Maximum Independent Set of Rectangles

Parinya Chalermsook [*] Julia Chuzhoy [†]

October 25, 2008

Abstract

We study the Maximum Independent Set of Rectangles (MISR) problem: given a collection $\mathcal{R}$ of n axis-parallel rectangles, find a maximum-cardinality subset of disjoint rectangles. MISR is a special case of the classical Maximum Independent Set problem, where the input is restricted to intersection graphs of axis-parallel rectangles. Due to its many applications, ranging from map labeling to data mining, MISR has received a significant amount of attention from various research communities. Since the problem is NP-hard, the main focus has been on the design of approximation algorithms. Several groups of researches have independently suggested $O(\log n)$-approximation algorithms for MISR, and this remained the best currently known approximation factor for the problem. The main result of our paper is an $O(\log \log n)$-approximation algorithm for MISR. Our algorithm combines existing approaches for solving special cases of the problem, in which the input set of rectangles is restricted to containing specific intersection types, with new insights into the combinatorial structure of sets of intersecting rectangles in the plane.

We also consider a generalization of MISR to higher dimensions, where rectangles are replaced by d-dimensional hyper-rectangles. Our results for MISR imply an $O((\log n)^{d-2} \log \log n)$-approximation algorithm for this problem, improving upon the best previously known $O((\log n)^{d-1})$-approximation.

1 Introduction

In the Maximum Independent Set of Rectangles problem (MISR), the input is a set $\mathcal{R}$ of n rectangles whose sides are parallel to the axes, and the goal is to find a maximum cardinality subset of non-intersecting rectangles. It is easy to see that MISR is a special case of the classical Maximum Independent Set problem, in which the input is a graph G and the goal is to find a maximum cardinality subset S of vertices that does not contain any edge. MISR is equivalent to Maximum Independent Set where the input is restricted to intersection graphs of axis-parallel rectangles, assuming the rectangle representation is given.

Maximum Independent Set is probably one of the most fundamental and extensively studied combinatorial optimization problems. The current best approximation algorithm achieves an $O(n/(\log n)^2)$ factor [3], and the problem is known to be $n^{1-\epsilon}$-hard to approximate for any $\epsilon > 0$ unless NP = ZPP [17], thus most probably ruling out the possibility of provably good approximation algorithms. However, for many applications, it is enough to solve the problem on restricted classes of instances, which often turn out to be more tractable. For example, Maximum Independent Set is known to be efficiently solvable on interval graphs (intersection graphs of intervals of the real line). MISR can be seen as the generalization of this problem to two dimensions. More generally, for $d \geq 1$, a d-box graph is an intersection graph of d-dimensional axis-parallel hyper-rectangles (boxes). In this paper we study the Maximum Independent Set problem on d-box graphs (MISB_d), our main focus being on the two-dimensional case ($d = 2$), which is equivalent to MISR. We assume that the box representation of the graph is given as problem input.

MISR is a basic problem in computational geometry and has many applications, e.g. in data mining [20, 14, 22], map labeling [1, 11], and channel admission controls with advance reservation [23]. We sketch some of the applications below. It is not surprising then that MISR has attracted a considerable amount of interest from various research communities. Since the problem is known to be NP-hard [13, 19], the main focus has been on designing approximation algorithms. Several groups of researches have independently suggested $O(\log n)$-approximation algorithms for MISR [1, 20, 25]. Further, Berman et al. [2] have shown that the constant term in the $O(\log n)$ factor can be made arbitrarily small, and for every k, there is a $\lceil \log_k n \rceil$-approximation algorithm that runs in $O(n^k \mathsf{OPT})$ time, where OPT is the cost of the optimal solution. Chan [5] improved the running time

[*]Department of Computer Science, University of Chicago, Chicago, IL 60637. Email: parinya@uchicago.edu

[†]Toyota Technological Institute, Chicago, IL 60637. Email: cjulia@tti-c.org.

of the algorithm to min $\left\{O(n \log n + nq^{k-2}), n^{O(k/\log k)}\right\}$, where q is the maximum clique size in the corresponding intersection graph. Lewin-Eytan, Naor and Orda [23] designed a factor $4q$-approximation algorithm for MISR. These are the best currently known approximation algorithms for MISR, and to the best of our knowledge no hardness of approximation results are known. We notice however that Lingas and Wahlen [24] designed an exact $2^{O(\sqrt{n}\log n)}$-time algorithm for MISR. Some special cases of MISR are known to have polynomial time approximation schemes (PTAS): Hochbaum and Maass [18] and Agarwal et al. [1] showed a PTAS for the case where all rectangles have unit height and Chan [5] improved the running time of the algorithm. Erlebach et al. [12] provided a PTAS for squares and rectangles with bounded aspect ratio, and Chan [4] independently designed a PTAS for squares. For the more general MISB_d problem, Berman et al. [2] gave a polynomial time $O((\log n)^{d-1})$-approximation algorithm, and an algorithm achieving $\lfloor 1 + \frac{1}{c}\log n \rfloor^{d-1}$-approximation in time $n^{O(2^{c(d-1)})}$ for all $c \geq 1$, $d > 2$. On the negative side, Chlebík and Chlebíková [9] proved that MISB_d is APX-hard for every $d \geq 3$.

In graph theory, the *boxicity* of graph G, denoted by $\mathsf{box}(G)$ (term introduced by Roberts [26]), is the minimum dimension d, such that G can be represented as an intersection graph of d-dimensional boxes. Roberts showed that the graph obtained by removing a perfect matching from a complete $2n$-vertex graph has boxicity exactly n. A graph has boxicity 1 iff it is an interval graph. Outer-planar graphs have boxicity at most 2 [28] and planar graphs have boxicity at most 3 [31]. Some optimization problems become easier on graphs with bounded boxicity, if the box representation is known. For instance, Maximum Clique can be solved in polynomial time on graphs with constant boxicity given their box representation[1] [27]. However, finding such a representation, or even determining whether graph boxicity is K, is NP-hard even for $K = 2$ [10, 32, 21]. Chandran, Francis and Sivadasan [6] show an efficient algorithm that finds a box representation of any graph G with dimension $\lceil(\Delta + 2)\ln n\rceil$, where Δ is the maximum degree of G. They also show that for all graphs G, $\mathsf{box}(G) \leq 2\Delta^2$ [7] and $\mathsf{box}(G) \leq \mathsf{tw}(G) + 2$ [8], where $\mathsf{tw}(G)$ is the treewidth of G.

Related Work. The Maximum Independent Set problem for intersection graphs of various geometric objects has also been studied extensively. Hochbaum and Maass [18] designed a PTAS for unit d-cubes in $\mathbb{R}^d$. Building on their techniques, Erlebach, Jansen and Seidel [12] obtained a PTAS for disks and disk-like geometric objects. Chan [4] designed a PTAS for arbitrary "fat" objects[2].

In the related rectangle piercing problem, the input is a set of axis-parallel rectangles in the plane, and the goal is to choose the smallest number of points such that each input rectangle contains at least one chosen point. This problem is the dual of MISR, and therefore the optimal solution cost of the piercing problem upper-bounds that of MISR. For interval graphs, the optimal solution costs of the two problems are the same. Nielsen [25] showed an $O(\log n)$-approximation algorithm for the rectangle piercing problem. The results of Hochbaum and Maass [18] yield a PTAS for the piercing problem for unit squares. Chan [4] designed a PTAS for the piercing problem for arbitrary fat objects.

Applications. An important problem in geographic information systems is that of labeling of map features, or the Name Placement problem [11]. The goal is to place labels on a map in a way that provides an unambiguous association between the features and the labels, and ensures that labels do not overlap. Agarwal et al. [1] model this problem as follows. We are given a set S of n points in the plane, and each point $p_i \in S$ is associated with a label, represented by a rectangle R_i of a fixed size. Each such rectangle R_i has a set $\mathcal{P}_i$ of possible placements on the map, where for each placement $P \in \mathcal{P}_i$, the boundary of the corresponding rectangle contains point p_i. The goal is to maximize the number of labels placed on the map, with the restriction that the corresponding rectangles do not overlap. It is easy to see that this is a special case of the MISR problem.

MISR is also a natural abstraction of network resource allocation with advance reservation for line topologies. In some network resource allocation scenarios, the resources may be requested in advance of when they are needed [16, 29, 30, 15]. This is useful for both the users, who can then be sure that the resources they request will be available, and for the network, since it enables for better planning. Each advance request specifies the source and the destination network vertices, as well as the time interval during which the connection will be needed. The goal is to maximize the number of requests accommodated. For the special case of line topologies the problem can naturally be modeled in the MISR framework.

[1] Notice that this is not equivalent to solving Maximum Independent Set on low boxicity graphs, since the boxicity of a graph and its complement may vary.

[2] A collection $\mathcal{C}$ of objects in $\mathbb{R}^d$ is called fat iff for every $r > 0$ and every box R of size r, we can choose a constant number of points inside R, such that every object intersecting R whose size is at least r contains at least one chosen point.

Our Results and Techniques. Our main result is summarized in the following theorem.

Theorem 1.1. *There is a randomized polynomial-time approximation algorithm for the* MISR *problem that produces an* $O(\log\log n)$*-approximate solution with high probability.*

The $O(\log n)$-approximation algorithm for MISR of Agarwal et al. [1] can be viewed as a generalization of the exact algorithm for maximum independent set on interval graphs to two dimensions. We show that this approach can be further extended to higher dimensions, and in particular, for any $d \geq 1$, an $f(n)$-approximation algorithm for MISB_{d-1} implies an $O(f(n)\log n)$-approximation for MISB_d. Therefore, the following theorem is a direct consequence of Theorem 1.1:

Theorem 1.2. *For every* $d \geq 2$*, there is a randomized polynomial-time* $O((\log n)^{d-2}\log\log n)$*-approximation algorithm for* MISB_d*.*

We now sketch our main techniques. We distinguish between two types of rectangle intersections. We say that a pair of intersecting rectangles R, R' have a *corner intersection* iff one of them contains at least one corner of the other rectangle, and otherwise we say that they have a *non-corner intersection*. Our starting point is the natural LP-relaxation of MISR, where each rectangle R is associated with a variable z_R, and the goal is to maximize $\sum_R z_R$, subject to the constraint that for every point p of the plane, the summation of values z_R for all rectangles R containing p is at most 1. We use standard randomized LP-rounding techniques to convert this solution into a "canonical form", where every rectangle has LP-value either 0 or $1/M$ for some $M = \Theta(\log n)$, thus ensuring that every point of the plane is contained in at most M rectangles with non-zero LP-weight. We say that a set $\mathcal{S}$ of rectangles forms a clique iff the intersection of all rectangles in $\mathcal{S}$ is non-empty. The size of the clique is the cardinality of $\mathcal{S}$.

If the input set of rectangles only contains non-corner intersections, and corner intersections are not allowed, a simple LP-rounding procedure gives a constant approximation:[3] For each rectangle $R \in \mathcal{R}$, let $v(R)$ be the size of the maximum clique induced by rectangles intersecting R whose width is smaller than that of R. Output the set $S_i = \{R \mid v(R) = i\}$ of maximum cardinality over all $i : 0 \leq i \leq M-1$. On the other hand, if the input instance only contains corner intersections, and non-corner intersections are not allowed, Lewin-Eytan et al. [23] show a factor-4 approximation algorithm. The main challenge is therefore handling both types of intersections simultaneously.

A naive approach is to generalize the algorithm for non-corner intersections as follows. For each rectangle R, let $v(R)$ denote the size of a maximum clique induced by rectangles intersecting R in a non-corner manner, whose width is smaller than the width of R. We can now partition all rectangles into sets $S_0, \ldots, S_{M-1}$ according to values $v(R)$ as before, and then solve the problem on the largest subset S_i. However, even though each S_i is guaranteed to only contain corner intersections, rectangles in S_i may induce cliques whose size is as large as M, and therefore recovering more than a $1/M$-fraction of rectangles from S_i may be impossible.

Our algorithm combines the two approaches as follows. We perform $\Theta(\log\log n)$ iterations. In the first iteration, we start with a coarse partition of the input set $\mathcal{R}$ of rectangles into subsets $S'_1, \ldots, S'_k$ according to the values of $v(R)$, where k is some constant. In each subsequent iteration we remove large cliques from the sets S'_i and define a new, refined partition, that serves as the input to the next iteration. In the final partition, obtained after $\Theta(\log\log n)$ iterations, each set only contains constant-size cliques.

We also provide a lower bound asymptotically approaching $3/2$ on the integrality gap of the LP relaxation we are using.

Organization. We start with preliminaries in Section 2. Section 3 is devoted to proving Theorem 1.1, and the proof of Theorem 1.2 appears in Section 4. We provide a lower bound on the integrality gap of the LP relaxation in Appendix C

2 Preliminaries

The input to the MISR problem is a set $\mathcal{R}$ of n axis-parallel rectangles. We assume that each rectangle $R \in \mathcal{R}$ is given by a quadruple $(x^l(R), x^r(R), y^t(R), y^b(R))$ of real numbers, corresponding to the x-coordinates of its left and right boundaries and the y-coordinates of its top and bottom boundaries, respectively.

[3] This special case of the problem can be solved exactly since the corresponding intersection graph is perfect [23].

Furthermore, we assume that the rectangles are closed, i.e., each $R \in \mathcal{R}$ is defined as follows: $R = \{(x,y) \mid x^l(R) \le x \le x^r(R) \text{ and } y^b(R) \le y \le y^t(R)\}$. We say that rectangles R and R' intersect iff $R \cap R' \neq \emptyset$. A set $\mathcal{S} \subseteq \mathcal{R}$ of rectangles is called *independent* iff no pair of rectangles in $\mathcal{S}$ intersect. The goal of the MISR problem is to find a maximum-cardinality independent set of rectangles. We assume w.l.o.g. that no rectangle of $\mathcal{R}$ is contained in another rectangle.

For any pair R, R' of intersecting rectangles, we say that they have a *corner intersection* iff one of the two rectangles contains at least one corner of the other. Otherwise, their intersection is called *non-corner intersection* (see Figure 1). Notice that in the case of a non-corner intersection, it is impossible that the widths of R and of R' are identical; if the width of R is smaller than the width of R', then $x^l(R') < x^l(R) < x^r(R) < x^r(R')$ and $y^t(R) > y^t(R') > y^b(R') > y^b(R)$.

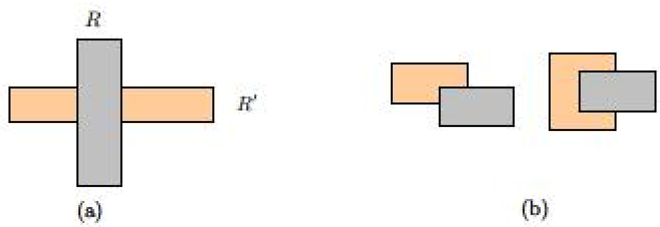

Figure 1: (a): a non-corner intersection; (b): corner intersections

Our starting point is a natural LP-relaxation for MISR. For each rectangle $R \in \mathcal{R}$, we have an indicator variable z_R for choosing R to the solution. Let $X = \{x^l(R), x^r(R) \mid R \in \mathcal{R}\}$ and $Y = \{y^t(R), y^b(R) \mid R \in \mathcal{R}\}$ be the sets of all the x and y coordinates of the corners of input rectangles, respectively. We define $\mathcal{P}$ to be the set of "interesting" points of the plane: $\mathcal{P} = \{(x,y) \mid x \in X \text{ and } y \in Y\}$. The LP relaxation is as follows.

$$\begin{aligned} (LP) \quad \max \quad & \sum_{R \in \mathcal{R}} z_R \\ \text{s.t.} \quad & \sum_{R: p \in R} z_R \le 1 \text{ for all } p \in \mathcal{P} \\ & z_R \ge 0 \text{ for all } R \in \mathcal{R} \end{aligned}$$

Notice that $|\mathcal{P}| \le (2n)^2$, and moreover, if z is a feasible solution for (LP), then for every point p in the plane, $\sum_{R: p \in R} z_R \le 1$. Let OPT denote the value of the optimal feasible solution for (LP).

We say that a set $\mathcal{Q}$ of rectangles forms a *clique* iff the intersection of all rectangles in $\mathcal{Q}$ is non-empty. The size of the clique is the cardinality of $\mathcal{Q}$. Lewin-Eytan, Naor and Orda [23] designed an LP-rounding $(4q)$-approximation algorithm for MISR, where q is the size of the maximum clique in $\mathcal{R}$. We state their result formally in the next theorem and use it in our algorithm.

THEOREM 2.1. *[23] There is a polynomial time algorithm, that, given an instance $\mathcal{R}$ with an LP-solution of cost Z, produces an integral solution whose cost is at least $Z/4q$, where q is the size of the maximum clique in $\mathcal{R}$.*

It will be convenient for us to work with a certain type of solutions for (LP) that we call *canonical solutions*, defined as follows. We use a parameter $M = 64 \log n$. A canonical solution to (LP) is given by a multi-subset $\mathcal{R}'$ of $\mathcal{R}$ (i.e., $\mathcal{R}'$ may contain several copies of each rectangle $R \in \mathcal{R}$), such that the size of the maximum clique in $\mathcal{R}'$ is at most M. Such a set $\mathcal{R}'$ is naturally associated with an LP-solution, where each $R \in \mathcal{R}'$ is assigned an LP-value of $z'_R = 1/M$. Clearly, z' is a feasible LP-solution for instance $\mathcal{R}'$, and it also induces a feasible LP-solution for $\mathcal{R}$, where the LP-value of rectangle $R \in \mathcal{R}$ is the sum of LP-values of its copies in $\mathcal{R}'$. The next lemma states that any solution to (LP) can be converted into a canonical solution, while losing only a constant factor in the LP solution cost. The proof uses standard randomized rounding techniques and can be found in Appendix A.

Lemma 2.1. *There is an efficient randomized algorithm, that, given an optimal LP-solution of cost* OPT *for* $\mathcal{R}$*, produces, with high probability, a feasible canonical solution* $\mathcal{R}'$ *whose associated LP-cost is* $\frac{|\mathcal{R}'|}{M} \geq c\mathsf{OPT}$*, for some constant* c*.*

Let OPT' be the cost of the LP-solution associated with $\mathcal{R}'$, $\mathsf{OPT}' = \Omega(\mathsf{OPT})$. From now on we focus on finding a near-optimal integral solution for $\mathcal{R}'$, by performing LP-rounding of the canonical LP-solution associated with $\mathcal{R}'$. If $R', R'' \in \mathcal{R}'$ are copies of the same rectangle $R \in \mathcal{R}$, then we say that they are *identical*, and we view them as having a corner intersection. To simplify notation, we will refer to $\mathcal{R}'$ as $\mathcal{R}$ and to OPT' as OPT from now on.

We need to define a notion of "thinness" of a rectangle, with respect to other rectangles intersecting it in a non-corner manner. We use this notion later in defining iterative partitions of $\mathcal{R}$.

Definition 2.1. *For each rectangle* $R \in \mathcal{R}$*, let* $V(R)$ *be the set of all rectangles* $R' \in \mathcal{R}$ *such that* R *and* R' *have a non-corner intersection and the width of* R' *is smaller than the width of* R*. Let* $v(R)$ *be the size of the maximum clique in* $V(R)$*.*

Notice that since $\mathcal{R}$ is a canonical solution, $0 \leq v(R) \leq M-1$ for all R. We will view the value $v(R)$ as the measure of "thinness" of R, and it will only be computed once at the beginning of the algorithm. Notice that for each rectangle R, if $\mathcal{Q} \subseteq V(R)$ is a clique of size $v(R)$, then $\left(\bigcap_{R' \in \mathcal{Q}} R'\right) \cap R$ is a rectangle intersecting the upper and the lower boundary of R. It is easy to see that if R and R' have a non-corner intersection then $v(R) \neq v(R')$: assume w.l.o.g. that the width of R is smaller than the width of R'. Then $\{R\} \cup V(R) \subseteq V(R')$ and therefore $v(R') \geq v(R) + 1$.

3 The Algorithm

In this section we prove Theorem 1.1, by presenting an $O(\log \log n)$-approximation algorithm for MISR. We start with a high level overview. Recall that we are given a canonical set $\mathcal{R}$ of rectangles, with maximum clique size bounded by $M = O(\log n)$. A natural approach is to partition $\mathcal{R}$ into M subsets $S_1, \ldots, S_M$, where $R \in S_i$ iff $v(R) = i-1$. If set $\mathcal{R}$ is restricted to only contain non-corner intersections, then every set S_i is an independent set of rectangles, and moreover, set S_i with maximum cardinality over all $1 \leq i \leq M$ contains at least $\mathsf{OPT}' = \Omega(\mathsf{OPT})$ rectangles. However, in our more general setting, where corner intersections are allowed, it is possible that sets S_i contain large cliques, and only a small collection of independent rectangles can be recovered from each of them. Our goal is to build a similar partition of rectangles according to their values of $v(R)$, while ensuring that the maximum clique size in every set of the partition is bounded by a constant. This is done gradually over $\Theta(\log \log n)$ iterations. Each iteration starts from some partition of $\mathcal{R}$ and produces a new, refined partition, while removing large cliques from each set of the partition. The threshold defining large cliques decreases from iteration to iteration, and after the last iteration only constant-sized cliques remain in each set of the partition. In addition to $\mathcal{R}$, the algorithm maintains a set $\mathcal{T}$ containing some rectangles that were removed from $\mathcal{R}$. We ensure that the cost of the fractional solution associated with $\mathcal{T}$ is close to the LP-weight of the removed rectangles, while on the other hand there is an integral solution for $\mathcal{T}$ whose cost is close to that of the fractional solution. The final solution will either contain a subset of $\mathcal{T}$ or a subset of one of the sets S_i from the final partition.

The removal of large cliques from sets S_i is performed as follows. We observe that if some set S_i of rectangles contains a large clique, then there is a rectangle $R \in S_i$ that has a large "α-cover": that is, a quadruple $X_1, X_2, X_3, X_4 \subseteq S_i$ of sets of rectangles, such that the intersection of all rectangles in $\{R\} \cup X_1 \cup X_2 \cup X_3 \cup X_4$ is non-empty and $|X_1| = |X_2| = |X_3| = |X_4| = \alpha$. Moreover, every rectangle in X_1 intersects the top boundary of R, while every rectangle in X_2 intersects the bottom boundary of R. Similarly, rectangles in X_3 and X_4 intersect the left and the right boundaries of R, respectively. We then place R in $\mathcal{T}$ and remove rectangles of X_1, X_2, X_3 and X_4 from the instance, charging their LP-weight to R. The special structure of the α-covers allows us to argue that if the algorithm chooses $\mathcal{T}$ for its output, a good solution can be recovered from it. In order to ensure this, we also maintain a partition of set $\mathcal{T}$ into independent sets. We now proceed to describe the algorithm more formally.

3.1 Algorithm Description We compute values $v(R)$ once at the beginning of the algorithm, and use these original values throughout the algorithm. We use a parameter $\beta = 20$. The algorithm works in iterations. The input to iteration i consists of two disjoint sets $\mathcal{R}_i$, $\mathcal{T}_i$ of rectangles. Set $\mathcal{R}_i$ is partitioned into $k_i = \beta^i$ subsets

$S_1^i, S_2^i, \ldots, S_{k_i}^i$, where set S_j^i contains all rectangles $R \in \mathcal{R}_i$ with $(j-1)M/\beta^i \le v(R) < jM/\beta^i$. Notice that in the first iteration, sets $S_1^1, \ldots, S_{k_1}^1$ define a coarse partition of the rectangles, where each set S_j^1 may contain rectangles with values $v(R)$ differing by logarithmic factors, and in each subsequent iteration the partition becomes finer. Set $\mathcal{T}_i$ is also partitioned into k_i subsets $T_1^i, \ldots, T_{k_i}^i$, and the following properties hold:

C1. For each $j : 1 \le j \le k_i$, the rectangles of T_j^i do not intersect each other and do not intersect rectangles of $S_j^i \cup S_{j-1}^i$.

C2. Each rectangle $R \in \mathcal{T}_i$ is associated with *charge* $c_R \le 10^6 \log\log n/\beta^i$.

C3. Let W_i denote the total LP-weight of rectangles in $\mathcal{R}_i$ (where the weight of each rectangle is $1/M$), and let C_i denote the total charge of rectangles in $\mathcal{T}_i$. Then $W_i + C_i \ge \mathsf{OPT}(1 - 1/\log\log n)^i$.

C4. For each set S_j^i, the maximum clique size is at most $100M/\beta^i$.

Iteration i receives as input sets $\mathcal{R}_i, \mathcal{T}_i$, together with a partition $T_1^i, \ldots, T_{k_i}^i$ of set $\mathcal{T}_i$ and a charge $c : \mathcal{T}_i \to \mathbb{R}_{>0}$ for which properties C1–C4 hold. The output of the iteration is a set $\mathcal{R}_{i+1} \subseteq \mathcal{R}_i$ and $\mathcal{T}_{i+1}$, together with partition $\left\{T_j^{i+1}\right\}_{j=1}^{k_{i+1}}$ of $\mathcal{T}_{i+1}$ and charge c' for rectangles in $\mathcal{T}_{i+1}$, for which properties C1–C4 hold. We show in the next section how to implement each iteration. Assuming this can be done in polynomial time, we now show that an $O(\log\log n)$-approximation for MISR follows.

The input to the first iteration is $\mathcal{R}_1 = \mathcal{R}$, $\mathcal{T}_1 = \emptyset$. It is easy to see that conditions C1–C4 hold for this input. We then run the algorithm for h iterations, where h is the smallest integer for which $\beta^h \ge M/100$. It is easy to see that $h = \Theta(\log\log n)$. Let $\mathcal{R}_{h+1}, \mathcal{T}_{h+1}$ be the output of the last iteration. By property C3, $W_{h+1} + C_{h+1} \ge \mathsf{OPT}(1 - 1/\log\log n)^{h+1} \ge \Omega(\mathsf{OPT})$. Therefore, either $W_{h+1} \ge \Omega(\mathsf{OPT})$, or $C_{h+1} \ge \Omega(\mathsf{OPT})$. Assume first that $W_{h+1} \ge \Omega(\mathsf{OPT})$, and consider the final partition $S_1^{h+1}, \ldots, S_{k_{h+1}}^{h+1}$ of $\mathcal{R}_{h+1}$. We thus have $k_{h+1} = \beta^{h+1} \ge M/100$ sets S_j^{h+1}, and the total LP-weight of rectangles in all these sets is $\Omega(\mathsf{OPT})$, while the weight of each rectangle is $1/M$. Therefore, there are $\Omega(M \cdot \mathsf{OPT})$ rectangles in $\cup_{j=1}^{k_{h+1}} S_j^{h+1}$, and at least one set S_j^{h+1} contains $\Omega(\mathsf{OPT})$ rectangles. By property C4, the size of the maximum clique in S_j^{h+1} is bounded by some constant q. We can now define a new feasible LP-solution for set S_j^{h+1} of rectangles, where each rectangle $R \in S_j^{h+1}$ is assigned an LP-value $z_R = 1/q$. The value of this LP-solution is $\Omega(\mathsf{OPT}/q)$, and by Theorem 2.1, we can recover $\Omega(\mathsf{OPT}/q^2) = \Omega(\mathsf{OPT})$ independent rectangles in S_j^{h+1}.

Assume now that $C_{h+1} \ge \Omega(\mathsf{OPT})$. Again we have $k_{h+1} \ge M/100$ sets T_j^{h+1}, for $1 \le j \le k_{h+1}$, whose total charge is $\Omega(\mathsf{OPT})$. Thus, at least one set T_j^{h+1} has charge $\Omega(\mathsf{OPT}/M)$. From property C2, for each $R \in T_j^{h+1}$, the charge $c_R \le 10^6 \log\log n/\beta^{h+1}$. Therefore, $|T_j^{h+1}| \ge \Omega(\mathsf{OPT}/M)/(10^6 \log\log n/\beta^{h+1}) = \Omega(\mathsf{OPT}/\log\log n)$. Since the rectangles of T_j^{h+1} do not intersect, we get an $O(\log\log n)$-approximation.

3.2 Iteration Description Consider some iteration i. We are given as input a subset $\mathcal{R}_i$ of rectangles, which is subdivided into sets S_j^i for $1 \le j \le k_i$, where $k_i = \beta^i$. Set S_j^i contains all rectangles $R \in \mathcal{R}_i$ with $(j-1)M/\beta^i \le v(R) < jM/\beta^i$. We are also given a set $\mathcal{T}_i$ of rectangles, which is subdivided into subsets T_j^i for $1 \le j \le k_i$, together with charge c. We assume that properties C1–C4 hold for this input. Our goal is to produce subsets $\mathcal{R}_{i+1}, \mathcal{T}_{i+1}$ together with the corresponding partitions $\left\{S_j^{i+1}\right\}_{j=1}^{k_{i+1}}$, $\left\{T_j^{i+1}\right\}_{j=1}^{k_{i+1}}$ and charge c' for rectangles in $\mathcal{T}_{i+1}$ for which properties C1–C4 hold. The algorithm has three major steps. In the first step, we perform an initial partition of sets $\mathcal{R}_i$ and $\mathcal{T}_i$, into subsets $\left\{S_j^{i+1}\right\}_{j=1}^{k_{i+1}}$, $\left\{T_j^{i+1}\right\}_{j=1}^{k_{i+1}}$. This step will ensure properties C1, C2 and C3. However, property C4 does not necessarily hold, and some of the sets S_j^{i+1} may have large cliques. The next two steps take care of this, while preserving the other properties. In the second step we remove from $\mathcal{R}_i$ those rectangles (called bad rectangles) that have corner intersections with many other rectangles. Finally, in the last step, which is the heart of our algorithm, we repeatedly find large cliques in sets S_j^{i+1}, for $1 \le j \le k_{i+1}$, and remove them, while adding one of the clique rectangles to $\mathcal{T}_{i+1}$ and charging the LP-weight of all the clique rectangles to it. The removal of bad rectangles in Step 2 allows us to bound the charge as required by Property C2. We now proceed to describe these three steps more formally.

Step 1: Initial Partition. We start with $\mathcal{R}_{i+1} = \mathcal{R}_i$, and partition set $\mathcal{R}_{i+1}$ into $k_{i+1} = \beta^{i+1}$ sets, $\left\{S_j^{i+1}\right\}_{j=1}^{k_{i+1}}$, as follows. Set S_j^{i+1} contains all rectangles $R \in \mathcal{R}_{i+1}$ with $(j-1)M/\beta^{i+1} \le v(R) < jM/\beta^{i+1}$. Notice that this partition is a refinement of the original partition $\left\{S_j^i\right\}_{j=1}^{k_i}$ of $\mathcal{R}_i$, where the rectangles of S_j^i now belong to sets $S_{(j-1)\beta+1}^{i+1}, \ldots, S_{j\beta}^{i+1}$.

Set $\mathcal{T}_i$ is used to produce k_{i+1} subsets $T_1^{i+1}, \ldots, T_{k_{i+1}}^{i+1}$ as follows. For each $j : 1 \le j \le k_i$, sets $T_{(j-1)\beta+1}^{i+1}, \ldots, T_{j\beta}^{i+1}$ are copies of T_j^i. The charge of each rectangle $R \in \mathcal{T}_i$ is split evenly among the β new copies. Let $\mathcal{T}_{i+1}$ denote the union of rectangles in sets T_j^{i+1}, for $1 \le j \le k_{i+1}$. We will use the following properties of the new partition:

D1. For each $j : 1 \le j \le k_{i+1}$, the rectangles of T_j^{i+1} do not intersect each other and do not intersect rectangles of $S_j^{i+1} \cup S_{j-1}^{i+1} \cup S_{j-2}^{i+1}$: To see this, let j' be the index for which $S_j^{i+1} \subseteq S_{j'}^i$. Then $T_j^{i+1} = T_{j'}^i$, and $S_j^{i+1} \cup S_{j-1}^{i+1} \cup S_{j-2}^{i+1} \subseteq S_{j'}^i \cup S_{j'-1}^i$. We can now use property C1.

D2. For each rectangle $R \in \mathcal{T}_{i+1}$, the new charge c ensures that $c_R \le 10^6 \log\log n/\beta^{i+1}$. This follows since the original instance had property C2, and the new charges are defined to be $1/\beta$-fraction of the original charges.

D3. If W is the total LP-weight of rectangles in $\mathcal{R}_{i+1}$ (where the weight of each rectangle is $1/M$), and C is the total charge of rectangles in $\mathcal{T}_{i+1}$, then $W + C \ge \mathsf{OPT}(1 - 1/\log\log n)^i$. This property holds due to property C3, and since the total charge and the LP-weight of rectangles have not been changed.

Step 2: Removal of Bad Rectangles. Let $\mathcal{H}$ be any set of rectangles with maximum clique size M', and let $R \in \mathcal{H}$ be any rectangle. For $\gamma > 0$, we say that R is (γ, M')-bad with respect to $\mathcal{H}$, if R has corner intersections with at least $\gamma M'$ rectangles of $\mathcal{H}$. The next claim shows that the number of bad rectangles in $\mathcal{H}$ is small. The proof uses averaging arguments similar to those used in Lemma 1 in [23] and can be found in Appendix B.

Claim 3.1. *For any set $\mathcal{H}$ of rectangles with maximum clique size M', the number of (γ, M')-bad rectangles w.r.t. $\mathcal{H}$ is at most $\frac{4}{\gamma}|\mathcal{H}|$.*

Consider some rectangle $R \in \mathcal{R}_{i+1}$, and assume that $R \in S_j^{i+1}$. We say that R is a *bad rectangle* iff it is (γ, M')-bad w.r.t. $S_j^{i+1} \cup S_{j+1}^{i+1} \cup S_{j+2}^{i+1}$, for parameters $\gamma = 12 \log\log n$, $M' = 300M/\beta^i$. Due to Property C4, the size of the maximum clique in $S_j^{i+1} \cup S_{j+1}^{i+1} \cup S_{j+2}^{i+1}$ is at most M'.

In Step 2 of the algorithm, we remove all bad rectangles from $\mathcal{R}_{i+1}$. From Claim 3.1, the number of rectangles removed from $S_j^{i+1} \cup S_{j+1}^{i+1} \cup S_{j+2}^{i+1}$ is bounded by $\frac{1}{3\log\log n}|S_j^{i+1} \cup S_{j+1}^{i+1} \cup S_{j+2}^{i+1}|$, and therefore, we remove at most $\frac{1}{\log\log n}|\mathcal{R}_{i+1}|$ rectangles from $\mathcal{R}_{i+1}$ in this step overall. While properties D1 and D2 continue to hold, if we denote by W the total LP-weight of the rectangles remaining in $\mathcal{R}_{i+1}$ and by C the total charge to rectangles in $\mathcal{T}_{i+1}$, we now have that $W + C \ge \mathsf{OPT}(1 - 1/\log\log n)^{i+1}$. Therefore, current partitions $\left\{S_j^{i+1}\right\}, \left\{T_j^{i+1}\right\}$ of sets $\mathcal{R}_{i+1}, \mathcal{T}_{i+1}$ satisfy properties C1, C2 and C3. Our goal is now to ensure property C4, while maintaining these properties.

Step 3: Removal of Large Cliques. This step is the heart of our algorithm. To simplify notation, we refer from now on to sets S_j^{i+1} as S_j and to sets T_j^{i+1} as T_j. We also denote $\mathcal{R}_{i+1}, \mathcal{T}_{i+1}$ and k_{i+1} by $\mathcal{R}'$, $\mathcal{T}'$ and k, respectively. One of our main observations is that the existence of a large clique implies a large α-coverage, which is defined as follows.

Definition 3.1. *Let $R \in S_j$ for some j, and let $X_1, X_2, X_3, X_4 \subseteq S_j$ be distinct collections of rectangles, $|X_1| = |X_2| = |X_3| = |X_4| = \alpha$. We say that they form an α-covering of R iff:*

- *Each rectangle in X_1 (resp. X_2) intersects the top (resp. bottom) boundary of R, and each rectangle in X_3 (resp. X_4) intersects the left (resp. right) boundary of R.*
- *There is a point p contained in every rectangle in set $X_1 \cup X_2 \cup X_3 \cup X_4 \cup \{R\}$.*

For each $R \in S_j$, for $1 \le j \le k$, we denote by $\alpha(R)$ the maximum value α such that there exist $X_1, X_2, X_3, X_4 \subseteq S_j$ that form an α-covering of R.

The following claim shows that whenever there is a large clique in any set S_j, there must be a rectangle R with large covering number $\alpha(R)$. In the rest of the algorithm we will repeatedly find such rectangles and reduce the clique size by moving them from $\mathcal{R}'$ to $\mathcal{T}'$, while removing some of the rectangles from $\mathcal{R}'$ completely.

Claim 3.2. *Consider set S_j for any $j : 1 \leq j \leq k$, and assume that S_j contains a clique of size c. Then there is at least one rectangle $R \in S_j$ with $\alpha(R) \geq \lfloor c/4 \rfloor - 1$.*

Proof. Let $\mathcal{C} \subseteq S_j$ be any clique of size c, and let p be a point contained in every rectangle of $\mathcal{C}$. Let $X_1 \subseteq \mathcal{C}$ denote the set of $\lfloor c/4 \rfloor - 1$ rectangles R with highest top boundaries (i.e., largest values of $y^t(R)$), breaking ties arbitrarily. Let $X_2 \subseteq \mathcal{C} \setminus X_1$ denote the set of $\lfloor c/4 \rfloor - 1$ rectangles with lowest bottom boundaries (smallest values of $y^b(R)$), breaking ties arbitrarily. Define X_3 and X_4 similarly for the left and right boundaries. Let R be any rectangle in $\mathcal{C} \setminus (X_1 \cup X_2 \cup X_3 \cup X_4)$.

It is easy to see that every rectangle in X_1 (resp. X_2) intersects the upper (resp. lower) boundary of R, and every rectangle in X_3 (resp. X_4) intersects the left (resp. right) boundary of R. Thus, (X_1, X_2, X_3, X_4) is a $(\lfloor c/4 \rfloor - 1)$-covering of R. □

If S_j does not contain rectangles R with $\alpha(R) \geq 20M/\beta^{i+1} = 20M/k$, then Claim 3.2 ensures that the size of the largest clique in S_j is at most $100M/\beta^{i+1}$, and Property C4 holds in S_j. The next claim shows that the last two sets S_{k-1} and S_k in the partition cannot contain large cliques.

Claim 3.3. *If $\alpha(R) \geq 20M/k$ and $R \in S_j$, then $j \leq k-2$.*

Proof. Assume for contradiction that $j > k-2$. Consider set $\mathcal{C} \subseteq V(R)$ of rectangles that form a clique of size $v(R)$. Since $j > k-2$, $v(R) \geq M - 2M/k$ must hold. Let ℓ be any vertical line segment contained in $R \cap \left(\bigcap_{R' \in \mathcal{C}} R'\right)$, such that ℓ contains a point on the top and a point on the bottom boundary of R. We denote the x-coordinate of ℓ by x_ℓ.

Let (X_1, X_2, X_3, X_4) be the $\alpha(R)$-cover of R, and let $p = (x', y')$ be a point contained in the intersection of all rectangles in $\{R\} \cup X_1 \cup X_2 \cup X_3 \cup X_4$. We assume w.l.o.g. that $x_\ell \leq x'$, and the other case is symmetric. Consider the point $q = (x_\ell, y')$ (see Figure 2). We claim that more than M rectangles contain q. First, since q lies on ℓ, every rectangle in $\mathcal{C}$ contains q. Additionally, since every rectangle in X_3 contains p and intersects the left boundary of R, q also belongs to every rectangle of X_3. Moreover, $X_3 \cap \mathcal{C} = \emptyset$, since $\mathcal{C}$ only contains rectangles in $V(R)$, while all rectangles in X_3 intersect the left boundary of R and therefore do not belong to $V(R)$. Therefore, the number of rectangles containing q is at least $v(R) + |X_3| \geq M - 2M/k + 20M/k > M$. This is a contradiction since $\mathcal{R}$ is a canonical set. □

We are now ready to describe the algorithm executed at Step 3. While there exist rectangles $R \in \mathcal{R}'$ with $\alpha(R) \geq 20M/k$ (where $\alpha(R)$ is computed with respect to set S_j to which R belongs), we select such a rectangle R that has smallest width, breaking ties arbitrarily. If $R \in S_j$, then by Claim 3.3, $1 \leq j \leq k-2$. We then run the following procedure.

Round(R)

Assume that $R \in S_j$, and let X_1, X_2, X_3, X_4 be the corresponding α-coverage with point p contained in all rectangles of $X_1 \cup X_2 \cup X_3 \cup X_4 \cup \{R\}$.

- Remove R from $\mathcal{R}'$ and add it to T_{j+2}.
- Remove all rectangles $R' \in S_j \cup S_{j+1} \cup S_{j+2}$ containing p from $\mathcal{R}'$.

  Let $R_l^1 \in X_1$ be the rectangle minimizing the value of $x^l(R)$ in X_1, and let $R_r^1 \in X_1$ be the rectangle maximizing value $x^r(R)$ in X_1, breaking ties arbitrarily. Similarly, let $R_l^2, R_r^2 \in X_2$ be the two rectangles minimizing $x^l(R)$ and maximizing $x^r(R)$ in X_2, respectively.
- Remove from $\mathcal{R}'$ all rectangles $R' \in S_j \cup S_{j+1} \cup S_{j+2}$ that have corner-intersections with at least one of the rectangles in $\left\{R, R_l^1, R_r^1, R_l^2, R_r^2\right\}$.
- Set the charge of R to be the LP-weight of all rectangles removed from $\mathcal{R}'$ by this procedure.

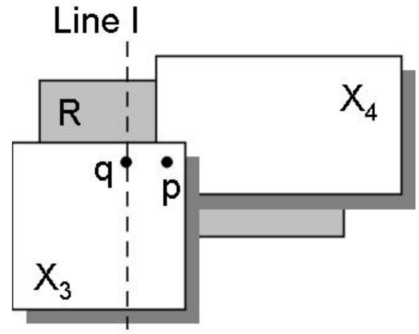

Figure 2: Illustration for proof of Claim 3.3

We now prove that properties C1–C4 hold when the algorithm terminates. Due to Claim 3.2, once the algorithm stops, the maximum clique size in any set S_j, $1 \le j \le k$ is at most $4 \cdot (20M/k + 2) \le 100M/k$, and therefore property C4 holds. Moreover, observe that during the rectangle rounding step the sum of total charge of rectangles in $\mathcal{T}'$ and the LP-weight of rectangles in $\mathcal{R}'$ do not change. This is since the LP-weight of every rectangle removed from $\mathcal{R}'$ is charged to some rectangle newly added to $\mathcal{T}'$. This ensures property C3. We now prove that property C2 is also preserved.

Claim 3.4. *The charge of each rectangle R that is added to $\mathcal{T}$ by procedure* Round(R) *is bounded by* $10^6 \log\log n/k$.

Proof. Recall that we only remove the following rectangles. (1) Rectangles in set $S_j \cup S_{j+1} \cup S_{j+2}$ containing p: due to property C4 in the input to the current iteration, the maximum clique size in each such set is bounded by $100M/\beta^i$, and thus the total LP-weight of these rectangles is at most $300/\beta^i = 6000/\beta^{i+1}$. (2) Rectangles in set $S_j \cup S_{j+1} \cup S_{j+2}$ that have corner intersection with one of the rectangles $\left\{R, R_l^1, R_r^1, R_l^2, R_r^2\right\}$. Since we have removed all bad rectangles from $\mathcal{R}'$, each rectangle in $\left\{R, R_l^1, R_r^1, R_l^2, R_r^2\right\}$ intersects with at most $(12 \log\log n) \cdot (300M/\beta^i) = 3600M \log\log n/\beta^i$ rectangles. The total LP-weight of such rectangles is at most $5 \cdot \beta \cdot 3600 \log\log n/\beta^{i+1}$. Therefore, the total charge to R is bounded by $10^6 \log\log n/\beta^{i+1}$ as required. □

Finally, the next theorem shows that Property C1 is preserved as well, completing the proof of Theorem 1.1.

Theorem 3.1. *For every $1 \le j \le k$, rectangles of T_j do not intersect each other and do not intersect rectangles in $S_j \cup S_{j-1}$.*

Proof. Recall that before the rounding steps were performed, property D1 ensured that for each j, the rectangles of T_j do not intersect each other and do not intersect rectangles in set $S_j \cup S_{j-1} \cup S_{j-2}$. It is enough to prove the following lemma:

Lemma 3.1. *Whenever rectangle $R \in S_j$ is added to T_{j+2} due to procedure* Round(R)*, it does not intersect any rectangle in $T_{j+2} \cup S_{j+2} \cup S_{j+1}$. Moreover, there is no rectangle $R' \in S_j$ that intersects R in a non-corner manner, whose width is greater than the width of R.*

Proof. The proof is by induction on the order in which rectangles R are added to T_{j+2}. Assume first for contradiction that there is a rectangle $R' \in T_{j+2}$ that intersects with R. If R' belonged to T_{j+2} before the execution of Step 3, then Property D1 ensures that R and R' do not intersect. Otherwise, R' has been added to T_{j+2} during Step 3 before R and so its width is smaller than the width of R. When procedure Round(R') was executed, we have removed from $S_j \cup S_{j+1} \cup S_{j+2}$ all rectangles that have corner intersection with R', and therefore the intersection between R and R' must be a non-corner one. Due to the induction hypothesis it is then impossible that R and R' intersect.

Assume now for contradiction that R intersects some rectangle $R' \in S_{j+1} \cup S_{j+2}$. Since we remove all rectangles in $S_j \cup S_{j+1} \cup S_{j+2}$ that have corner-intersections with R during procedure $\text{Round}(R)$, it must be a non-corner intersection. Moreover, since $R \in S_j$ while $R' \in S_{j+1} \cup S_{j+2}$, $v(R) < v(R')$ and therefore the width of R' must be greater than the width of R. To finish the proof of the theorem, it now remains to show that if R and R' have a non-corner intersection and the width of R' is greater than the width of R then $R' \notin S_j \cup S_{j+1} \cup S_{j+2}$. The next claim will then finish the proof of the theorem.

Claim 3.5. *Let R' be any rectangle intersecting R in a non-corner manner, such that the width of R' is greater than the width of R. Then $v(R') \geq v(R) + \alpha(R)/4 \geq v(R) + 5M/k$. It follows that $R' \notin S_j \cup S_{j+1} \cup S_{j+2}$.*

Proof. Consider the $\alpha(R)$-covering $(X_1, X_2, X_3, X_4) \subseteq S_j$ of R, and let $p = (x, y)$ be the point contained in every rectangle in set $\{R\} \cup X_1 \cup X_2 \cup X_3 \cup X_4$. Since R' was not removed during procedure $\text{Round}(R)$, it does not contain point p. Assume w.l.o.g. that R' lies strictly above p (the other case is symmetric).

We first show that for every rectangle $P \in X_1$, R' intersects P in a non-corner manner, and the width of R' is greater than the width of P (see Figure 3). Let $P \in X_1$. Recall that P intersects the top boundary of R and contains point p, while R' lies strictly between the top boundary of R and point p. Therefore, R' cannot contain corners of P. On the other hand, since R' was not removed by procedure $\text{Round}(R)$, R' and R_l^1 intersect in a non-corner manner, and R' and R_r^1 intersect in a non-corner manner. This can only happen if the left boundary of R' lies completely to the left of the left boundary of R_l^1 (and thus to the left of P), and the right boundary of R' lies to the right of the right boundary of R_r^1 (and thus to the right of P). Thus, $X_1 \subseteq V(R')$.

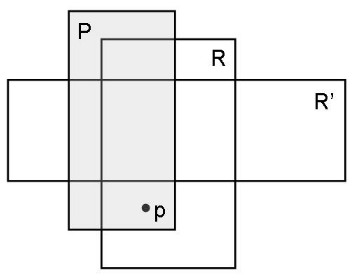

Figure 3: Illustration for claim that $X_1 \subseteq V(R')$

Consider now the vertical line L passing through p. Let $\mathcal{Q} \subseteq X_1$ be the set of $\lfloor |X_1|/2 \rfloor$ rectangles whose left boundary is closest to L in X_1, and let $P \in \mathcal{Q}$ be the rectangle whose right boundary is closest to L in $\mathcal{Q}$ (see Figure 4). Notice that all rectangles in $X_1 \setminus \mathcal{Q}$ intersect the left boundary of P, and all rectangles in $\mathcal{Q}$ intersect the right boundary of P. Therefore, no rectangle of X_1 may belong to $V(P)$. Let $\mathcal{C}' \subseteq V(P)$ be a clique of size $v(P)$. Let $q = (x', y')$ be any point in the intersection of rectangles in $\mathcal{C}' \cup \{P, R'\}$. Assume first that $x' \leq x$. Then every rectangle in $X_1 \setminus \mathcal{Q}$ contains q, and $\mathcal{C}' \cup (X_1 \setminus \mathcal{Q}) \subseteq V(R')$ form a clique of size at least $v(P) + \lceil \alpha(R)/2 \rceil \geq v(R) - M/k + \lceil \alpha(R)/2 \rceil \geq v(R) + \alpha(R)/4$. Similarly, if $x > x'$, then every rectangle in $\mathcal{Q}$ contains q, and $\mathcal{C}' \cup \mathcal{Q} \subseteq V(R')$ form a clique of size at least $v(P) + \lfloor \alpha(R)/2 \rfloor \geq v(R) + \alpha(R)/4$. □ □ □

4 A Generalization To Higher Dimensions

Let $\mathcal{A}$ be any $f(n)$-approximation algorithm for the MISB_d problem for any $d \geq 2$. We show that there is an $O(\log n \cdot f(n))$-approximation algorithm for $\mathsf{MISB_{d+1}}$. Our algorithm is similar to the algorithm of Agarwal et al. [1] which can be seen as obtaining an $O(\log n)$-approximation algorithm for MISB_2 from an exact algorithm for MISB_1. The proof of the next theorem, combined with Theorem 1.1 will complete the proof of Theorem 1.2.

Theorem 4.1. *If there is a factor $f(n)$-approximation algorithm $\mathcal{A}$ for MISB_d, then there is an $O(f(n)\log n)$-approximation algorithm for $\mathsf{MISB_{d+1}}$, for any $d \geq 1$.*

Proof. Let $\mathcal{R}$ be an instance of $\mathsf{MISB_{d+1}}$. For each rectangle $R \in \mathcal{R}$, let γ_R be the largest value of $(d+1)$th coordinate of any point in R. Let $y \in \mathbb{R}$ be such that exactly $\lfloor \frac{|\mathcal{R}|}{2} \rfloor$ rectangles of $\mathcal{R}$ have $\gamma_R \leq y$. We partition $\mathcal{R}$

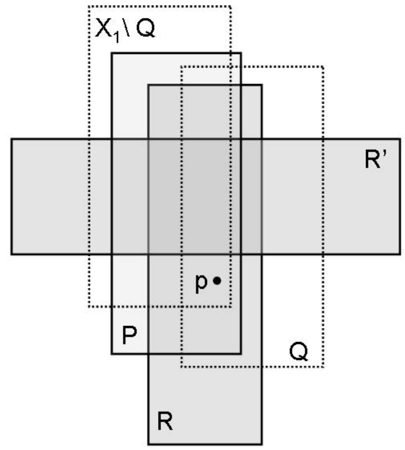

Figure 4: Illustration for proof of Claim 3.5

into three sets, I, I_l and I_r as follows. Let H be the hyperplane $x_{d+1} = y$. Set I contains all rectangles R that intersect H, and sets I_l and I_r contain all rectangles lying completely to the left or to the right of H, respectively. More formally, if we denote the $(d+1)$th coordinate of point p by p_{d+1}, then $I = \{R \in \mathcal{R} | \exists p \in R : p_{d+1} = y\}$, $I_l = \{R \in \mathcal{R} | \forall p \in R : p_{d+1} < y\}$ and $I_r = \{R \in \mathcal{R} | \forall p \in R : p_{d+1} > y\}$.

Let I' be the instance of MISB_d obtained from I by ignoring the last coordinate of each rectangle. We run $\mathcal{A}$ on I', and solve the problem recursively on I_l and I_r. Let S, T_l, T_r be the corresponding solutions. The algorithm outputs either S or $T_l \cup T_r$, whichever contains more rectangles.

We show by induction on the input size that this algorithm produces an $O(f(n) \log n)$-approximation. We consider two cases. First, if the optimal solution cost on I is at least $\mathsf{OPT}/\log n$, then $|S| \geq \mathsf{OPT}/(f(n) \log n)$ since $\mathcal{A}$ is an $f(n)$-approximation algorithm. Otherwise, denote by OPT_l and OPT_r the sizes of maximum independent sets in I_l and I_r respectively. We have that $\mathsf{OPT}_r + \mathsf{OPT}_l \geq \mathsf{OPT}(1 - 1/\log n)$. Using the induction hypothesis, since $|I_l|, |I_r| \leq n/2$, we have that $|T_l| \geq \mathsf{OPT}_l/(f(n)(\log n - 1))$ and $|T_r| \geq \mathsf{OPT}_r/(f(n)(\log n - 1))$. Therefore, $|T_l \cup T_r| \geq \mathsf{OPT}/(f(n) \log n)$. $\square$

5 Conclusion

We have shown an $O(\log \log n)$-approximation algorithm for MISR. However, the exact approximability status of MISR still remains open, and in particular it is interesting whether this problem admits a PTAS. Another interesting open question is establishing the integrality gap of (LP). Our results imply an $O(\log \log n)$ upper bound and a lower bound asymptotically approaching $3/2$. It is also interesting whether our approach can be extended to $O((\log \log n)^d)$-approximation for MISB_d, and whether it can used to obtain sub-logarithmic approximation for the weighted version of MISR.

References

[1] P. K. Agarwal, M. J. van Kreveld, and S. Suri. Label placement by maximum independent set in rectangles. *Computational Geometry*, 11(3-4):209–218, 1998.

[2] P. Berman, B. DasGupta, S. Muthukrishnan, and S. Ramaswami. Improved approximation algorithms for rectangle tiling and packing. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 427–436, 2001.

[3] R. B. Boppana and M. M. Halldórsson. Approximating maximum independent sets by excluding subgraphs. *BIT*, 32(2):180–196, 1992.

[4] T. M. Chan. Polynomial-time approximation schemes for packing and piercing fat objects. *Journal of Algorithms*, 46(2):178–189, 2003.

[5] T. M. Chan. A note on maximum independent sets in rectangle intersection graphs. *Information Processing Letters*, Volume 89, Number 1, pp. 19-23, 2004.

[6] L. S. Chandran, M. C. Francis and N. Sivadasa. Geometric Representation of Graphs in Low Dimension Using Axis Parallel Boxes. *Algorithmica*, to appear.

[7] L. S. Chandran, M.C. Francis, and N. Sivadasan. Boxicity and maximum degree. *Journal of Combinatorial Theory* Series B (2007).

[8] L.S. Chandran, M.C. Francis, N. Sivadasan. Boxicity and Treewidth. *Journal of Combinatorial Theory* Series B 97(5), 733744 (2007).

[9] Miroslav Chlebík and Janka Chlebíková. Approximation hardness of optimization problems in intersection graphs of d-dimensional boxes. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 267–276, 2005.

[10] M. B. Cozzens. Higher and multidimensional analogues of interval graphs. Ph.D. thesis, Rutgers University, New Brunswick, NJ (1981).

[11] J. S. Doerschler and H. Freeman. A rule-based system for dense-map name placement. *Communication of The ACM* 35, 1, pp. 68-79, 1992.

[12] Thomas Erlebach, Klaus Jansen, and Eike Seidel. Polynomial-time approximation schemes for geometric intersection graphs. *SIAM Journal on Computing*, 34(6):1302–1323, 2005.

[13] R.J. Fowler, M.S. Paterson and S.L. Tanimoto, Optimal packing and covering in the plane are NP-complete. *Information Processing Letter.* 12 3 (1981), pp. 133137.

[14] T. Fukuda, Y. Morimoto, S. Morishita and T. Tokuyama. Data Mining with optimized two-dimensional association rules. *ACM Trans. Database Syst.* 26, 2, pp. 179-213, 2001.

[15] A. G. Greenberg and D. Wischik. Admission Control for Booking Ahead Shared Resources, In *Proceedings IEEE INFOCOM '98*, pp. 873–882, 1998.

[16] R. A. Guerin and A. Orda. Networks with advance reservations: The routing perspective. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies* (INFOCOM), 2000.

[17] J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

[18] Dorit S. Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and vlsi. *Journal of the ACM*, 32(1):130–136, 1985.

[19] H. Imai, T. Asano. Finding the Connected Components and a Maximum Clique of an Intersection Graph of Rectangles in the Plane. *Journal of Algorithms* 4(4): 310-323, 1983.

[20] S. Khanna, S. Muthukrishnan, and M. Paterson. On approximating rectangle tiling and packing. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 384–393, 1998.

[21] Jan Kratochvíl A Special Planar Satisfiability Problem and a Consequence of Its NP-completeness. *Discrete Applied Mathematics*, 52(3):233–252, 1994.

[22] B. Lent, A. Swami, and J. Widom. Clustering association rules. In *Proceedings of the International Conference on Data Engineering* (ICDE'97), pages 220–231, 1997.

[23] Liane Lewin-Eytan, Joseph Naor, and Ariel Orda. Routing and admission control in networks with advance reservations. In *Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, pages 215–228, 2002.

[24] A. Lingas and M. Wahlen. A note on maximum independent set and related problems on box graphs. *Information Processing Letter*, 93(4):169–171, 2005.

[25] Frank Nielsen. Fast stabbing of boxes in high dimensions. *Theoretical Computer Science*, 246(1-2):53–72, 2000.

[26] F.S. Roberts. On the boxicity and Cubicity of a graph. In: *Recent Progresses in Combinatorics*, pp. 301-310. Academic, New York (1969).

[27] B. Rosgen, L. Stewart. Complexity results on graphs with few cliques. *Discrete Math. Theor. Comput. Sci.* 9, 127136 (2007) 21. Scheinerman, E.R.: Intersection

[28] E.R. Scheinerman, Intersection classes and multiple intersection parameters. Ph.D. thesis, Princeton University (1984).

[29] O. Schelén and S. Pink. Resource sharing in advance reservation agents. *Journal of High Speed Networks* 7, 3-4, pp. 213-228, 1998.

[30] A. Schill, F. Breiter, and S. Kühn. Design and evaluation of an advance reservation protocol on top of RSVP. In *Proceedings of The International Conference on Broadband Communications, Stuttgart, Chapman & Hall*, pp. 23–40, 1998.

[31] C. Thomassen. Interval representations of planar graphs. *Journal of Combinatorial Theory* Series B 40, 920 (1986).

[32] Mihalis Yannakakis. The complexity of the partial order dimension problem. *SIAM Journal on Algebraic Discrete Methods*, 3(3):351-358, 1982.

A Proof of Lemma 2.1

Proof. For each rectangle $R \in \mathcal{R}$, let $c_R = \lceil z_R M \rceil$. We create c_R copies of R, with the first $c_R - 1$ copies having LP-value $1/M$, and the last copy with LP-value $z_R - (c_R - 1)/M$. Let $\mathcal{R}^*$ be the resulting multi-set of rectangles and z^* be its corresponding LP-solution. Clearly, the sum of LP-values z^* of copies of R in $\mathcal{R}^*$ is exactly z_R.

We now construct set $\mathcal{R}'$ as follows. Each rectangle $R \in \mathcal{R}^*$ is independently selected to $\mathcal{R}'$ with probability $\frac{M}{2} z^*_R$. Let z' be the LP-solution for set $\mathcal{R}'$, where every rectangle is assigned value $1/M$.

Observe first that the expected solution value is $\mathsf{OPT}/2$. Using Chernoff bound, the probability that the LP-value of z' is less than $\mathsf{OPT}/4$ is at most $e^{-\mathsf{OPT}/16} < 1/4$ (assuming that $\mathsf{OPT} > c$ for some constant c).

To show that the solution is feasible, it is enough to prove that for each point $p \in \mathcal{P}$, the probability that more than M rectangles containing p are in $\mathcal{R}'$ is at most $1/n^4$. We can then use the union bound to show that the solution is feasible with high probability.

For any point $p \in \mathcal{P}$, let $\mathcal{C}_p$ be the set of rectangles in $\mathcal{R}'$ containing p. Observe that $\mathbf{E}\left[\, |\mathcal{C}_p| \,\right] \le M/2$. Using Chernoff bound, we get

$$\mathbf{Pr}\left[|\mathcal{C}_p| \ge M\right] \le e^{-M/16} \le \frac{1}{n^4}$$

Applying the union bound to the n^2 points in $\mathcal{P}$, we get that with high probability the solution is a feasible solution. Thus, with constant probability, we get a feasible solution whose LP-value is at least $\mathsf{OPT}/4$.

B Proof of Claim 3.1

Proof. For each rectangle $R \in \mathcal{H}$, let $\mathcal{I}(R)$ be the set of rectangles identical to R in $\mathcal{H}$, including R. For two non-identical rectangles R, R', let $c(R, R')$ be the number of corners of R contained in R'. Observe that $c(R, R') \in \{0, 1, 2\}$, and, since the maximum clique size in $\mathcal{R}'$ is at most M', for each rectangle R, $|\mathcal{I}(R)| + \sum_{R' \notin \mathcal{I}(R)} c(R, R') \le 4M'$. For each rectangle $R \in \mathcal{H}$, let $N_{\mathcal{H}}(R)$ be the number of rectangles in $\mathcal{H}$ that have corner intersections with R. We can now bound the average value of $N_{\mathcal{H}}(R)$ as follows:

$$\begin{aligned}
\sum_{R \in \mathcal{H}} |N_{\mathcal{H}}(R)| &= |\{(R, R') : R, R' \text{ corner-intersect }\}| \\
&\le \sum_{R \in \mathcal{H}} \left(|\mathcal{I}(R)| + \sum_{R' \in \mathcal{H} \setminus \mathcal{I}(R)} c(R, R') \right) \\
&\le 4M' |\mathcal{H}|
\end{aligned}$$

To see that the first inequality is true, observe that there are two types of possible corner-intersections for non-identical rectangles R and R': either each rectangle contains exactly one corner of the other, in which case this intersection contributes 2 to both sides of the inequality; or one rectangle contains two corners of the other, in which case again the intersection contributes 2 to both sides of the inequality.

We can now use standard averaging arguments to complete the claim. Assume for contradiction that there are more than $\frac{4}{\gamma}|\mathcal{H}|$ rectangles that are (γ, M')-bad. Since each such rectangle has $|N_{\mathcal{H}}(R)| > \gamma M'$, we get that $\sum_{R \in \mathcal{H}} |N_{\mathcal{H}}(R)| > 4M'|\mathcal{H}|$, a contradiction.

C A Lower Bound on Integrality Gap

In this section we show a lower bound on the integrality gap of (LP) that asymptotically approaches $3/2$. Our instance only contains corner intersections. Our construction is recursive, and we construct a sequence $I_1, \ldots, I_n$ of instances, where the integrality gap of I_n is $\frac{3n+2}{2(n+1)}$.

Each instance I_j contains three special rectangles: R_1^j, R_2^j and R_3^j. Additionally, we have a "virtual" rectangle R_v^j that has the property that all rectangles of I_j excluding R_1^j, R_2^j, R_3^j are contained in R_v^j. We notice that R_v^j is not part of the problem instance, but is convenient to use as a bounding box for all rectangles in I_j excluding R_1^j, R_2^j, R_3^j. Figure 5 shows a schematic view of I_j.

Instance I_1 contains five rectangles, whose intersection graph is just a 5-cycle, as shown in Figure 6.

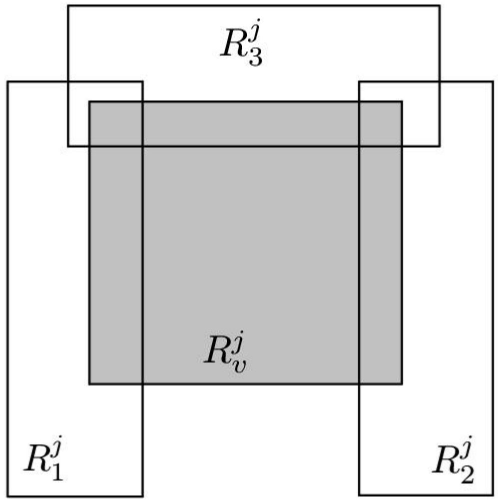

Figure 5: Schematic View of I_j

In order to obtain instance I_{j+1} from instance I_j we proceed as follows. First, we rotate I_j clockwise by 90 degrees. Next, we place R_1^{j+1} to the left of I_j, intersecting R_1^j and R_2^j but not intersecting R_v^j. Similarly, R_2^{j+1} is placed to the right of I_j, intersecting R_3^j but not R_v^j. Finally, R_3^{j+1} is placed above I_j, intersecting R_1^{j+1} and R_2^{j+1} only. Figure 7 shows how instance I_{j+1} is constructed from instance I_j, including the virtual rectangle R_v^{j+1}.

We now proceed to analyze the integrality gap of instance I_j. First, it is clear that I_j contains $3j+2$ rectangles, and the size of the maximum clique is 2. Therefore, a fractional solution assigning LP-weight $\frac{1}{2}$ to each rectangle is a feasible solution of cost $(3j+2)/2$. We next show that the cost of the integral solution is at most $j+1$.

Claim C.1. *The optimal integral solution of I_j contains at most $j+1$ rectangles.*

Proof. We prove by induction on j. It is easy to verify that the optimal solution for I_1 contains 2 rectangles, and the optimal solution for I_2 contains 3 rectangles. Assume that maximum independent set for I_j contains $(j+1)$ rectangles and consider I_{j+1}. Let $\mathcal{S}$ be any solution for I_{j+1}. Notice that there are exactly three rectangles in I_{j+1} that do not appear in I_j: R_1^{j+1}, R_2^{j+1} and R_3^{j+1}. Therefore, if $\mathcal{S}$ contains at most one of these three rectangles, we can use the induction hypothesis to conclude that $|\mathcal{S}| \leq j+2$. Assume now that $\mathcal{S}$ contains at least two of these rectangles. The only way for this to happen is when both R_1^{j+1} and R_2^{j+1} are in $\mathcal{S}$. Then none of the rectangles $R_3^{j+1}, R_1^j, R_2^j, R_3^j$ may belong to $\mathcal{S}$. The size of $\mathcal{S} \setminus \left\{R_1^{j+1}, R_2^{j+1}\right\}$ is then bounded by optimal integral solution for I_{j-1}, and therefore by the induction hypothesis, $|\mathcal{S}| \leq j+2$.

It now follows that the integrality gap for I_n is $(3n+2)/2(n+1)$.

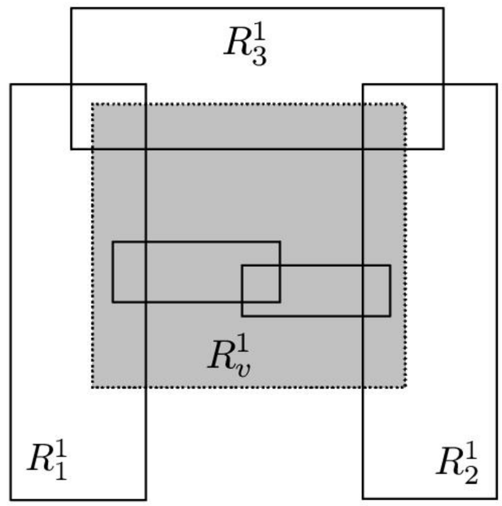

Figure 6: Instance I_1

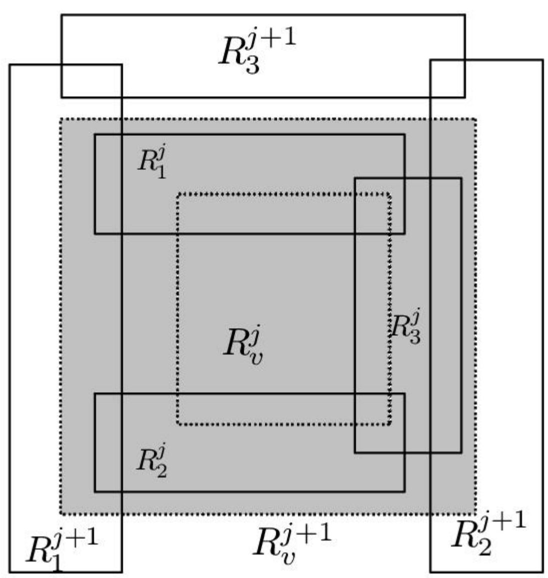

Figure 7: Instance I_{j+1}