Approximation Algorithms for Capacitated Rectangle Stabbing

Guy Even* Dror Rawitz† Shimon (Moni) Shahar*

May 24, 2006

Abstract

In the rectangle stabbing problem we are given a set of axis parallel rectangles and a set of horizontal and vertical lines, and our goal is to find a minimum size subset of lines that intersect all the rectangles. In this paper we study the capacitated version of this problem in which the input includes an integral capacity for each line. The capacity of a line bounds the number of rectangles that the line can cover. We consider two versions of this problem. In the first, one is allowed to use only a single copy of each line (*hard capacities*), and in the second, one is allowed to use multiple copies of every line provided that multiplicities are counted in the size of the solution (*soft capacities*).

For the case of d-dimensional rectangle stabbing with soft capacities, we present a $6d$-approximation algorithm and a 2-approximation algorithm when $d = 1$. For d-dimensional rectangle stabbing problem with hard capacities, we present a bi-criteria algorithm that computes $16d$-approximate solutions that use at most two copies of every line. For the one dimensional case, an 8-approximation algorithm for hard capacities is presented. Finally, we present hardness results for rectangle stabbing when the dimension is part of the input and for a two-dimensional weighted version with hard capacities.

Keywords: Approximation Algorithms; Rectangle Stabbing; Capacitated Covering.

1 Introduction

Understanding the combinatorial and algorithmic nature of capacitated covering problems is still an open problem. Only a few capacitated problems were studied including the general case of Set Cover [10] and the restricted case of Vertex Cover [2, 7]. Capacity constraints appear naturally in many applications, for example, bounded number of clients an antenna can serve. In this paper we consider a capacitated version of a covering problem, called rectangle stabbing. The geometric nature of the problem is used to obtain approximation algorithms.

The problems. The *rectangle stabbing* problem (RS) is a covering problem. Its uncapacitated version is defined as follows. The input is a finite set $\mathcal{U}$ of axis parallel rectangles and a finite set $\mathcal{S}$ of horizontal and vertical lines. A *cover* is a subset of $\mathcal{S}$ that intersects every rectangle in $\mathcal{U}$ at least once. The goal is to find a cover of minimum size. We denote the set of rectangles that a line $S \in \mathcal{S}$ intersects by $\mathcal{U}(S)$. Using this notation, an RS instance is simply a Set Cover instance in which the

*School of Electrical Engineering, Tel-Aviv University, Tel-Aviv, Israel. {guy, moni}@eng.tau.ac.il.

†Caesarea Rothschild Institute, University of Haifa, Haifa 31905, Israel. rawitz@cri.haifa.ac.il.

goal is to find a collection of subsets $\mathcal{U}(S)$, the union of which equals $\mathcal{U}$. Without loss of generality, one may assume that the RS instance is discrete in the following sense [5]: rectangle corners have integral coordinates and lines intersect the axes at integral points. In the one-dimensional version, the set $\mathcal{U}$ consists of horizontal interval and the set $\mathcal{S}$ consists of points. This is the well known polynomial clique cover problem in interval graphs.

The RS problem can be extended to d dimensions (d-RS). For $d \geq 3$, the set $\mathcal{U}$ consists of axis parallel d-dimensional rectangles (i.e., "boxes") and the set $\mathcal{S}$ consists of hyperplanes that are orthogonal to one of the d axes (i.e., "walls"). In the sequel we stick to the two-dimensional terminology, that is, we refer to $\mathcal{U}$ as a set of rectangles and to $\mathcal{S}$ as a set of lines.

In the *capacitated d-dimensional rectangle stabbing* problem the input includes an integral capacity $c(S)$ for every line $S \in \mathcal{S}$. The capacity $c(S)$ bounds the number of rectangles that S can cover. This means that in the capacitated case one has to specify which line covers each rectangle. The assignment of rectangles to lines may not assign more than $c(S)$ rectangles to a line S. We discuss two variants of capacitated d-dimensional rectangle stabbing called covering with hard capacities (HARD-d-RS) and covering with soft capacities (SOFT-d-RS).

A cover in SOFT-d-RS is formally defined as follows. The input consists of a set $\mathcal{U}$ of d-dimensional axis-parallel rectangles and a set $\mathcal{S}$ of lines (i.e., hyperplanes) that are orthogonal to one of the d axis. Each line $S \in \mathcal{S}$ is given a nonnegative integral capacity $c(S)$. An *assignment* is a function $A : \mathcal{S} \to 2^{\mathcal{U}}$ where $A(S) \subseteq \mathcal{U}(S)$, for every S. A rectangle u is *covered* by a line S if $u \in A(S)$. An assignment A is a *cover* if every rectangle is covered by some line, i.e., $\bigcup_{S \in \mathcal{S}} A(S) = \mathcal{U}$. The *multiplicity* (or number of copies) of a line $S \in \mathcal{S}$ in an assignment A equals $\lceil |A(S)|/c(S) \rceil$. We denote the multiplicity of S in A by $\alpha(A, S)$. The *size* of a cover A is the sum $\sum_{S \in \mathcal{S}} \alpha(A, S)$. We denote the size of A by $|A|$. The goal is to find a cover of minimum size.

Given the multiplicities of every line in a cover A, one can compute a cover with the same multiplicities by solving a flow problem. We therefore often refer to a cover simply as a multi-set of lines. The *support* of an assignment A is the set of lines $\{S \in \mathcal{S} : A(S) \neq \emptyset\}$. Note that the support is a set and not a multi-set. We denote the support of A by $\sigma(A)$.

In HARD-d-RS, a line may appear at most once in a cover. Hence, in this case, a cover is an assignment A for which $|A(S)| \leq c(S)$, (or $\alpha(A, S) \leq 1$) for every $S \in \mathcal{S}$. In this setting, we refer to a cover as the set of lines it contains (i.e., its support). Note that SOFT-d-RS is a special case of HARD-d-RS, since given a SOFT-d-RS instance one can always transform it into a HARD-d-RS instance by duplicating each line $|\mathcal{U}|$ times.

All the problems mentioned above have weighted versions, in which we are given a weight function w defined on the lines. In this case the cost of a cover A is $w(S) = \sum_S \alpha(S) \cdot w(S)$, and the goal is to find a cover of minimum weight.

Previous results. Since 1-RS is equivalent to clique cover in interval graphs, it can be solved in linear time [6]. Hassin and Megiddo [8] showed that RS is NP-hard, for $d \geq 2$. Gaur et al. [5] presented a d-approximation algorithm for d-RS that uses linear programming to reduce d dimensions to one dimension.

Capacitated covering problems (even with weights) date back to Wolsey [10] (see also [1, 2]). Wolsey presented a greedy algorithm for weighted Set Cover with hard capacities that achieves a logarithmic approximation ratio. Guha et al. [7] presented a 2-approximation primal-dual algorithm for the weighted Vertex Cover problem with soft capacities. Chuzhoy and Naor [2] presented a 3-approximation algorithm for Vertex Cover with hard capacities (without weights) which is based on randomized rounding with alterations. They also proved that the weighted version of this problem is as hard to approximate as set cover. Gandhi et al. [4] improved the approximation ratio for capacitated Vertex Cover to 2.

Our results. We present a 2-approximation algorithm for SOFT-1-RS. This algorithm is a dynamic programming algorithm that finds an optimal solution of a certain form. We also show that this algorithm extends to weighted SOFT-1-RS. We present $6d$-approximation algorithm for SOFT-d-RS, where d is arbitrary. This algorithm solves an LP relaxation of the problem, and rounds it using on the geometrical structure of the problem. For the case of hard capacities we show that the same technique can be used to obtain a bi-criteria algorithm for HARD-d-RS that computes solutions that are $16d$-approximate and use at most two copies of each line. An 8-approximation algorithm for the one dimensional case is also presented.

Finally, we present two hardness results. The first result mimics the hardness result given in [2], to show that weighted HARD-2-RS is Set Cover hard, even if all weights are in $\{0, 1\}$. The second hardness result proves that it is NP-hard to approximate d-RS with a ratio of $c \cdot \log d$, for some constant c. Note that the dimension d is considered here to be part of the input.

2 One dimensional rectangle stabbing with soft capacities

In this section we present a 2-approximation algorithm for SOFT-1-RS. In the one-dimensional case rectangles are simply intervals that we draw as horizontal intervals. To facilitate the task of drawing overlapping intervals, we separate intervals by drawing them at different heights. Hyperplanes in the one dimensional case are simply points. Since intervals are drawn as horizontal intervals with different heights, we refer to the hyperplanes as vertical lines instead of points. To summarize, the input in SOFT-1-RS consists of a set $\mathcal{U}$ of horizontal intervals and a set $\mathcal{S}$ of vertical lines with capacities $c(S)$.

The presentation of the approximation algorithm is divided into two parts. First, we define special covers, called *decisive covers*. We show that restricting the cover to be a decisive cover incurs a penalty that is bounded by a factor of two. Second, we present a dynamic programming algorithm that computes an optimal decisive cover.

2.1 Decisive Covers

Definition 1 *The total order $\prec$ is defined over the set $\mathcal{S}$ of vertical lines as follows: $S \prec S'$ if either (i) $c(S) > c(S')$ or (ii) $c(S) = c(S')$ and S is to the left of S'.*

The *support* of a cover A is the set of lines that participate in the cover, namely, $\{S \in \mathcal{S} : A(S) \neq \emptyset\}$. The support of a cover A is denoted by $\sigma(A)$. We sort the support of a cover A according to the $\prec$-order, namely, $\sigma(A) = \{S_1, S_2, \ldots, S_k\}$ where $S_1 \prec S_2 \prec \cdots \prec S_k$.

Recall that $\mathcal{U}(S)$ denotes the set of intervals intersected by the line S.

Definition 2 *A cover A is a* decisive cover *if $A(S_i) = \mathcal{U}(S_i) \setminus \bigcup_{j<i} \mathcal{U}(S_j)$, for every $1 \leq i \leq k$.*

Note that in a decisive cover each interval u is covered by the smallest (according the order $\prec$) line $S \in \sigma(A)$ that intersects u. Hence, "preference" is given to lines of higher capacity.

Given a cover A, the decisive cover A' *induced* by A is the cover obtained by assigning each interval u to the first line $S \in A$ that intersects it. Note that if A' is the decisive cover induced by a cover A, then $\sigma(A') \subseteq \sigma(A)$.

Claim 3 *The decisive cover A' induced by a cover A satisfies $|A'| \leq 2|A|$.*

Proof: We prove the slightly stronger inequality $|A'| \leq |A| + |\sigma(A)|$ using the following charging scheme.

Suppose that the purchasing power of a coupon is one copy of a vertical line. We say that a fractional distribution of coupons to intervals and lines is *valid* with respect to a cover $\tilde{A}$, if: (i) each line $S \in \sigma(\tilde{A})$ holds at least one coupon, and (ii) each interval $u \in \tilde{A}(S)$ holds at least $1/c(S)$ coupons.

Note that if a distribution of coupons is valid with respect to a cover $\tilde{A}$ then the number of coupons distributed to the intervals and lines is not less than the size of $\tilde{A}$. Indeed, if we consider each line $S \in \sigma(\tilde{A})$ separately, then the intervals together with S have at least $1 + |\tilde{A}(S)|/c(S) \geq \alpha(\tilde{A}, S)$ coupons.

We now consider the following distribution of coupons. Every line $S \in \sigma(A)$ gets one coupon and every interval $u \in A(S)$ gets $\alpha(A,S)/|A(S)|$ coupons. Note that (i) the number of coupons distributed to the intervals equals the size of A, (ii) the number of coupons distributed to the vertical lines equals the size of the support $\sigma(A)$.

To complete the proof, we show that this distribution of coupons is valid with respect to A'. Consider an interval u. The number of coupons given to u is $\alpha(A,S)/|A(S)| \geq 1/c(S)$. Let S' denote the line assigned to u in A', namely, $u \in A'(S')$. Since $S' \prec S$, it follows that $c(S') \geq c(S)$, and hence the number of coupons assigned to u is at least $1/c(S')$, as required. ■

We show that the previous claim is tight (see Fig. 1). The instance in Fig. 1 consists of $k+1$ vertical lines $\{S_1, \ldots, S_{k+1}\}$ and $k^2 + k$ horizontal intervals. The capacity of all the lines is k. The first k intervals (depicted in the bottom of the figure) intersect a single line; namely, the ith interval intersects only S_i, for every $i = 1, \ldots, k$. The remaining k^2 intervals are divided into k groups of size k. The intervals in the ith group intersect the sets $S_i, \ldots, S_{k+1}$ for $i = 1, \ldots, k$. Due to the first k intervals, every feasible solution must contain at least one copy of $S_1, \ldots, S_k$. This implies that there exists only one decisive cover, namely, the cover that contains two copies of S_i, for every $i = 1, \ldots, k$. The size of the decisive cover is $2k$. However, the optimal cover consists of a single copy of every line, and hence its size is $k+1$.

2.2 Dynamic programming

In this section we present a dynamic programming algorithm that finds an optimal decisive cover. According to Claim 3 this cover is 2-approximate.

We use the following notation. Given an interval u, we denote the coordinates of its endpoints by $\ell(u) < r(u)$. We assume, without loss of generality, that the coordinates are integers between 1 and $2|\mathcal{U}|$. Indeed, if two vertical lines intersect the same set of intervals, then we can unite them into one line by deleting the line with the smaller capacity. For every two integers $i < j$, let $\mathcal{U}(i,j)$ denote the set of intervals contained in the range $[i,j]$, namely, $\mathcal{U}(i,j) = \{u \in \mathcal{U} \mid i \leq \ell(u) < r(u) \leq j\}$. Also, let $\mathcal{S}(i,j,k)$ denote the set of vertical lines of capacity at most k whose x-coordinate is in the range $[i,j]$.

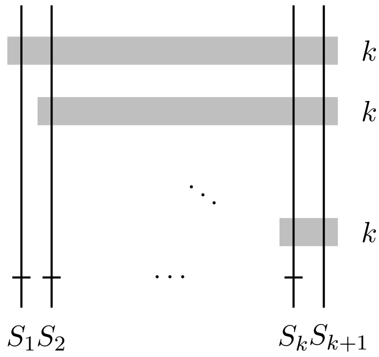


Figure 1: An example with a gap between an optimal decisive cover and an optimal cover for SOFT-1-RS. Each gray box represents k identical intervals.

The dynamic programming table Π of size $O(n^3)$ is defined as follows. The entry $\Pi(i, j, k)$ equals the size of an optimal decisive cover $A_{i,j,k}$ that covers the intervals in $\mathcal{U}(i, j)$ by lines from $\mathcal{S}(i, j, k)$. We initialize the table as follows:

- $\Pi(i, j, k) = 0$ if $\mathcal{U}(i, j) = \emptyset$.
- $\Pi(i, j, k) = \infty$ if there exist an interval $u \in \mathcal{U}(i, j)$ that is not intersected by lines in $\mathcal{S}(i, j, k)$.

The remaining table entries $\Pi(i, j, k)$ are calculated in polynomial time as follows. Let x_S denote the x-coordinate of a vertical line $S \in \mathcal{S}$. Let $\alpha(S, i, j)$ denote the number of copies of S required to cover all the intervals it intersects in $\mathcal{U}(i, j)$; namely, $\alpha(S, i, j) = \lceil |\{u \in \mathcal{U}(i, j) : \ell(u) \leq x_S \leq r(u)\}| / c(S) \rceil$.

The following recurrence is used:

$$\Pi(i, j, k) \leftarrow \min \left\{ \Pi(i, j, k-1), \min_{\substack{S \in \mathcal{S}(i,j,k) \\ c(S)=k}} \{\Pi(i, x_S - 1, k-1) + \alpha(S, i, j) + \Pi(x_S + 1, j, k)\} \right\}$$

Note that, in this recurrence, if $i = x_S$ then $\Pi(i, x_S - 1, k-1) = 0$. Similarly, if $x_S = j$ then $\Pi(x_S + 1, j, k) = 0$.

The justification for the recurrence is as follows. Consider two integers $i < j$. If $\Pi(i, j, k)$ is smaller than $\Pi(i, j, k-1)$, then the cover $A_{i,j,k}$ must contain a line of capacity k. Consider the leftmost line S of capacity k in $A_{i,j,k}$. Since $A_{i,j,k}$ is decisive, the line S must cover all the intervals that it intersects. Hence, $\alpha(S, i, j)$ copies of S are required. The remaining intervals are partitioned into intervals to the left of S and intervals to the right of S. The intervals in $\mathcal{U}(i, x_S - 1)$ are covered in $A_{i,j,k}$ by lines of capacity strictly less than k. The recurrence simply considers all possible lines of capacity k between i and j.

2.3 Weighted instances

The 2-approximation algorithm can be extended to weighted SOFT-1-RS as sketched below. In the weighted version, each vertical line S has a weight $w(S)$, and the weight of a cover A is

$\sum_{S\in\sigma(A)} w(S)\cdot\alpha(A,S)$. We define the *normalized weight* of S by $\tilde{w}(S) \triangleq w(S)/c(S)$. The total order $\prec$ is now defined by $S_1 \prec S_2$ if (i) $\tilde{w}(S_1) < \tilde{w}(S_2)$, or (ii) $\tilde{w}(S_1) = \tilde{w}(S_2)$ and $x_{S_1} < x_{S_2}$. (If two lines have the same normalized weight and the same x-coordinate, we may order them arbitrarily.) The claim analogous to Claim 3 is that the decisive cover A' induced by a cover A satisfies $w(A') \le w(A) + w(\sigma(A))$. The dynamic programming table has a state for every pair of integers (i,j) and normalized weight. (Note that we may have more than $2|\mathcal{U}|$ vertical lines in the weighted case since combinatorially equivalent lines might have different normalized weights.) Finally, the term $\alpha(S,i,j)$ in the dynamic programming recurrence should be multiplied by $w(S)$.

3 Fractional rectangle stabbing

In this section we present LP relaxations of d-dimensional rectangle stabbing with soft and hard capacities. We then show that the LP relaxations can be seen as network flow problems.

3.1 LP formulation

Following [2], we consider the linear programming relaxation for HARD-d-RS. To simplify notation we write $u \in S$ instead of $u \in \mathcal{U}(S)$.

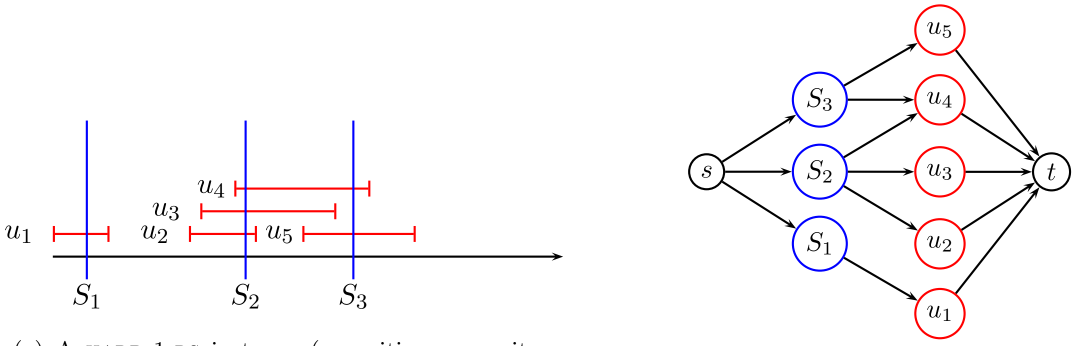
$$\begin{aligned}
&\min \sum_{S\in\mathcal{S}} x(S) && \\
&\text{s.t.} \sum_{S\,|\,u\in S} y(S,u) \ge 1 && \forall u \in \mathcal{U} \quad (1)\\
&\sum_{u\in S} y(S,u) \le c(S)x(S) && \forall S \in \mathcal{S} \quad (2)\\
&y(S,u) \le x(S) && \forall S,u \quad (3)\\
&x(S) \le 1 && \forall S \in \mathcal{S} \quad (4)\\
&x(S), y(S,u) \ge 0 && \forall S,u \quad (5)
\end{aligned}$$

We denote this LP by LP-HARD. The variable $x(S)$ indicates the "portion" of S that belongs to the cover. The variable $y(S,u)$ indicates the portion of u that is covered by S. Constraints of type (1) are simply covering constraints. Capacity constraints are formulated using constraints of types (2) and (3). Constraints of type (4) and type (5) are fractional relaxations of $x(S), y(S,u) \in \{0,1\}$. Note that there is a variable $y(S,u)$ only if $u \in S$. However, to simplify notation, we consider all pairs (S,u), regardless of whether $u \in S$. In case $u \notin S$, we simply assign $y(S,u) = 0$.

An LP-relaxation of SOFT-d-RS is obtained by omitting constraints of type (4). We denote the LP-relaxation without constraints of type (4) by LP-SOFT.

The integrality gap of both LP-HARD and LP-SOFT is at least $2-o(1)$ even in the one-dimensional case. Consider an instance that contains $k+1$ rectangles and two lines of capacity k that intersect all the rectangles. A fractional optimal solution is $x^*(S) = (k+1)/(2k)$ for each line S and $y^*(S,u) = 1/2$ for every line S and rectangle u. This means that the value of the fractional minimum is $1 + \frac{1}{k}$, while the integral optimum is 2.

The following definitions apply to both LP-HARD and LP-SOFT. We refer to a pair (x,y) as a *partial cover* if it satisfies all the constraints, except (perhaps) constraints of type (1). A rectangle



(a) A HARD-1-RS instance (capacities are omitted).

(b) The corresponding network.

Figure 2: A HARD-1-RS instance and the corresponding network.

is *covered* if its type (1) constraint is satisfied. If $\sum_{S \,|\, u \in S} y(S,u) \geq \alpha$, we refer to u as *α-covered*. If $\sum_{S \,|\, u \in S} y(S,u) > 0$ we say that u is *positively covered*.

We denote an optimal solution by (x^*, y^*). The sum $\sum_{S \in \mathcal{S}} x^*(S)$ is denoted by $\textsc{opt}^*$. Without loss of generality we assume that the covering constraints are tight, i.e., that $\sum_{S \,|\, u \in S} y^*(S,u) = 1$ for every $u \in \mathcal{U}$.

3.2 A network flow formulation

This section is written in HARD-d-RS terms, but similar arguments can be made in the case of SOFT-d-RS. It is very useful to view the LP relaxation as a network flow problem [1, 2]. Here we are given a (fractional) set of lines x and wish to find the best possible assignment y.

The network N_x is the standard construction used for bipartite graphs (see Fig. 2 for an example). On one side we have all the lines and on the other side we have all the rectangles. There is an arc (S,u) if $u \in S$. The capacity of an arc (S,u) equals $x(S)$. There is a source s that feeds all the lines. The capacity of each arc (s,S) emanating from the source equals $x(S) \cdot c(S)$. There is a sink t that is fed by all the rectangles. The capacity of every arc (u,t) entering the sink equals 1.

Observation 4 *There is a one-to-one correspondence between vectors y such that (x,y) is a partial cover and flows f in N_x. The correspondence $y \leftrightarrow f_y$ satisfies $f_y(u,t) = \sum_{S|u \in S} y(S,u)$, for every rectangle $u \in \mathcal{U}$, and $f_y(s,S) = \sum_{u \in S} y(S,u)$, for every line $S \in \mathcal{S}$.*

Proof: Given y simply define f_y as follows.

$$f_y(e) \triangleq \begin{cases} \sum_{u \in S} y(S,u) & \text{if } e = (s,S), \\ y(S,u) & \text{if } e = (S,u), \\ \sum_{S \,|\, u \in S} y(S,u) & \text{if } e = (u,t). \end{cases}$$

The mapping from flows to vectors is defined similarly. ■

We often refer to $f_y(s,S)$ as the *flow supplied by S* and to $f_y(u,t)$ as the *flow delivered to u*. To simplify notation, we denote $f_y(s,S)$ by $f_y(S)$ and $f_y(u,t)$ by $f_y(u)$. We say that y is *maximum with respect to x* if f_y is a maximum flow in N_x.

Next, we show that we can identify infeasible instances of HARD-d-RS.

Observation 5 *Feasibility of a* HARD-d-RS *instance can be verified by computing a maximum integral flow in a network N_x, where $x(S) = 1$, for every $S \in \mathcal{S}$.*

The following observation is based on the integrality of a max-flow in a network with integral capacities. It implies that it suffices to compute a feasible cover (x, y), where x is integral.

Observation 6 ([2]) *Let (x, y) be a feasible solution of* LP-HARD. *If x is integral, then an integral y' such that (x, y') is a feasible solution can be computed in polynomial time.*

Definition 7 *Let (x, y) and (x, y') be partial covers. We say that y'* dominates *y if (i) $f_{y'}(u) \geq f_y(u)$, for every $u \in \mathcal{U}$, and (ii) $f_{y'}(S) \geq f_y(S)$, for every $S \in \mathcal{S}$. We write $y' \succeq y$ to denote that y' dominates y.*

Observation 8 *Let (x, y) denote a partial cover. Then one can find in polynomial time a maximum vector y' with respect to x that also dominates y.*

Proof: We use an augmenting path algorithm to compute a maximum flow f' in N_x starting with f_y. The flow f' induces the desired vector $y' \succeq y$ since saturating an augmenting path from s to t never decreases the flow in edges exiting s, or in edges entering t. ■

Let **aug-flow** be an efficient algorithm that given a partial cover (x, y), finds a vector $y' \succeq y$ that is maximum with respect to x. Note that **aug-flow** may change the assignment of lines to rectangles. In terms of the network flow, the flow of certain edges may decrease, but the sum of flows that enters (exits, respectively) every rectangle (line, respectively) does not decrease.

4 Rectangle stabbing with soft capacities

In this section we present a $6d$-approximation algorithm for SOFT-d-RS. The algorithm is based on solving LP-SOFT, and then rounding the solution. For the sake of simplicity, the algorithm is presented for the 2-dimensional case ($d = 2$).

Let $\varepsilon = 1/6d$ and let (x^*, y^*) be an optimal solution of LP-SOFT. We define

$$H \triangleq \{S \mid x^*(S) \geq \varepsilon\} \qquad \text{and} \qquad L \triangleq \{S \mid x^*(S) < \varepsilon\} \ .$$

Let $L = L^h \cup L^v$ denote a partition of L into horizontal and vertical lines. We partition the horizontal line in L^h into "contiguous blocks" by accumulating lines in L^h from "left" to "right" until the sum of fractional values $x(S)$ in the block exceeds ε. We denote the blocks by $L^h_1, L^h_2, \ldots, L^h_{b(h)}$ and the (possibly empty) leftover block by $\tilde{L}^h$. By the construction,

$$\forall j \leq b(h) \ : \ \varepsilon \leq \sum_{S \in L^h_j} x^*(S) < 2\varepsilon$$
$$\sum_{S \in \tilde{L}^h} x^*(S) < \varepsilon.$$

The same type of partitioning is applied to the vertical lines in L^v to obtain the blocks $L^v_1, \ldots, L^v_{b(v)}$ and the leftover block $\tilde{L}^v$.

Observation 9 *The number of blocks (not including the leftover block) in each dimension satisfies* $b(h) \le \frac{1}{\varepsilon} \cdot \sum_{S \in L^h} x^*(S)$ *and* $b(v) \le \frac{1}{\varepsilon} \cdot \sum_{S \in L^v} x^*(S)$.

Let $S^*_{h,j}$ and $S^*_{v,j}$ denote lines of maximum capacity in L^h_j and L^v_j, respectively. Let

$$L^* \triangleq \left\{ S^*_{h,j} \,|\, 1 \le j \le b(h) \right\} \cup \left\{ S^*_{v,j} \,|\, 1 \le j \le b(v) \right\}.$$

Definition 10 *We define the partial cover* (x, y) *as follows. The support of the cover is* $H \cup L^*$. *For every* $S \in H$ *and* $u \in \mathcal{U}(S)$, *we keep* $x(S) = x^*(S)$ *and* $y(S, u) = y^*(S, u)$. *For every* $S \in L^*$ *and* $u \in \mathcal{U}(S)$, *let* $B(S)$ *denote the block that contains* S. *Then,*

$$x(S) = \sum_{S' \in B(S)} x^*(S') \qquad \text{and} \qquad y(S, u) = \sum_{S' \in B(S)} y^*(S', u).$$

Note that if $S = S^*_{h,j}$ and $u \in S^*_{h,j}$, then $y(S^*_{h,j}, u)$ covers u to the same extent that u is covered by lines in L^h_j according to y^*. Hence, rectangles that are intersected by $S^*_{h,j}$ are "locally satisfied". Also notice that $\sum_S x(S) = \sum_S x^*(S)$. We now prove that (x, y) is a indeed partial cover.

Claim 11 (x, y) *is a partial cover.*

Proof: We first show that constraints of type (3) are satisfied, namely, that $y(S, u) \le x(S)$, for every u, S. Clearly, this is true for $S \notin L^*$. Consider a line $S^* \in L^*$. Let B denote the block of lines in L that contains S^*. For every rectangle u intersected by S^*, the following holds:

$$y(S^*, u) = \sum_{S' \in B} y^*(S', u) \le \sum_{S' \in B} x^*(S') = x(S^*).$$

We now show that constraints of type (2) are satisfied, namely, that $\sum_{u \in S} y(u, S) \le c(S)x(S)$, for every $S \in \mathcal{S}$. This trivially holds for $S \notin H \cup L^*$ since both $x(S) = 0$, and $y(S, u) = 0$. Constraint (2) holds for $S \in H$, since $x(S) = x^*(S)$, and $y(S, u) = y^*(S, u)$. It remains to consider lines in $S^* \in L^*$. Let B denote the block of lines in L that contains S^*.

$$\begin{aligned}
\sum_{u \in S^*} y(S^*, u) &= \sum_{u \in S^*} \sum_{S \in B} y^*(S, u) \\
&\le \sum_{S \in B} \sum_{u \in S} y^*(S, u) \\
&\le \sum_{S \in B} c(S) x^*(S) \\
&\le \max_{S \in B} c(S) \cdot \sum_{S \in B} x^*(S) \\
&= c(S^*) \cdot x(S^*) \ .
\end{aligned}$$

The first inequality follows from the fact that some rectangles may lose part of their flow, the second inequality is due to the LP constraints, and the third inequality follows from Def. 10. ■

Claim 12 *The coverage of every rectangle* u *is greater than* $(1 - 4d\varepsilon)$ *in the partial cover* (x, y).

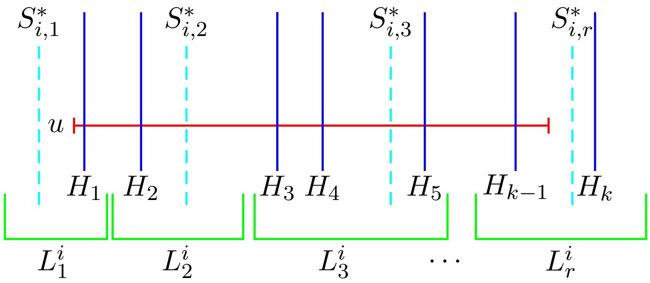


Figure 3: An interval u covered by both lines in H and by lines in L. S^*_1 and S^*_r do not intersect u, and therefore, the flow supplied to u by f_{y^*} in the blocks L_1 and L_r is lost.

Proof: Consider a rectangle u. We show that, in each dimension, the coverage of u decreases by less than 4ε due to the transition from y^* to y. By definition, coverage by lines in H is preserved. In addition, if a rectangle u intersects all the lines in a block L^h_j, then the coverage of u by lines in L^h_j is now covered by $S^*_{h,j}$. Namely, $\sum_{S\in L^h_j} y^*(S,u) = y(S^*_{h,j},u)$. It follows that u may lose coverage only in the "leftmost" and "rightmost" blocks that u intersects. In each such block, the coverage of u is bounded by 2ε, (a one dimensional example is given in Fig. 3). Since u is covered in (x^*, y^*), it follows that $\sum_S y(S,u) > 1 - d \cdot 4\varepsilon$, and the claim follows. ■

Since $\varepsilon = 1/6d$, by Claim 12 we get that each rectangle is 1/3-covered by (x, y). A cover is obtained by scaling as follows. Let $x'(S) = \lceil 3x(S) \rceil$ for every $S \in \mathcal{S}$, and $y'(u) = 3y(u)$ for every $u \in \mathcal{U}$. Clearly, every rectangle is covered by (x', y'). Moreover, by Obs. 6 an integral y'' such that (x', y'') is a cover can by computed in polynomial time.

It remains to show that (x', y'') is a $6d$-approximation. It suffices to show that $x'(S) \leq 6d \cdot x(S)$, for every $S \in H \cup L^*$. If $x(S) \geq 1/3$ then, $x'(S) \leq 3x(S) + 1 \leq 6x(S)$. If $x(S) < 1/3$, then $x'(S) = 1$ and $x(S) \geq \varepsilon$ for every line $S \in H \cup L^*$. Therefore $x'(S) = 1 = 6d\varepsilon \leq 6d \cdot x(S)$, as required.

5 Rectangle stabbing with hard capacities

We present a bi-criteria approximation algorithm for HARD-d-RS that computes $16d$-approximate cover that uses at most two copies of each line. The algorithm is similar to the $6d$-approximation algorithm for SOFT-d-RS. We first computed an optimal solution for LP-HARD. Afterwards, we set $\varepsilon = \frac{1}{8d}$ and compute H and L^* using the same algorithm defined in the previous section. Finally, we take two copies of each line in $H \cup L^*$ and use flow to compute an integral cover.

We first show that this a cover. The rounding of the LP-solution yields a $(1 - 4d\varepsilon)$-cover according to Claim 12. We obtain a 1/2-cover by setting $\varepsilon = \frac{1}{8d}$. Note that $x(S) \leq 1$, for every line S, hence $\lceil 2 \cdot x(S) \rceil \leq 2$. Note that we rely on Obs. 5 to insure that there is an integral cover using these two copies of each line in the support of x.

The approximation ratio of $16d$ is proved as follows. Note that $x(S) > 0$ only if $S \in H \cup L^*$. Since we take two copies of lines in $H \cup L^*$, it suffices to prove that $|H \cup L^*| \leq 8d \cdot \sum_{S\in\mathcal{S}} x^*(S)$. Clearly,

$$|H| \leq \frac{1}{\varepsilon} \cdot \sum_{S\in H} x^*(S).$$

Due to the bound on the number of blocks (Obs. 9) we obtain,

$$|L^*| \le \frac{1}{\varepsilon} \cdot \sum_{S \in L} x^*(S).$$

It follows that $|H \cup L^*| \le \frac{1}{\varepsilon} \cdot \sum_{S \in \mathcal{S}} x^*(S)$, as required.

6 One dimensional rectangle stabbing with hard capacities

In this section we present an 8-approximation algorithm for HARD-1-RS. The algorithm augments the positive cover obtained by Claim 12 with $\varepsilon = 1/4$. A local greedy rule is used to select the line to be added to the partial cover.

6.1 Thirsty lines and dams

Throughout this section we consider a partial cover (x, y) such that x is integral and y is maximum with respect to x. The following definition considers two types of lines in x.

Definition 13 *A line $S \in x$ is a* dam *with respect to (x, y) if y remains maximum with respect to x even if the capacity $c(S)$ is (arbitrarily) increased. Otherwise, S is* thirsty *with respect to (x, y).*

Note that if S is not saturated (i.e., $f_y(S) < x(S) \cdot c(S)$), then obviously S is not thirsty, so S is a dam. However, S may be saturated (i.e., $f_y(S) = c(S)$) and yet not thirsty. Such a case is easily described using the network flow formalism: the arc does not belong to every min-cut in N_x.

Lemma 14 *Suppose that $S \in x$ and S is a dam. Then: (1) every interval $u \in S$ is covered (i.e., $f_y(u) = 1$), and (2) if $u \in S$ and $y(S', u) > 0$, then S' is also a dam.*

Proof: Proof of (1). If u is not covered, then an increase in $c(S)$ can be used to increase $y(S, u)$, contradicting the assumption that S is a dam.

Proof of (2). Since x is integral and $y(S', u) > 0$, it follows that $S' \in x$. We show that if S' is thirsty, then S is also thirsty. Loosely speaking, we show that increasing $c(S)$ enables an increase in the flow, since $y(S', u)$ can be decreased and this "released" flow can be used to "serve" another interval. An illustration of this case is given in Fig. 4.

We show this formally by presenting an augmenting path in the residual graph of N_x after the capacity of S is increased. Let p denote an augmenting path in N_x obtained when $c(S')$ is increased (p exists since we assume that S' is thirsty). Obviously, the first arc in p is (s, S').

Observe that the three arcs $(s, S), (S, u)$, and (u, S') are in the residual graph of N_x after $c(S)$ is increased. This follows since: (i) $f_y(S)$ is less than the increased capacity of S, (ii) $f_y(S, u) \le 1 - y(S', u) < 1 = x(S)$, and (iii) $y(S', u) > 0$. Thus the path $s \to S \to u \to S'$ concatenated with $p \setminus (s, S')$ is an augmenting path in the residual of N_x after the capacity of S is increased, as required. ■

The following corollary is directly implied by Lemma 14.

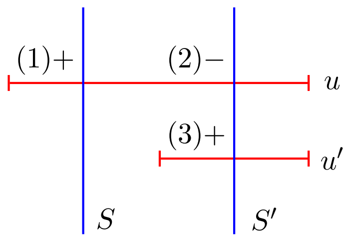


Figure 4: Illustration of Lemma 14: Line S is a dam and line S' partly covers u. We show that S' is also a dam. If S' is thirsty, then an increase in $c(S)$ increases the flow (contradicting the assumption that S is a dam). Changes in flow are depicted by plus and minus signs. (1) An increase in the capacity $c(S)$ enables increasing $y(S,u)$. (2) $y(S',u)$ may be decreased, since $y(S,u)$ is increased. (3) If S' is thirsty and can serve an interval u', then $y(S',u')$ can be increased.

Corollary 15 (Decomposition) *Define:*

$$D \triangleq \{S \in x \mid S \text{ is a dam}\}$$
$$\mathcal{U}_D \triangleq \{u \in \mathcal{U} \mid \exists S \in D \text{ such that } u \in S\}.$$

Then, for every $u \in \mathcal{U}_D$,

$$\sum_{S \in D} y(S,u) = 1.$$

The following corollary shows that if no thirsty lines exist in a positive partial cover, then the cover is feasible.

Corollary 16 *Let (x,y) be a partial cover such that x is integral and y is maximum with respect to x. If every interval is positively covered and no line is thirsty, then (x,y) is a feasible cover.*

Proof: Since every interval is positively covered and there are no thirsty lines, it follows that $\mathcal{U}_D = \mathcal{U}$, and by Coro. 15, every rectangle is covered. ■

6.2 Decomposition into strips

Let (x,y) be a partial cover, where x is integral and y is maximum with respect to x. Consider two consecutive dams S_1 and S_2 (i.e., there is no dam between S_1 and S_2). The subproblem induced by S_1 and S_2 consists of the following lines and intervals: (i) the vertical lines that are strictly between S_1 and S_2 and (ii) the intervals that are contained in the open strip, the boundaries of which are S_1 and S_2. We refer to the subproblem induced by two consecutive dams as a *strip* and denote it by $B = (\mathcal{S}^B, \mathcal{U}^B)$. Note that extreme dams induce marginal strips that are bounded just from one side.

Definition 17 *The* residual capacity *of a line $S \in \mathcal{S}^B$ in a strip $B = (\mathcal{S}^B, \mathcal{U}^B)$ is defined by*

$$c^B(S) = \min\{c(S), |S \cap \mathcal{U}^B|\}.$$

Definition 18 *Let $B = (\mathcal{S}^B, \mathcal{U}^B)$ denote a strip with respect to (x, y).*

- *The flow supplied by f_y to strip B is defined by*

$$f_y(B) \triangleq \sum_{S \in \mathcal{S}^B} \sum_{u \in \mathcal{U}^B} y(S, u).$$

- *The* deficit *in strip B of a partial cover (x, y) is defined by*

$$\Delta_y(B) \triangleq |\mathcal{U}^B| - f_y(B).$$

- *A strip B is called* active *if $\Delta_y(B) > 0$.*

Let $\{B_i\}_{i \in I}$ denote the set of strips induced by the dams corresponding to (x, y). The following observation uses a "flooding" argument to show that feasibility follows from lack of active strips.

Observation 19 *$\Delta_y(B_i) \leq 0$, for every $i \in I$, if and only if (x, y) is a feasible cover.*

6.3 The approximation algorithm

The approximation algorithm for HARD-1-RS begins like the bi-criteria approximation algorithm and then applies a new augmentation procedure, called **make-feasible**.

The algorithm proceeds as follows: (i) Solve the the linear program LP-HARD. (ii) Set $\varepsilon = 1/4$. Fix x_0 to be the indicator function of the set $H \cup L^*$. Fix y_0 to be the aggregation of coverage of the LP solution as described in Def. 10. (iii) Apply **aug-flow**(x_0, y_0) to compute a maximum flow y'_0 with respect to x_0 that dominates y_0. (iv) Run **make-feasible**(x_0, y'_0) to obtain a cover (x^I, y^F) in which x^I is integral but y^F is fractional. (v) Obtain an integral cover (x^I, y^I) using a maximum flow algorithm (Obs. 8). The remainder of this section deals with description of the **make-feasible** Algorithm.

Algorithm **make-feasible** iteratively augments the partial cover until a cover (x, y) is obtained. By Obs. 19, Algorithm **make-feasible** stops adding lines to the partial cover when there are no active strips. Otherwise, a new line is added to the cover as follows: (i) pick an active strip B and a line $S_{\max}$ with the largest residual capacity among the lines in $\mathcal{S}^B$ that have not been added yet to the partial cover, (ii) add the line $S_{\max}$ to the partial cover x to obtain x', and (iii) find a maximum flow $y' \succeq y$ with respect to x; by calling **aug-flow**(x', y). The algorithm then repeats with the partial cover (x', y'). A recursive description of the algorithm is listed as Algorithm 1 (this is tail recursion, so it is identical to a loop).

Throughout the execution of Algorithm **make-feasible** (including the recursive calls) two invariants are satisfied: (i) The x-component of the partial cover is integral, namely, it is the indicator function of a subset of lines. To simplify notation, we treat the x-component as the subset itself. So $x' \leftarrow x \cup \{S\}$ means that x' is the indicator function of the subset corresponding to x together with $\{S\}$. (ii) Every interval is positively covered by the partial cover (x, y).

Both invariants hold initially with respect to (x_0, y'_0). Integrality of x is kept since a new line is added each time. Positive coverage is kept since flow is augmented.

First, we show that Algorithm **make-feasible** finds a feasible cover if one exists. Observe that as long as there is an active strip, we add a line $S_{\max}$ to x. As soon as every strip is not active, the cover is feasible by Obs. 19. Hence, it remains to prove that $S_{\max}$ is well defined.

Algorithm 1 - make-feasible($\mathcal{S}$,$\mathcal{U}, x, y$) : Augment a partial cover (x, y) to a feasible cover. We assume that x is integral and y that is maximum with respect to x.

1: Stopping condition: If (x, y) is feasible then **Return**(x, y).
2: Let $B = (\mathcal{S}^B,\mathcal{U}^B)$ denote an active strip with respect to (x, y).
3: Find a max-residual-capacity line (where $c^B(S) \triangleq \min\{c(S), |S \cap \mathcal{U}^B|\}$).

$$S_{\max} \leftarrow \operatorname{argmax}\{c^B(S) : S \in \mathcal{S}^B \setminus x\}.$$

4: Add $S_{\max}$ to x:

$$x' \leftarrow x \cup \{S_{\max}\}.$$

5: Augment flow:

$$y' \leftarrow \textbf{aug-flow}(x', y).$$

6: Recurse: **Return make-feasible**($\mathcal{S}$,$\mathcal{U}, x', y'$).

Claim 20 *If $B = (\mathcal{S}^B,\mathcal{U}^B)$ is an active strip, then $\mathcal{S}^B \setminus x \neq \emptyset$.*

Proof: We assume that the problem is feasible. Hence $(\mathcal{S}, y^*)$ is a feasible cover and $(\mathcal{S}^B, y^*_B)$ is a feasible cover of B, where y^*_B is the restriction of y^* to $\mathcal{S}^B \times \mathcal{U}^B$. Assume for the sake of contradiction that $\mathcal{S}^B \subseteq x$. Since y is maximum with respect to x, it follows that (x, y) is feasible in B, which means by Obs. 19 that B is not active, a contradiction. ■

Algorithm **make-feasible** runs in polynomial time, since there are at most $|\mathcal{S}|$ recursive calls, and the running time of each recursive call is polynomial by Obs. 8.

6.4 Approximation ratio

In this section we prove the approximation ratio (recall that $\varepsilon = 1/4$).

Theorem 21 *The approximation ratio of the algorithm for* HARD-1-RS *is* $\frac{2}{\varepsilon} = 8$.

The main idea in the proof is that each line added by Algorithm **make-feasible** to the partial cover becomes a dam together with at least one of the original thirsty lines. Since there are no more than $\frac{1}{\varepsilon}\text{OPT}^*$ lines in $H \cup L^*$, we reach a total of at most $\frac{2}{\varepsilon} \cdot \text{OPT}^* = 8 \cdot \text{OPT}^*$ lines.

We begin by showing that the capacity of a line $S^* \in L^*$ is at least two times bigger than the flow delivered by f_{y^*} to intervals intersected by lines in the block of S^*. We use this observation to argue that if a line $S^* \in L^*$ is saturated then the initial partial cover (x_0, y'_0) is "locally competitive".

Observation 22 *Let L_j be the block that contains a line $S^* \in L^*$. Then, $\sum_{S \in L_j} f_{y^*}(S) < 2\varepsilon \cdot c(S^*)$.*

Proof:

$$\sum_{S \in L_j} f_{y^*}(S) \leq \sum_{S \in L_j} c(S)x^*(S) \leq \max_{S \in L_j} c(S) \cdot \sum_{S \in L_j} x^*(S) < 2\varepsilon \cdot c(S^*).$$

■

Consider a recursive call of Algorithm **make-feasible**. Consider the active strip $B = (\mathcal{S}^B, \mathcal{U}^B)$ chosen in Line 2 and the line $S_{\max} \in \mathcal{S}^{\mathcal{B}}$ chosen in Line 3. The following claim bounds the deficit in B.

Claim 23 $\Delta_y(B) < c^B(S_{\max})$.

Proof: Let $L_\ell, \ldots, L_r$ be the blocks that are contained in the strip B, and let S_i^* denote the line of maximum capacity selected in L_i. In addition, there could be marginal blocks, $L_{\ell-1}$ and L_{r+1}, that are partly contained in B. We show that the deficit of B is "concentrated" in these marginal blocks. Since every line $S \in x \cap \mathcal{S}^B$ is thirsty, it is also saturated. Hence, $f_y(S) = c(S)$. By Obs. 22, it follows that $f_y(S_i^*) > \sum_{S \in L_i} f_{y^*}(S)$ for every $i \in \{\ell, \ldots, r\}$. Hence, the only possible loss of flow in strip B may be due to the extreme blocks $L_{\ell-1}$ and L_{r+1}.

We now focus on the parts of the extreme blocks that are contained in the strip B, namely, $L_{\ell-1} \cap \mathcal{S}^B$ and $L_{r+1} \cap \mathcal{S}^B$. We denote these parts by L' and L''. Note that Obs. 22 does not suffice, since we wish to bound the deficit by the residual capacity.
The fraction flow y^* in B satisfies:

$$f_{y^*}(B) = \sum_{S \in \mathcal{S}^B} \sum_{u \in \mathcal{U}^B} y^*(S, u) \leq \sum_{S \in \mathcal{S}^B} x^*(S) \cdot c^B(S).$$

On the other hand, the flow y satisfies:

$$f_y(B) \geq f_y(L' \cup L'') + \sum_{i=\ell}^{r} c(S_i^*) \geq f_y(L' \cup L'') + \sum_{S \in L_\ell \cup \cdots \cup L_r} x^*(S) \cdot c^B(S).$$

It follows that the deficit is bounded by

$$\Delta_y(B) \leq \sum_{S \in L' \cup L''} x^*(S) \cdot c^B(S) - f_y(L' \cup L'') < 4\varepsilon \cdot c^B(S_{\max}).$$

The justification for the last inequality is as follows. Consider each extreme block L' and L'' separately. If $c^B(S) \leq c^B(S_{\max})$, for every line in $S \in L'$, then clearly $\sum_{S \in L'} x^*(S) \cdot c^B(S) < 2\varepsilon \cdot c^B(S_{\max})$. Indeed, this holds if $f_y(L') = 0$. However, if $f_y(L') > 0$, we cannot rule out (at this stage) the possibility that a line $S' \in L'$ has been added to x, and hence $c^B(S') > c^B(S_{\max})$.

If $f_y(L') > 0$, let S' denote the first line in L' that was added to x. Since the residual capacity does not increase and since S' is saturated, it follows that $f_y(S') > \sum_{S \in L'} x^*(S) \cdot c^B(S)$. The same holds for L''. The claim follows since $\varepsilon = 1/4$. ■

Observation 24 *Every active strip contains at least one thirsty line.*

Proof: Each interval is positively covered throughout Algorithm **make-feasible**. By Lemma 14, if intersected by a dam, an interval is covered. Hence, a non-covered interval in an active strip is positively covered by a thirsty line. ■

Claim 25 *$S_{\max}$ is a dam with respect to (x', y'), and there exists a line $S \in H \cup L^*$ that is thirsty with respect to (x, y) and is a dam with respect to (x', y').*

Proof: By Claim 23, $\Delta_y(B) < c^B(S_{\max})$, hence $S_{\max}$ cannot be saturated, and must be a dam. Since $S_{\max}$ is not saturated, there exists an interval $u \in S_{\max} \cap \mathcal{U}^B$ such that $y'(S_{\max}, u) < 1$. By Lemma 14, since $S_{\max}$ is a dam, it follows that $f_{y'}(u) = 1$. Hence, there exists a line $S \in H \cup L^*$ such that $y'(S, u) > 0$. By Lemma 14, S is also a dam with respect to (x', y'). On the other hand, since S is in the strip B it must be thirsty with respect to (x, y), as required. ■

By Claim 25 it follows that each addition of a line transforms at least one of the original thirsty lines into a dam. Since no new thirsty lines are introduced, as soon as the original thirsty lines are exhausted, the algorithm stops. It follows that the size of the solution is bounded by $\frac{2}{\varepsilon} \cdot \text{OPT}^*$, and Theorem 21 follows.

7 Hardness results

In this section we present two hardness results. Both results rely on the fact that Set Cover cannot be approximated within a factor of $c \log n$ for some $c > 0$, unless P=NP [3, 9]. The first reduction shows that the HARD-d-RS problem is Set Cover hard, for $d \geq 2$, if weights are given to the lines. The second reduction shows that it is NP-hard to approximate d-RS within $c \log d$, for some constant c when the dimension d is part of the input.

7.1 Weighted rectangle stabbing with hard capacities

We show that for $d \geq 2$ the weighted capacitated rectangle stabbing is as hard to approximate as Set Cover. Chuzhoy and Naor [2] presented a reduction of Set Cover to weighted capacitated Vertex Cover in bipartite graphs. Our reduction mimics their reduction by applying it to the adjacency matrix of bipartite graphs (as in the Egerváry-König Theorem).

Consider the Set Cover instance with the collection $\mathcal{C} = \{C_1, \ldots, C_m\}$ of subsets of $\{1, \ldots, n\}$. We construct the two-dimensional instance $(\mathcal{S}, \mathcal{U})$ of weighted capacitated rectangle stabbing as follows. The rows, indexed $1, \ldots, n$ correspond the elements, and the columns, indexed $1, \ldots, m$ correspond to the subsets. The set of rectangles $\mathcal{U}$ contains a unit (one by one) square u_{ij} for every pair (i, C_j) such that $i \in C_j$. The coordinates of the center of u_{ij} are (i, j). The vertical lines in $\mathcal{S}$ are $(x = j)$, for $1 \leq j \leq m$. Each vertical line is assigned unit weight and capacity n. The horizontal lines in $\mathcal{S}$ are the lines $(y = i)$, for $1 \leq i \leq n$. The weight of every horizontal line is zero, and the capacity of the line $(y = i)$ is $|\{C_j \mid i \in C_j\}| - 1$.

Given a solution $\mathcal{C}' \subseteq \mathcal{C}$ to the Set Cover instance, the corresponding solution to the rectangle stabbing instance consists of the vertical lines $(x = j)$, where $C_j \in \mathcal{C}'$ and all the horizontal lines. Given a solution $\mathcal{S}' \subseteq \mathcal{S}$ to the rectangle stabbing instance, it is easy to see that $\{C_j \mid (x = j) \in \mathcal{S}'\}$ is a Set Cover. This completes the approximation preserving reduction.

7.2 Rectangle stabbing with dimension d

We present an approximation preserving reduction from Set Cover to Rectangle Stabbing. The dimension of the reduced instance equals the number of sets in the Set Cover instance. Therefore, it is NP-hard to approximate d-RS within $c \log d$, for some constant c.

Consider the Set Cover instance with the collection $\mathcal{C} = \{C_1, \ldots, C_m\}$ of subsets of $\{1, \ldots, n\}$. We construct the following instance of m-dimensional rectangle stabbing. For every $1 \leq j \leq n$, let $\chi_j \in \mathbb{R}^m$ be a vector that indicates which subsets in $\mathcal{C}$ contain j. Formally, $\chi_j = (\chi_{j1}, \ldots, \chi_{jm})$, where $\chi_{ji} = 1$ if $j \in S_i$, and zero otherwise. For every $1 \leq j \leq m$, let u_j denote the rectangle whose opposite corners are χ_j and $(-1, \ldots, -1)$. Each set C_i is represented by the hyperplane $x_i = 1/2$ (i.e., the ith coordinate equals 1/2). Since a rectangle u_j intersects a hyperplane ($x_i = 1/2$) if and only if $j \in C_i$, the reduction follows.

8 Open problems and remarks

We list a few open problems. The hardness of the one-dimensional capacitated rectangle stabbing (soft or hard) is open. An $O(d)$-approximation algorithm for HARD-d-RS is also open, as well as an $O(d)$-approximation algorithm for weighted SOFT-d-RS. We show that weighted HARD-d-RS is set-cover hard.

Gaur et al. [5] presented a d-approximation algorithm for weighted d-RS that uses linear programming to reduce the problem into d one-dimensional instances. The analysis of their algorithm relies on the integrality of the LP relaxation in the one dimensional case. Our 2-approximation for weighted SOFT-1-RS does not prove a bound on the integrality gap. Our 6-approximation algorithm for unweighted SOFT-1-RS proves that integrality gap of the one-dimensional case is bounded by 6. Hence another $6d$-approximation algorithm follows by combining a reduction similar to the Gaur et al. [5] and our 6-approximation algorithm for SOFT-1-RS.

Acknowledgment

We thank Alexander Ageev for pointing out an error in an earlier version of the paper.

References

[1] J. Bar-Ilan, G. Kortsarz, and D. Peleg. Generalized submodular cover problems and applications. *Theoretical Computer Science*, 250:179–200, 2001.

[2] J. Chuzhoy and J. Naor. Covering problems with hard capacities. In *43nd IEEE Symposium on Foundations of Computer Science*, pages 481–489, 2002.

[3] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[4] R. Gandhi, E. Halperin, S. Khuller, G. Kortsarz, and A. Srinivasan. An improved approximation algorithm for vertex cover with hard capacities. In *30th Annual International Colloquium on Automata, Languages and Programming*, volume 2719 of *LNCS*, pages 164–175, 2003.

[5] D. R. Gaur, T. Ibaraki, and R. Krishnamurti. Constant ratio approximation algorithms for the rectangle stabbing problem and the rectilinear partitioning problem. *Journal of Algorithms*, 43:138–152, 2002.

[6] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York, 1980.

[7] S. Guha, R. Hassin, S. Khuller, and E. Or. Capacitated vertex covering. *Journal of Algorithms*, 48(1):257–270, 2003.

[8] R. Hassin and N. Megiddo. Approximation algorithms for hitting objects with straight lines. *Discrete Applied Mathematics*, 30(1):29–42, 1991.

[9] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *29th ACM Symposium on the Theory of Computing*, pages 475–484, 1997.

[10] L. A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2:385–393, 1982.