Approximation Algorithms for Capacitated Rectangle Stabbing

Guy Even[1], Dror Rawitz[2], and Shimon (Moni) Shahar[1]

[1] School of Electrical Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel
{guy, moni}@eng.tau.ac.il
[2] Caesarea Rothschild Institute, University of Haifa, Haifa 31905, Israel
rawitz@cri.haifa.ac.il

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. We study the capacitated version of this problem in which the input includes an integral capacity for each line that 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 the case of 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.

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 [1] and the restricted case of vertex-cover [2, 3]. 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. 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 $\mathcal{U}$. The goal is to find a cover of minimum size. We denote the set of rectangles that a line S intersects by $\mathcal{U}(S)$. Using this notation, an RS instance is a set-cover instance in which the goal is to find a collection of subsets $\mathcal{U}(S)$, the union of which equals $\mathcal{U}$. W.l.o.g., we assume that the RS instance is discrete in the following sense [4]: rectangle corners have integral coordinates and lines intersect

T. Calamoneri, I. Finocchi, G.F. Italiano (Eds.): CIAC 2006, LNCS 3998, pp. 18–29, 2006.

the axes at integral points. In the one-dimensional version, the set $\mathcal{U}$ consists of horizontal intervals and the set $\mathcal{S}$ consists of points. This is the well known polynomial clique cover problem in interval graphs. RS can be extended to d dimensions (dRS). For $d \geq 3$, $\mathcal{U}$ consists of axis parallel d-dimensional rectangles ("boxes") and the set $\mathcal{S}$ consists of hyper-planes that are orthogonal to one of the d axes ("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-dRS) and covering with soft capacities (SOFT-dRS).

A SOFT-dRS cover is formally defined as follows. 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 $\sum_{S \in \mathcal{S}} \alpha(A, S)$. 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 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-dRS, 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-dRS is a special case of HARD-dRS, since given a SOFT-dRS instance one can always transform it into a HARD-dRS 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 [5]. Hassin and Megiddo [6] showed that RS is NP-hard, for $d \geq 2$. Gaur et al. [4] presented a d-approximation algorithm for dRS that uses linear programming to reduce d dimensions to one dimension.

Capacitated covering problems (even with weights) date back to Wolsey [1] (see also [7, 2]). Wolsey presented a greedy algorithm for weighted set-cover with hard capacities that achieves a logarithmic approximation ratio. Guha et al. [3] 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. [8] improved the approximation ratio for capacitated vertex cover to 2.

Our results. We present a 2-approximation algorithm for SOFT-1RS. This algorithm is a dynamic programming algorithm that finds an optimal solution of a certain form. In the full paper we show that this algorithm extends to weighted SOFT-1RS. We present a $6d$-approximation algorithm for SOFT-dRS, where d is arbitrary. This algorithm solves an LP relaxation of the problem, and rounds it using 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-dRS. Our algorithm 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. In the full paper, we present two hardness results. The first mimics the hardness result given in [2] to show that weighted HARD-2RS is set-cover-hard, even if all weights are in $\{0, 1\}$. The second result proves that it is NP-hard to approximate dRS 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 Interval Stabbing with Soft Capacities

In this section we present a 2-approximation algorithm for SOFT-1RS. 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. Hyper-planes in the one dimensional case are simply points. Since intervals are drawn as horizontal intervals with different heights, we refer to the hyper-planes as vertical lines instead of points. To summarize, the input in SOFT-1RS consists of a set $\mathcal{U}$ of horizontal intervals, and a set $\mathcal{S}$ of vertical lines with capacities $c(S)$.

The presentation 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.

Definition 1. *The total order $\prec$ is defined over the set $\mathcal{S}$ of 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'.*

Consider a cover A. Suppose that the support $\sigma(A)$ of a cover A is $\{S_1, S_2, \ldots, S_k\}$, where $S_1 \prec S_2 \prec \cdots \prec S_k$.

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

In a decisive cover each interval u is covered by the smallest (according to $\prec$) 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 1. *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 a 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. □

Next, we present a dynamic programming algorithm that finds an optimal decisive cover. According to Claim 1 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 $[i, j]$.

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 $\mathcal{U}(i, j) = \emptyset$, and $\Pi(i, j, k) = \infty$ 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, $\alpha(S, i, j) = \lceil |\{u \in \mathcal{U}(i, j) \mid \ell(u) \leq x_S \leq r(u)\}| / c(S) \rceil$. The following recurrence is used:

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

Note that, 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 $\Pi(i,j,k) < \Pi(i,j,k-1)$, 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.

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-dRS. To simplify notation we write $u \in S$ instead of $u \in \mathcal{U}(S)$.

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

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 covering constraints. Capacity constraints are formulated using constraints of types (2) and (3). Constraints of type (4) and (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). If $u \notin S$, we assign $y(S,u) = 0$.

An LP-relaxation of SOFT-dRS is obtained by omitting constraints of type (4). We denote the LP-relaxation 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 is *covered* if its type (1) constraint is satisfied.

If $\sum_{S \mid u \in S} y(S,u) \geq \alpha$, we refer to u as *α-covered*. If $\sum_{S \mid u \in S} y(S,u) > 0$ 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 OPT*. W.l.o.g. we assume that the covering constraints are tight, i.e., that $\sum_{S \mid u \in S} y^*(S,u) = 1$ for every $u \in \mathcal{U}$.

3.2 A Network Flow Formulation

This section is written in HARD-dRS terms, but similar arguments can be made in the case of SOFT-dRS. It is very useful to view the LP relaxation as a network flow problem [7, 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. 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 1. *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 $f_y(u,t) = \sum_{S \mid u \in S} y(S,u)$, 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.

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

The mapping from flows to vectors is defined similarly. □

We refer to $f_y(s,S)$ as the *flow supplied by* S and to $f_y(u,t)$ as the *flow delivered to* u. For simplicity, 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-dRS.

Observation 2. *Feasibility of a* HARD-dRS *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 implies that it suffices to compute a feasible cover (x,y), where x is integral.

Observation 3. *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 3. *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 4. *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-dRS. 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 $H \triangleq \{S \,|\, x^*(S) \geq \varepsilon\}$ and $L \triangleq \{S \,|\, x^*(S) < \varepsilon\}$. 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, $\varepsilon \leq \sum_{S \in L^h_j} x^*(S) < 2\varepsilon$ for every $j \leq b(h)$ and $\sum_{S \in \tilde{L}^h} x^*(S) < \varepsilon$. 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 5. *The number of blocks (not including the leftover block) in each dimension satisfies $b(h) \leq \frac{1}{\varepsilon} \cdot \sum_{S \in L^h} x^*(S)$ and $b(v) \leq \frac{1}{\varepsilon} \cdot \sum_{S \in L^v} x^*(S)$.*

Let $S^*_{h,j}$ and $S^*_{v,j}$ denote lines of maximum capacity in L^h_j and L^v_j, respectively. Let $L^* \triangleq \{S^*_{h,j} \,|\, 1 \leq j \leq b(h)\} \cup \{S^*_{v,j} \,|\, 1 \leq j \leq b(v)\}$.

Definition 4. *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, $x(S) = \sum_{S' \in B(S)} x^*(S')$ and $y(S, u) = \sum_{S' \in B(S)} y^*(S', u)$. The remaining components of the solution (x, y) are set to zero.*

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 2. *(x, y) is a partial cover.*

Proof. We first show that constraints of type (3) are satisfied. Clearly, this is true for $S \notin L^*$. Consider a line $S^* \in L^*$, and let B denote the block of lines in L that contains S^*. For every rectangle $u \in S^*$, the following holds: $y(S^*, u) = \sum_{S' \in B} y^*(S', u) \leq \sum_{S' \in B} x^*(S') = x(S^*)$. Next, we show that constraints of type (2) are satisfied. 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^*. It follows that

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

where the first inequality follows from the fact that some rectangles may lose part of their flow, the second inequality is due to the feasibility of (x^*, y^*), and the third inequality follows from Def. 4. □

Claim 3. *The coverage of every rectangle u is greater than $(1 - 4d\varepsilon)$ in the partial cover (x, y).*

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. 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 3 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. 3 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-dRS 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-dRS. 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 3. We obtain a 1/2-cover simply by setting $\varepsilon = \frac{1}{8d}$. Note that $x(S) \leq 1$, for every line S, hence two copies are not less than scaling by two and rounding up. Note that we rely on Obs. 2 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. 5) we obtain, $|L^*| \leq \frac{1}{\varepsilon} \cdot \sum_{S \in L} x^*(S)$. It follows that $|H \cup L^*| \leq \frac{1}{\varepsilon} \cdot \sum_{S \in \mathcal{S}} x^*(S)$, as required.

6 Interval Stabbing with Hard Capacities

In this section we present an 8-approximation algorithm for HARD-1RS. The algorithm augments the positive cover obtained by Claim 3 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.

Definition 5. *Let (x, y) be a partial cover such that x is integral and y is maximum with respect to x. 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 (s, S) belongs to a min-cut in N_x but not to every min-cut.

Lemma 1. *Let (x, y) be a partial cover such that x is integral and y is maximum with respect to x. If $S \in x$ 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). First, $S' \in x$ since x is integral and $y(S', u) > 0$. 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. 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 be an augmenting path in N_x obtained when $c(S')$ is increased (p exists since 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) $f_y(S,u) \leq 1 - y(S',u) < 1 = x(S)$, 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 1.

Corollary 1. *Let (x,y) be a partial cover such that x is integral and y is maximum with respect to x. Define: $D \triangleq \{S \in x \mid S \text{ is a dam}\}$ and $\mathcal{U}_D \triangleq \{u \in \mathcal{U} \mid \exists S \in D \text{ such that } u \in S\}$. Then, for every $u \in \mathcal{U}_D$, $\sum_{S \in D} y(S,u) = 1$.*

Next, we show that if no thirsty lines exist in a positive partial cover, then the cover is feasible.

Corollary 2. *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. 1, 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 6. *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 7. *Let (x,y) be a partial cover, where x is integral and y is maximum with respect to x. 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 (x,y) denote a partial cover with an integral x and y that is maximum with respect to x. 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 6. *$\Delta_y(B_i) \leq 0$, for every $i \in I$, if and only if (x,y) is feasible.*

6.3 The Approximation Algorithm

The approximation algorithm for HARD-1RS begins like the bi-criteria approximation algorithm and then applies a new augmentation procedure, called **make-feasible**. The algorithm proceeds as follows: (i) Solve 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 rounding of the LP solution as described in Def. 4. (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. 4).

Algorithm **make-feasible** iteratively augments the partial cover until a cover is obtained. Since a new line is added to the cover in each iteration, the output component x is integral. By Obs. 6, Algorithm **make-feasible** stops 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 in B, (ii) add $S_{\max}$ 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 recurses with (x', y'). Throughout this section, the x-component of every partial cover is integral. 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 $\{S\}$.

Algorithm 1. **make-feasible**$(\mathcal{S}, \mathcal{U}, x, y)$

1: Termination 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 $S_{\max} \leftarrow \text{argmax}\{c^B(S) : S \in \mathcal{S}^B \setminus x\}$.
4: Add $S_{\max}$ to x: $x' \leftarrow x \cup \{S_{\max}\}$.
5: Augment flow: $y' \leftarrow$ **aug-flow**(x', y).
6: Recurse: **Return make-feasible**$(\mathcal{S}, \mathcal{U}, x', y')$.

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. 6. Hence, it remains to prove that $S_{\max}$ is well defined.

Claim 4. *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. 6 that B is not active, a contradiction. □

The algorithm 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. 4.

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

The proof is omitted for lack of space. 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 8OPT^* lines.

7 Open Problems

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

Gaur et al. [4] presented a d-approximation algorithm for weighted dRS that uses linear programming to reduce the problem to d one-dimensional instances. Their analysis relies on the integrality of the LP relaxation in the one dimensional case. Our 2-approximation for weighted SOFT-1RS does not prove a bound on the integrality gap. Our 6-approximation algorithm for unweighted SOFT-1RS proves that integrality gap of the one-dimensional case is bounded by 6. Hence another $6d$-approximation ratio follows by combining a reduction similar to the Gaur et al. [4] and our 6-approximation algorithm for SOFT-1RS.

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

References

1. Wolsey, L.A.: An analysis of the greedy algorithm for the submodular set covering problem. Combinatorica **2** (1982) 385–393
2. Chuzhoy, J., Naor, J.: Covering problems with hard capacities. In: 43nd IEEE Symposium on Foundations of Computer Science. (2002) 481–489
3. Guha, S., Hassin, R., Khuller, S., Or, E.: Capacitated vertex covering. Journal of Algorithms **48**(1) (2003) 257–270
4. Gaur, D.R., Ibaraki, T., Krishnamurti, R.: Constant ratio approximation algorithms for the rectangle stabbing problem and the rectilinear partitioning problem. Journal of Algorithms **43** (2002) 138–152
5. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Academic Press, New York (1980)
6. Hassin, R., Megiddo, N.: Approximation algorithms for hitting objects with straight lines. Discrete Applied Mathematics **30**(1) (1991) 29–42
7. Bar-Ilan, J., Kortsarz, G., Peleg, D.: Generalized submodular cover problems and applications. Theoretical Computer Science **250** (2001) 179–200
8. Gandhi, R., Halperin, E., Khuller, S., Kortsarz, G., Srinivasan, A.: An improved approximation algorithm for vertex cover with hard capacities. In: 30th Annual International Colloquium on Automata, Languages and Programming. Volume 2719 of LNCS. (2003) 164–175