution is given by $P(0, k, Q_0^*, Q_1^*)$, where Q_0^*, Q_1^* are two artificial ranges of zero-width : Q_0^* is anchored to ℓ_0 and is defined by corners $(0, 0)$ and $(0, 1)$; similarly, Q_1^* is anchored to ℓ_1 and is defined by corners $(0, 1)$ and $(1, 1)$.

Remark 1. We note that remembering a constant number of exposed points q_0, q_1 (DP-template-0) or a constant number of undeleted ranges Q_1, Q_2 (DP-template-1) by themselves cannot solve *both* Type-0 and Type-1 ranges. For instance, in Figure 6(a) with Type-0 ranges, if R_1, R_2 were both *not deleted* but we remembered one of them, then we will incorrectly expose one of p, p'. Similarly in Figure 6(b) with Type-1 ranges, if p_1, p_2 were both exposed but we only remembered one of them, we will pay for one of the ranges R, R' again when we expose p_3. However, since both the dynamic programs for Type-0 and Type-1 ranges express subproblems at event i in terms of subproblems at event $i + 1$, we can easily combine them with minor adjustments.

4.1.3 Combining them together

In the following, we will combine the dynamic programs for Type-0 and Type-1 ranges to obtain a dynamic program for max-exposure in a unit square C. We will need a couple of changes. First, the events at $x = x_i$ are now defined by either a point $p_i \in P$ or beginning of a Type-1 range R_i. Next, the *deleted range set* $\mathcal{R}_d$ at $x = x_i$ will only consist of Type-0 ranges and is defined as $\mathcal{R}_d = \mathcal{R}_{i0} \cap (\mathcal{R}(q_0) \cup \mathcal{R}(q_1))$ where $\mathcal{R}_{i0} \subseteq \mathcal{R}_i$ is the set of Type-0 ranges that intersect the vertical line $x = x_i$. The *forbidden point set* $P_f = P_i \cap (P(Q_0) \cup P(Q_1))$ stays the same. Here q_0, q_1, Q_0, Q_1 are same as defined before. (For the sake of convenience, Table 1 lists these notations with explanation.)

The subproblems represent the maximum number of points in $P_i \setminus P_f$ that can be exposed by deleting k' ranges from $\mathcal{R}_i \setminus \mathcal{R}_d$. If $k_i = |(\mathcal{R}(p_i) \cap \mathcal{R}_{i0}) \setminus \mathcal{R}_d|$, then we obtain the following combined recurrence.

$$S(i,\ k',\ q_0,\ q_1, Q_0, Q_1)\ =$$

$$\max \begin{cases} S(i+1,\ k',\ q_0,\ q_1,\ Q_0,\ Q_1) & \text{if } p_i \in P_f\text{, cannot expose } p_i \\ S(i+1,\ k',\ q_0,\ q_1,\ Q_0,\ Q_1) & \text{choose to not expose } p_i \\ S(i+1,\ k'-k_i,\ \mathit{closer}(q_0,p_i),\ \mathit{closer}(q_1,p_i),\ Q_0,\ Q_1)+1 & \text{expose } p_i \end{cases}$$

$$(\textit{event } x = x_i \textit{ was a point } p_i \in P_i)$$

$$\max \begin{cases} S(i+1,\ k'-1,\ q_0,\ q_1,\ Q_0,\ Q_1) & \text{delete Type-1 range } R_i \\ S(i+1,\ k',\ q_0,\ q_1,\ \mathit{farther}(Q_0,R_i),\ \mathit{farther}(Q_1,R_i)) & R_i \text{ not deleted} \end{cases}$$

$$(\textit{event } x = x_i \textit{ was beginning of a Type-1 range } R_i \in \mathcal{R}_i)$$

The optimal solution is given by $S(0,\ k,\ q_0^*,\ q_1^*,\ Q_0^*,\ Q_1^*)$. The correctness of the above formulation follows from the fact that when we choose to expose p_i, we are guaranteed that all Type-1 ranges in $\mathcal{R}(p_i)$ have already been deleted, and the expression k_i only charges for Type-0 ranges containing p_i. As for the running time, for each event $x = x_i$, we compute $O(kn^2m^2)$ entries and computing each entry takes constant time. Since there are $O(n+m)$ events, we obtain the following.

Lemma 8. *Given a set P of m points in a unit square C and a set of n unit square ranges $\mathcal{R}$, we can compute their max-exposure in $O(k(n+m)n^2m^2)$ time.*

4.2 A Constant Factor Approximation

We now use the preceding algorithm to solve the max-exposure problem for general set of points and unit square ranges within a factor 4 of optimum. In particular, we compute a set of $4k$ ranges in $\mathcal{R}$ such that the number of points exposed in P by deleting them is at least the optimal number of points. Suppose we embed the ranges $\mathcal{R}$ on a uniform unit-sized grid G, and define $\mathcal{C}$ as the collection of all cells in G that contain at least one point of P. Then we can solve exactly for each cell in $\mathcal{C}$ and combine them using dynamic programming as described in Algorithm 2 (DP-Approx). See also Figure 7.

Lemma 9. *If $P^* \subseteq P$ is the optimal set of exposed points, then global$(1, 4k) \geq |P^*|$, that is , the algorithm* DP-Approx *achieves a 4-approximation and runs in $O(k(n+m)n^2m^2)$ time.*

Proof. Consider the optimal set of ranges $\mathcal{R}^* \subseteq \mathcal{R}$. Observe that each range $R \in \mathcal{R}^*$ intersects at most four grid cells. Let $R_i = R \cap C_i$ be the rectangular region defined by intersection of R and C_i.

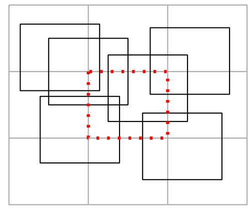

Figure 7: Embedding a max-exposure instance with unit square ranges on a unit-sized grid. Optimal solution in each grid cell can be computed exactly using Lemma 8.

Algorithm 2 DP-Approx

---

1. Apply Lemma 8 to solve max-exposure locally in every cell $C_i \in \mathcal{C}$ for all $0 \leq k_i \leq k$. Call this a *local* solution denoted by $\mathit{local}(P(C_i), \mathcal{R}(C_i), k_i)$, where $P(C_i) \subseteq P$ is the set of points contained in cell C_i and $\mathcal{R}(C_i)$ is the set of ranges intersecting C_i.

2. Process cells in $\mathcal{C}$ in any order $C_1, C_2, \ldots, C_g$, and define $\mathit{global}(i,\ k')$ as the maximum number of points exposed in the cells C_i through C_g using k' ranges. Combine local solutions to obtain $\mathit{global}(i,\ k')$ as follows.

$$\mathit{global}(i,\ k') \ = \ \max_{0 \leq k_i \leq k'} \ \mathit{global}(i+1,\ k' - k_i) \ + \ \mathit{local}(P(C_i),\ \mathcal{R}(C_i),\ k_i)$$

3. Return $\mathit{global}(1, 4k)$ as the number of exposed points.

Clearly, there are at most four regions R_i for each $R \in \mathcal{R}^*$ and therefore $4k$ in total. At this point, the regions in cell C_i are disjoint from regions in some other cell $C_j \in \mathcal{C}$. Therefore, optimal solution exposes $|P^*|$ points over a set of cells $\mathcal{C}^*$ such that the set $\mathcal{R}^*$ has at most $4k$ disjoint components in the cells $\mathcal{C}^*$. Since we can solve the problem exactly for each cell and can combine them using the above dynamic program, we have that $\mathit{global}(1, 4k) \geq |P^*|$ and we achieve a 4-approximation.

For the running time, we observe that solving max-exposure locally in a cell C_i takes $O(k(n_i + m_i)n_i^2 m_i^2)$ time, where n_i is the number of ranges that intersect C_i and m_i is the number of points in P that lie in C_i. Summed over all cells, we get the following bound.

$$\begin{aligned}\sum_i k(n_i + m_i)n_i^2 m_i^2 \ &\leq \ k \ \sum_i (n_i + m_i) \ \sum_i n_i^2 \ \sum_i m_i^2 \\ &\leq k(n+m) \ (\sum_i n_i)^2 \ (\sum_i m_i)^2 \ = \ O(k(n+m)n^2 m^2)\end{aligned}$$

Once the local solutions are computed, the dynamic program that merges them into a global solution has $O(k|\mathcal{C}|)$ subproblems and computing each subproblem takes $O(k)$ time. Recall that every cell in $\mathcal{C}$ contains at least one point, so $|\mathcal{C}| \leq n$ and the merge step takes an additional $O(k^2 n)$ time. $\square$

4.3 Towards a PTAS

In this section, we will show how to extend the exact algorithm for the restricted max-exposure instance where all points lie inside a unit square (Lemma 8) to obtain an exact solution for the max-exposure instance where all points are contained in a $h \times h$ square $\mathcal{C}$. Without loss of generality, we can assume that the lower left corner of C is at the origin $(0, 0)$ and $\mathcal{C}$ is subdivided into h^2 unit-sized grid cells.

Observe that a major hurdle in generalizing the dynamic program from Section 4.1 for max-exposure in a unit square cell to the grid $\mathcal{C}$ is that a range R can be *double counted* in multiple cells. Specifically, range R may contain exposed points in at most four cells of $\mathcal{C}$ and can be counted in each one of them. (See also Figure 8.) Indeed a natural generalization of the earlier dynamic program to h anchor lines avoids double counting of ranges in the same column of $\mathcal{C}$ (vertical neighbors). However, some additional work is required to avoid double counting in adjacent columns (horizontal and diagonal neighbors).

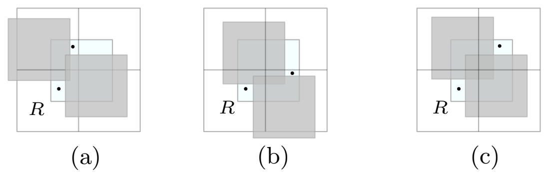

Figure 8: Examples where a deleted range R can potentially be counted in two cells that are: (a) vertical neighbors (b) horizontal neighbors (c) diagonal neighbors

To handle this, we first apply the following transformation which we call *flattening* of the grid $\mathcal{C}$.

Flattening the grid $\mathcal{C}$ Intuitively, the flattening process transforms a $h \times h$ grid into a $h^2 \times 1$ vertical slab by *shifting* the i-th column and aligning it on top of the $(i-1)$-th column. More precisely, we label the cells *column by column* from *left to right* and bottom to top in each column. That is, cells of the column 1 are labeled as $1, 2, \ldots, h$ and the cells of column 2 are labeled $h+1, \ldots, 2h$ and so on. Then, flattening refers to simply stacking all the cells in their numbered order. In other words, we shift the coordinates of all points and parts of ranges in column i of the grid $\mathcal{C}$ by $(-(i-1), (i-1)h)$, for all $2 \leq i \leq h$. (See also Figure 9.)

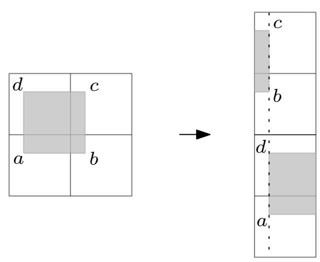

Figure 9: Flattening a 2×2 grid containing one unit square range that is split into Type-0 and Type-1 components.

After this transformation, all x-coordinates are within the range $[0, 1]$ and y-coordinates are within the range $[0, h^2]$. Moreover, every range R is split into two possibly disconnected half-ranges which preserve the following important property that follows readily from the fact that the ranges are unit squares.

Lemma 10. *Let R be a range and $\mathcal{C}_i, \mathcal{C}_{i+1}$ be the two consecutive columns of the grid $\mathcal{C}$ intersected by range R. Then, R is Type-1 with respect to cells in $\mathcal{C}_i$ and Type-0 with respect to cells in $\mathcal{C}_{i+1}$, and after the* flattening *transformation, the x-coordinate at which R* begins *as a Type-1 range in $\mathcal{C}_i$ is the same as the x-coordinate at which R* finishes *as Type-0 range in $\mathcal{C}_{i+1}$.*

Proof. The range R intersects the vertical line $x = i$ which is coincident with the right (resp. left) boundary of cells in $\mathcal{C}_i$ (resp. $\mathcal{C}_{i+1}$). Therefore, R is Type-1 in cells of $\mathcal{C}_i$ and Type-0 in cells of $\mathcal{C}_{i+1}$.

Let the x-coordinate of left boundary of R (that lies in i-th column) be $(i-1)+\delta$. Therefore, the x-coordinate of right boundary of R would be $(i-1)+\delta+1 = i+\delta$, and it will lie in $(i+1)$-th column. After the transformation both these coordinate values would be δ. $\square$

From the above lemma, it follows that every range $R \in \mathcal{R}$ has a Type-0 component and a Type-1 component which may lie in non-consecutive cells. In the rest of the discussion, we will refer to these components by their type as prefix. For example, Type-0 range R refers to the Type-0 component of R.

Once we have flattened the grid $\mathcal{C}$, our algorithm is an almost straightforward extension of the dynamic program from Section 4.1 to h^2+1 anchor lines $\ell_0, \ell_1, \ell_{h^2}$. Same as before, we process the two types of events : $x = x_i$ *is a point* p_i and $x = x_i$ *is beginning of Type-1 range* R_i. However at every $x = x_i$, we will now need to remember the set $\mathbf{q} = \{q_0^+, q_0^-, \ldots, q_{h^2}^+, q_{h^2}^-\}$ of $O(h^2)$ points consisting of *closest exposed points* q_j^+, q_j^- respectively above and below every anchor line ℓ_j. Similarly, we will need to remember the set $\mathbf{Q} = \{Q_0^+, Q_0^-, \ldots, Q_{h^2}^+, Q_{h^2}^-\}$ of $O(h^2)$ ranges consisting of *farthest undeleted Type-1 ranges* Q_j^+, Q_j^- respectively above and below every anchor line ℓ_j.

Then at $x = x_i$, we extend the definitions from Table 1 to obtain the *forbidden point set* $P_f = P_i \cap \bigcup_{Q\in\mathbf{Q}} P(Q)$ and the *deleted range set* $\mathcal{R}_d = \mathcal{R}_{i0} \cap \bigcup_{q\in\mathbf{q}} \mathcal{R}(q)$. Recall that P_i is the set of points with x-coordinate at least x_i and $\mathcal{R}_{i0}$ is the set of Type-0 ranges that are active at $x = x_i$. Also recall that $P(Q)$ denotes the set of points contained in range Q and $\mathcal{R}(q)$ denotes the set of ranges containing point q. This gives us the following dynamic program which we will refer to as DP-Flattened. Same as before, we have $k_i = |(\mathcal{R}(p_i) \cap \mathcal{R}_{i0}) \setminus \mathcal{R}_d|$.

$$
\begin{aligned}
&S(i,\ k',\ \mathbf{q},\ \mathbf{Q}) \ = \\
&\max\begin{cases} S(i+1,\ k',\ \mathbf{q},\ \mathbf{Q}) & \text{if } p_i \in P_f, \text{ cannot expose } p_i \\ S(i+1,\ k',\ \mathbf{q},\ \mathbf{Q}) & \text{choose to not expose } p_i \\ S(i+1,\ k'-k_i,\ \mathit{closer}(\mathbf{q},\ p_i),\ \mathbf{Q}) + 1 & \text{expose } p_i \end{cases} \\
&\qquad (\textit{event } x = x_i \textit{ was a point } p_i \in P_i) \\
&\max\begin{cases} S(i+1,\ k',\ \mathbf{q},\ \mathbf{Q}) & \text{if } R_i \in \mathcal{R}_d, \text{ already deleted} \\ S(i+1,\ k'-1, \mathbf{q},\ \mathbf{Q}) & \text{delete Type-1 range } R_i \\ S(i+1,\ k',\ \mathbf{q},\ \mathit{farther}(\mathbf{Q},\ R_i)) & R_i \text{ not deleted} \end{cases} \\
&\qquad (\textit{event } x = x_i \textit{ was beginning of a Type-1 range } R_i \in \mathcal{R}_i)
\end{aligned}
$$

Here, *closer*$(\mathbf{q}, p_i)$ denotes the operation of updating the appropriate closest exposed point in $\mathbf{q}$ with point p_i. More precisely, let C_j be the cell bounded by anchor lines ℓ_{j-1} and ℓ_j that contains the exposed point p_i. We update $\mathbf{q}$ such that $q_{j-1}^+ = \mathit{closer}(q_{j-1}^+, p_i)$ and $q_j^- = \mathit{closer}(q_j^-, p_i)$. Similarly, let ℓ_j be the anchor line intersecting R_i, then *farther*$(\mathbf{Q}, R_i)$ denotes the operation of updating $\mathbf{Q}$ with the farthest undeleted range on both sides of ℓ_j as $Q_j^+ = \mathit{farther}(Q_j^+, R_i)$ and $Q_j^- = \mathit{farther}(Q_j^-, R_i)$. The optimal solution is given by $S(0,\ k,\ \mathbf{q}^*,\ \mathbf{Q}^*)$, where $\mathbf{q}^*$, $\mathbf{Q}^*$ consist of the initial values for each anchor line.

At any event $x = x_i$, the above dynamic program accounts for the cost of deleting a range R in one of two ways: either as a Type-0 range included in the term k_i or as Type-1 range by paying unit cost. In the next lemma, we show that every deleted range is counted exactly once and use it to establish the correctness.

Lemma 11. *The dynamic program* DP-flattened *computes an optimal solution for max-exposure instance* $(\mathcal{R}, P, k)$ *in an* $h \times h$ *grid and runs in* $O(k(nm)^{O(h^2)})$ *time.*

Proof. The running time bound follows from the number of exposed points and undeleted ranges we need to remember.

To prove correctness, consider an optimal set of deleted ranges $\mathcal{R}^*$ and its exposed points P^*. Let the *value* of the solution returned by the dynamic program be the number of points it choses to expose and the *cost* of the solution is the total cost of ranges it deletes. First, we claim that there exists a sequences of choices at events $x = x_i$ where the dynamic program selects points and ranges consistent with the optimal solution, that is, chooses to only expose points in P^* and to only delete Type-1 ranges in $\mathcal{R}^*$. This is easy to verify because because $P^* \cap P_f = \emptyset$, so the dynamic program can choose to expose point $p_i \in P^*$ when $x = x_i$ is a point-event. Indeed the value of the solution is $|P^*|$. Next we will show that every range in $\mathcal{R}^*$ is counted exactly once, and therefore the cost of the solution is also k.

We claim that at every *point-event* $x = x_i$ where we expose the point $p_i \in P^*$, all ranges in $\mathcal{R}(p_i)$ are deleted and counted exactly once. To see this, let $R \in \mathcal{R}(p_i)$ be a range containing p_i and let $x = x_r$ be the x-coordinate at which R finishes as a Type-0 range and starts as a Type-1 range. We have two disjoint cases.

1. *p_i is contained in Type-0 component of R.* Let ℓ_j be the line to which Type-0 range R is anchored. We have two subcases.

   (a) *R does not contains any exposed point to the left of $x = x_i$.* In this case, we charge for R and remember that R has already been counted using the closest exposed points q_j^+, q_j^- above and below ℓ_j. Therefore, we will have $R \in \mathcal{R}_d$ at least until $x = x_r$, (when it switches from being Type-0 to Type-1). Since R cannot be charged at $x > x_r$, it is charged exactly once in total.

   (b) *R contains an exposed point to the left of $x = x_i$.* Then we will have $R \in \mathcal{R}_d$, and as discussed above R was already counted and would not be charged again.

2. *p_i is contained in Type-1 component of R.* Since p_i is exposed, it is not contained in the forbidden point set P_f. Therefore, R must be deleted when it began as a Type-1 event at $x = x_r$ or else we would have $p_i \in P_f$. As discussed above, if R was deleted as a Type-0 range to the left of $x = x_r$, we must have $R \in \mathcal{R}_d$ at $x = x_r$, so it would not be charged again. If R was not deleted as a Type-0 range, then it would be charged at $x = x_r$ as a Type-1 range and is never charged again.

Therefore, the solution returned by dynamic program $S(0, k, \mathbf{q}^*, \mathbf{Q}^*)$ has value at least optimal. ☐

4.4 A $(1 + \epsilon)$-Approximation Algorithm

We will now apply grid shifting technique by Hochbaum and Maas [20] to obtain an $(1 + \epsilon)$-approximation[1]. In particular, if P^* is the optimal set of exposed points, then we show how to compute a set of $(1 + \epsilon)k$ ranges deleting which will expose at least $|P^*|$ points. Using similar ideas but with small adjustments, we also show how to expose at least $(1 - \epsilon)|P^*|$ points by deleting exactly k ranges.

Theorem 3. *There exists an algorithm for max-exposure with unit-square ranges running in $k(mn)^{O(1/\epsilon^2)}$ time that exposes at least optimal number of points by deleting $(1 + \epsilon)k$ ranges.*

[1] The PTAS presented here simplifies and corrects an error in the PTAS that appeared in the conference version [21] of the paper.

Proof. For a given shift value $a, b \in \{0, \ldots, h-1\}$, we compute the optimal solution inside every $h \times h$ cell $\mathcal{C}_{ij} = [a + ih,\ a + (i+1)h] \times [b + jh,\ b + (j+1)h]$ for all $i, j \in \mathbb{Z}$. Using the exact solution in each cell as *local solution*, we use the algorithm DP-Approx (from Section 4.2) to combine them into a global solution for the entire grid given by $S_{ab} = \mathit{global}(1, k(1+\epsilon))$, with $\epsilon = \lceil 8/h \rceil$. We repeat this for every shift a, b, and return S_{ab} that achieves the maximum value.

To see why this exposes at least optimal number of points, consider an optimal set of deleted ranges $\mathcal{R}^*$ and sets $\mathcal{R}^*_a, \mathcal{R}^*_b \subseteq \mathcal{R}^*$ intersected by *boundary* grid lines $x = a + ih$ and $y = b + jh$ respectively, for all i, j. These grid lines split the intersected ranges into at most $Z_{ab} = 2|\mathcal{R}^*_a| + 2|\mathcal{R}^*_b|$ disjoint components.

$$\sum_{0 \le a, b < h} Z_{ab} \ = \ 2h \sum_{0 \le a < h} |\mathcal{R}^*_a| + 2h \sum_{0 \le b < h} |\mathcal{R}^*_b| \ \le \ 8hk$$
$$\implies \min_{0 \le a, b < h} Z_{ab} \ \le \ 8k/h$$

The first inequality holds because every range can touch at most two grid lines, so we have $\sum_{0 \le a < h} |\mathcal{R}^*_a| \le 2k$ and $\sum_{0 \le b < h} |\mathcal{R}^*_b| \le 2k$. Hence there exists a shift value a, b for which the ranges in optimal solution have at most $k(1+\epsilon)$ disjoint components in cells $\mathcal{C}_{ij}$. Therefore, $\mathit{global}(1, k(1+\epsilon))$ returns at least an optimal number of points. ☐

Theorem 4. *There exists an algorithm for max-exposure with unit-square ranges running in $k(mn)^{O(1/\epsilon^2)}$ time that exposes at least $(1-\epsilon)$ fraction of optimal number of points by deleting k ranges.*

Proof. For a given shift value $a, b \in \{0, \ldots, h-1\}$, we first *preprocess* the input by discarding points so that the set of ranges intersecting the boundary grid lines $x = a + ih$ and $y = b + jh$ do not contain any point. Specifically, for every shift value a, b, discard the points that are within a unit distance from grid boundary lines $x = a + ih$ or $y = b + jh$ for all i, j. On the modified input, we run the exact solution in each cell as *local solution*, and then use DP-Approx (from Section 4.2) to combine them into a global solution for the entire grid given by $S_{ab} = \mathit{global}(1, k)$. We repeat this for every shift a, b, and return S_{ab} that achieves the maximum value.

Let P^* be the optimal set of exposed points. It remains to show that the above algorithm exposes at least $(1-\epsilon)|P^*|$ points. To see this, for the shift value a, b, consider the set of discarded points $P^*_a, P^*_b \subseteq P^*$ that are within a unit distance from $x = a + ih$ and $y = b + jh$ respectively. These $|P^*_a| + |P^*_b|$ will not be exposed by our algorithm

$$\sum_{0 \le a, b < h} |P^*_a| + |P^*_b| \ = \ h \sum_{0 \le a < h} |P^*_a| + h \sum_{0 \le b < h} |P^*_b| \ \le \ 4h|P^*|$$
$$\implies \min_{0 \le a, b < h} (|P^*_a| + |P^*_b|) \ \le \ 4|P^*|/h$$

The first inequality holds because every point can lie within a unit distance of at most two horizontal (resp. vertical) lines, so we have $\sum_{0 \le a < h} |P^*_a| \le 2k$ and $\sum_{0 \le b < h} |P^*_b| \le 2|P^*|$. Therefore, there exists some a, b for which the number of remaining points in the input is at most $(1 - 4/h)|P^*|$. Since every $h \times h$ cell is mutually independent, $\mathit{global}(1, k)$ returns at least $(1-\epsilon)|P^*|$ exposed points, where $\epsilon = \lceil 4/h \rceil$. ☐

5 Extensions and Applications

In this section, we discuss some extensions and applications of our the results from previous section. We say that the range family $\mathcal{R}$ consists of *fat rectangles* if every range $R \in \mathcal{R}$ is a rectangle of bounded *aspect ratio*. Moreover, we say that $\mathcal{R}$ consists of *similar and fat rectangles*, if ranges in $\mathcal{R}$ are rectangles and the ratio of the largest to the smallest side in $\mathcal{R}$ is constant. We show that if $\mathcal{R}$ consists of *similar and fat* rectangles, one can achieve a constant approximation. Moreover, if $\mathcal{R}$ consists of *fat rectangles* one can achieve a bicriteria $O(\sqrt{k})$-approximation.

5.1 Approximation for Similar and Fat Rectangles

Let a, b be the length of smallest and largest sides of rectangles in $\mathcal{R}$ such that $b/a = c$ is constant. Then we can modify the input instance as follows. Replace each range $R \in \mathcal{R}$ by covering it with at most c^2 squares of sidelength a such that the areas occupied by R and its replacements are the same. Now, we have a modified set of ranges $\mathcal{R}'$ consisting of squares that have the same sidelength. Consider the optimal solution with k ranges $\mathcal{R}^*$ that exposes m^* points. It is easy to see that the set $\mathcal{R}^*$ corresponds to at most c^2k ranges in the modified instance, and therefore deleting c^2k ranges from $\mathcal{R}'$ exposes at least m^* points. Therefore, we can run the polynomial-time 4-approximation algorithm (Lemma 9) to obtain a set of at most $4c^2k$ ranges that expose at least m^* points.

Theorem 5. *Given a set of points P, a set of rectangle ranges $\mathcal{R}$ such that the ratio of the largest to the smallest side in $\mathcal{R}$ is bounded by a constant, then there exists a polynomial time $O(1)$-approximation algorithm for max-exposure.*

5.2 Approximation for Fat Rectangles

We now consider the case when rectangles in $\mathcal{R}$ have bounded aspect ratio. That is for all rectangles $R \in \mathcal{R}$, the ratio of its two sides is bounded by a constant c. We transform the input ranges $\mathcal{R}$ to obtain a modified set of ranges $\mathcal{R}'$ as follows. For each rectangle $R \in \mathcal{R}$, let x be the length of the smaller side of R. Then we replace R by at most $\lceil c \rceil$ *squares* each of sidelength x. If m^* is the optimal number of points exposed by deleting k ranges from $\mathcal{R}$, then there exists a set of $O(k)$ ranges in $\mathcal{R}'$ deleting which will expose at least m^* points. Observe that the set $\mathcal{R}'$ consists of *square* ranges, of possibly different sizes. Therefore, if we can obtain an f-approximation for square ranges, we can easily obtain $O(f)$-approximation with fat rectangles.

5.2.1 A Bicriteria $O(\sqrt{k})$-approximation for Squares

We will describe an approximation algorithm for the case when the set of ranges $\mathcal{R}$ consists of axis-aligned squares. We achieve an approximation algorithm in three steps. First, we partition the point set by assigning the points to one of the input squares. Next, we solve the problem exactly for a fixed input square. Finally, we combine these solutions to achieve a good approximation to the optimal solution.

We define $\mathcal{A} : P \to \mathcal{R}$ to be a function that assigns a point in P to exactly one range in $\mathcal{R}$. If $\mathcal{R}(p_i)$ is the set of squares that contain p_i, then $\mathcal{A}(p_i)$ is the smallest square in $\mathcal{R}(p_i)$. This assignment scheme ensures the following property.

Lemma 12. *Let $R \in \mathcal{R}$ be a square and let $\mathcal{P}_R = \mathcal{A}^{-1}(R)$ be the set of points* assigned *to it. Moreover, let $\mathcal{R}' \subseteq \mathcal{R}$ be the set of squares that intersect R and contain at least one point in $\mathcal{P}_R$. Then, every square $R' \in \mathcal{R}'$ must have sidelength bigger than that of R, and therefore contains at least one corner of R.*

Now suppose we fix a square R, and consider a restricted max-exposure instance with the set of its assigned points $\mathcal{P}_R$. Since, ranges that contain a point in $\mathcal{P}_R$ are all bigger then R, this case is essentially the same as points inside a unit square, and therefore Lemma 8 can be easily extended to solve it exactly. This gives us the following algorithm. Here $1 \le \alpha \le k$ is a parameter.

Algorithm 3 Greedy-Squares

1. For every square $R \in \mathcal{R}$, apply Lemma 8 over the point set $\mathcal{P}_R$ to expose the maximum set of points $P(R,k) \subseteq \mathcal{P}_R$ by deleting k ranges.

2. Order squares in $\mathcal{R}$ by decreasing $|P(R,k)|$ values, and pick the set $\mathcal{S} \subseteq \mathcal{R}$ of first α squares.

3. Return $\bigcup_{R \in \mathcal{S}} P(R,k)$ as the set of exposed points.

Lemma 13. *Let m^* be the optimal number of points exposed using k squares, then algorithm* Greedy-Squares *computes a set of at most αk squares that expose at least $\alpha m^*/k$ points.*

Proof. It is easy to see that the number of squares is at most αk. To show the bound on number of points exposed, consider the optimal set $\mathcal{R}^*$ of k ranges and let the optimal set of points exposed by $\mathcal{R}^*$ to be P^*. We will now use the same assignment procedure $\mathcal{A}^* : P^* \to \mathcal{R}^*$ to assign points in P^* to a square in $\mathcal{R}^*$. That is, $\mathcal{A}^*(p_i)$ is the smallest square in $\mathcal{R}^*$ that contains p_i. We claim that $\mathcal{A}^*(p_i) = \mathcal{A}(p_i)$ for all $p_i \in P^*$ since every square that contains p_i lies in $\mathcal{R}^*$. Moreover, let $\mathcal{P}^*_R$ denote the set of points of P^* assigned to R.

Let m' be the number of points exposed by the algorithm and assume that the squares in $\mathcal{R}$ are ordered such that $|P(R_i,k)| \geq |P(R_j,k)|$ for all $i < j$. Then, we have the following.

$$\begin{aligned} m^* &= \left| \bigcup_{R \in \mathcal{R}^*} \mathcal{P}^*_R \right| = \sum_{R \in \mathcal{R}^*} |\mathcal{P}^*_R| \\ &\leq \sum_{1 \leq i \leq k} |P(R_i,k)| \leq \frac{k}{\alpha} \sum_{1 \leq i \leq \alpha} |P(R_i,k)| = \frac{k}{\alpha} m' \end{aligned}$$

□

For $\alpha = \sqrt{k}$, the above algorithm achieves a bicriteria $O(\sqrt{k})$-approximation. Since an f-approximation for square ranges gives an $O(f)$-approximation for fat rectangles, we obtain the following.

Theorem 6. *Given a set of points P and a set of ranges $\mathcal{R}$ consisting of rectangles of bounded aspect ratio, then one can obtain a bicriteria $O(\sqrt{k})$-approximation for max-exposure in polynomial time.*

6 Conclusion

In this paper, we introduced the max-exposure problem, proved its hardness, and explored approximation schemes for it. We showed that the problem is hard to approximate even when the range space $\mathcal{R}$ consists of two types of rectangles. When the ranges are defined by translates of a single rectangle, we presented a polynomial-time approximation scheme (PTAS). Some natural questions to explore in the future include better approximation algorithms, and simpler range spaces such as those defined by axis-aligned squares. For instance, can one achieve a constant factor approximation for axis-aligned squares?

References

[1] E. Chlamtáč, M. Dinitz, Y. Makarychev, Minimizing the union: Tight approximations for small set bipartite vertex expansion, in: Proceedings of the 28th SODA, 2017, pp. 881–899. 2, 4

[2] U. Feige, A threshold of ln n for approximating set cover, Journal of the ACM (JACM) 45 (4) (1998) 634–652. 2

[3] R. J. Fowler, M. S. Paterson, S. L. Tanimoto, Optimal packing and covering in the plane are np-complete, Information processing letters 12 (3) (1981) 133–137. 2

[4] E. Chlamtác, M. Dinitz, C. Konrad, G. Kortsarz, G. Rabanca, The densest k-subhypergraph problem, SIAM Journal on Discrete Mathematics 32 (2) (2018) 1458–1477. 3

[5] U. Feige, D. Peleg, G. Kortsarz, The dense k-subgraph problem, Algorithmica 29 (3) (2001) 410–421. 3

[6] Y. Asahiro, K. Iwama, H. Tamaki, T. Tokuyama, Greedily finding a dense subgraph, Journal of Algorithms 34 (2) (2000) 203–221. 3

[7] S. Arora, D. Karger, M. Karpinski, Polynomial time approximation schemes for dense instances of np-hard problems, Journal of computer and system sciences 58 (1) (1999) 193–210. 3

[8] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, A. Vijayaraghavan, Detecting high log-densities: an $O(n^{1/4})$ approximation for densest k-subgraph, in: Proceedings of the 42nd STOC, ACM, 2010, pp. 201–210. 3

[9] P. K. Agarwal, J. Pan, Near-linear algorithms for geometric hitting sets and set covers, in: Proceedings of 30th SoCG, ACM, 2014, p. 271. 3

[10] H. Brönnimann, M. T. Goodrich, Almost optimal set covers in finite vc-dimension, Discrete & Computational Geometry 14 (4) (1995) 463–479. 3

[11] N. H. Mustafa, R. Raman, S. Ray, Settling the apx-hardness status for geometric set cover, in: Proceedings of 55th FOCS, IEEE, 2014, pp. 541–550. 3

[12] C. Chekuri, K. L. Clarkson, S. Har-Peled, On the set multi-cover problem in geometric settings, ACM Transactions on Algorithms (TALG) 9 (1) (2012) 9. 3

[13] M. Cygan, F. Grandoni, S. Leonardi, M. Mucha, M. Pilipczuk, P. Sankowski, Approximation algorithms for union and intersection covering problems, in: Proceedings of 31st FSTTCS, 2011, p. 28. 3

[14] S. Bandyapadhyay, N. Kumar, S. Suri, K. Varadarajan, Improved approximation bounds for the minimum constraint removal problem, in: Proceedings of 21st APPROX, 2018, pp. 2:1–2:19. 3

[15] E. Eiben, J. Gemmell, I. Kanj, A. Youngdahl, Improved results for minimum constraint removal, in: Proceedings of 32nd AAAI Conference on Artificial Intelligence, 2018. 3

[16] S. Bereg, D. G. Kirkpatrick, Approximating barrier resilience in wireless sensor networks, in: Proceedings of 5th ALGOSENSORS, 2009, pp. 29–40. 3

[17] M. Korman, M. Löffler, R. I. Silveira, D. Strash, On the complexity of barrier resilience for fat regions and bounded ply, Comput. Geom. 72 (2018) 34–51. 3

[18] U. Feige, M. Seltser, On the densest k-subgraph problems, Tech. rep., Weizmann Institute of Science, Jerusalem, Israel (1997). 3

[19] K. L. Clarkson, P. W. Shor, Application of random sampling in computational geometry, II, Discrete & Computational Geometry 4 (1989) 387–421. 7

[20] D. S. Hochbaum, W. Maass, Approximation schemes for covering and packing problems in image processing and vlsi, Journal of the ACM (JACM) 32 (1) (1985) 130–136. 16

[21] N. Kumar, S. Sintos, S. Suri, The Maximum Exposure Problem, in: (APPROX/RANDOM 2019), 2019, pp. 19:1–19:20. 16