A PTAS for the Horizontal Rectangle Stabbing Problem

Arindam Khan[1], Aditya Subramanian[1](✉), and Andreas Wiese[2]

[1] Indian Institute of Science, Bengaluru, India
{arindamkhan,adityasubram}@iisc.ac.in
[2] Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
a.wiese@vu.nl

Abstract. We study rectangle stabbing problems in which we are given n axis-aligned rectangles in the plane that we want to *stab*, i.e., we want to select line segments such that for each given rectangle there is a line segment that intersects two opposite edges of it. In the *horizontal rectangle stabbing problem* (Stabbing), the goal is to find a set of horizontal line segments of minimum total length such that all rectangles are stabbed. In *general rectangle stabbing problem*, also known as *horizontal-vertical stabbing problem* (HV-Stabbing), the goal is to find a set of rectilinear (i.e., either vertical or horizontal) line segments of minimum total length such that all rectangles are stabbed. Both variants are NP-hard. Chan, van Dijk, Fleszar, Spoerhase, and Wolff [5] initiated the study of these problems by providing $O(1)$-approximation algorithms. Recently, Eisenbrand, Gallato, Svensson, and Venzin [11] have presented a QPTAS and a polynomial-time 8-approximation algorithm for Stabbing but it is open whether the problem admits a PTAS.

In this paper, we obtain a PTAS for Stabbing, settling this question. For HV-Stabbing, we obtain a $(2+\varepsilon)$-approximation. We also obtain PTASes for special cases of HV-Stabbing: (i) when all rectangles are squares, (ii) when each rectangle's width is at most its height, and (iii) when all rectangles are δ-large, i.e., have at least one edge whose length is at least δ, while all edge lengths are at most 1. Our result also implies improved approximations for other problems such as *generalized minimum Manhattan network*.

Keywords: Geometric optimization · Approximation algorithms · Line stabbing · Rectangles

1 Introduction

Rectangle stabbing problems are natural geometric optimization problems. Here, we are given a set of n axis-parallel rectangles $\mathcal{R}$ in the two-dimensional plane. For each rectangle $R_i \in \mathcal{R}$, we are given points $(x_1^{(i)}, y_1^{(i)}), (x_2^{(i)}, y_2^{(i)}) \in \mathbb{R}^2$ that denote its bottom-left and top-right corners, respectively. Also, we denote its *width* and *height* by $w_i := x_2^{(i)} - x_1^{(i)}$ and $h_i := y_2^{(i)} - y_1^{(i)}$, respectively. Our goal

K. Aardal and L. Sanitá (Eds.): IPCO 2022, LNCS 13265, pp. 361–374, 2022.
https://doi.org/10.1007/978-3-031-06901-7_27

is to compute a set of line segments $\mathcal{L}$ that stab all input rectangles. We call a rectangle *stabbed* if a segment $\ell \in \mathcal{L}$ intersects both of its horizontal or both of its vertical edges. We study several variants. In the *horizontal rectangle stabbing problem* (Stabbing) we want to find a set of horizontal segments of minimum total length such that each rectangle is stabbed. The *general rectangle stabbing* (HV-Stabbing) problem generalizes Stabbing and involves finding a set of axis-parallel segments of minimum total length such that each rectangle in $\mathcal{R}$ is stabbed. The *general square stabbing* (Square-Stabbing) problem is a special case of HV-Stabbing where all rectangles in the input instance are squares. These problems have applications in bandwidth allocation, message scheduling with time-windows on a direct path, and geometric network design [3,5,10].

Note that Stabbing and HV-Stabbing are special cases of weighted geometric set cover problem, where the rectangles correspond to elements and potential line segments correspond to sets, and the weight of a set equals the length of the corresponding segment. A set contains an element if the corresponding line segment stabs the corresponding rectangle. This already implies an $O(\log n)$-approximation algorithm [9] for HV-Stabbing and Stabbing.

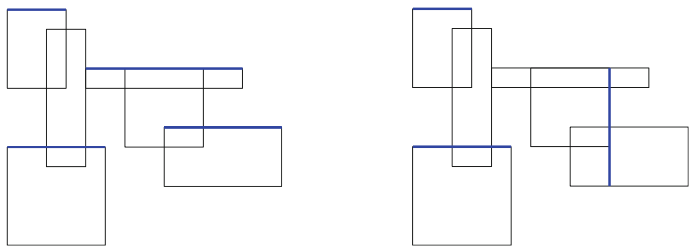
Chan, van Dijk, Fleszar, Spoerhase, and Wolff [5] initiated the study of Stabbing. They proved Stabbing to be NP-hard via a reduction from planar vertex cover. Also, they presented a constant[1] factor approximation algorithm using *decomposition techniques* and the *quasi-uniform sampling* method [19] for weighted geometric set cover. In particular, they showed that Stabbing instances can be decomposed into two disjoint *laminar* set cover instances of small shallow cell complexity for which the *quasi-uniform sampling* yields an $O(1)$-approximation using techniques from [8].

Recently, Eisenbrand, Gallato, Svensson, and Venzin [11] presented a quasi-polynomial time approximation scheme (QPTAS) for Stabbing. This shows that Stabbing is not APX-hard unless NP $\subseteq$ DTIME($2^{\text{poly}\log n}$). The QPTAS relies on *the shifting technique* by Hochbaum and Maass [14], applied to a grid, consisting of randomly shifted vertical grid lines that are equally spaced. With this approach, the plane is partitioned into narrow disjoint vertical strips which they then process further. Then, this routine is applied recursively. They also gave a polynomial time dynamic programming based exact algorithm for Stabbing for laminar instances (in which the projections of the rectangles to the x-axis yield a laminar family of intervals). Then they provided a simple polynomial-time 8-approximation algorithm by reducing any given instance to a laminar instance. It remains open whether there is a PTAS for the problem.

1.1 Our Results

In this paper, we give a PTAS for Stabbing and thus resolve this open question. Also, we extend our techniques to HV-Stabbing for which we present a polynomial time $(2 + \varepsilon)$-approximation and PTASes for several special cases: when all

[1] The constant is not explicitly stated, and it depends on a not explicitly stated constant in [7].



(a) Horizontal Rectangle Stabbing

(b) General Rectangle Stabbing

Fig. 1. A solution for an instance of Stabbing and HV-Stabbing.

input rectangles are squares, more generally when for each rectangle its width is at most its height, and finally for δ-large rectangles, i.e., when each rectangle has one edge whose length is within $[\delta, 1]$ and 1 is the maximum length of each edge of any input rectangle (in each dimension).

Our algorithm for Stabbing is in fact quite easy to state: it is a dynamic program (DP) that recursively subdivides the plane into smaller and smaller rectangular regions. In the process, it guesses line segments from OPT. However, its analysis is intricate. We show that there is a sequence of recursive decompositions that yields a solution whose overall cost is $(1+\varepsilon)$OPT. Instead of using a set of equally spaced grid lines as in [11], we use a *hierarchical* grid with several levels for the decomposition. In each level of our decomposition, we subdivide the given rectangular region into strips of narrow width and guess $O_\varepsilon(1)$ line segments from OPT inside them which correspond to the current level. One crucial ingredient is that we slightly extend the segments, such that the guessed horizontal line segments are aligned with our grid. The key consequence is that it will no longer be necessary to remember these line segments once we have advances three levels further in the decomposition. Also, for the guessed vertical line segments of the current level we introduce additional (very short) horizontal line segments, such that we do not need to remember them either, once we advanced three levels more in the decomposition. Therefore, the DP needs to remember previously guessed line segments from only the last three previous levels and afterwards these line segments *vanish*. This allows us to bound the number of arising subproblems (and hence of the DP-cells) by a polynomial.

Our techniques easily generalize to a PTAS for Square-Stabbing and to a PTAS for HV-Stabbing if for each input rectangle its width is at most its height, whereas the QPTAS in [11] worked only for Stabbing. We use the latter PTAS as a subroutine in order to obtain a polynomial time $(2+\varepsilon)$-approximation for HV-Stabbing – which improves on the result in [5].

Then, we extend our techniques above to the setting of δ-large rectangles of HV-Stabbing. This is an important subclass of rectangles and are well-studied for other geometric problems [1,15]. To this end, we first reduce the problem to the setting in which all input rectangles are contained in a rectangular box that admits a solution of cost $O(1/\varepsilon^3)$. Then we guess the relatively long line segments in OPT in polynomial time. The key argument is that then the remaining problem splits into two independent subproblems, one for the horizontal and one for the vertical line segments in OPT. For each of those, we then apply our PTAS for Stabbing which then yields a PTAS for HV-Stabbing if all rectangles are δ-large.

Finally, our PTAS for Stabbing implies improved approximation ratios for the Generalized Minimum Manhattan Network (GMMN) and x-separated 2D-GMMN problems, of $(4+\varepsilon)\log n$ and $4+\varepsilon$, respectively, by improving certain subroutines of the algorithm in [10].

Due to space limitations, many proofs had to be omitted, and we refer the reader to the full version of this paper [16].

1.2 Further Related Work

Finke et al. [12] gave a polynomial time exact algorithm for a special case of Stabbing where all input rectangles have their left edges lying on the y-axis. Das et al. [10] studied a related variant in the context of the Generalized Minimum Manhattan Network (GMMN) problem. In GMMN, we are given a set of n terminal-pairs and the goal is to find a minimum-length rectilinear network such that each pair is connected by a *Manhattan path*. They obtained a 4-approximation for a variant of Stabbing where all rectangles intersect a vertical line. Then they used it to obtain a $(6+\varepsilon)$-approximation algorithm for the x-separated 2D-GMMN problem, a special case of 2D-GMMN, and $(6+\varepsilon)(\log n)$-approximation for 2-D GMMN.

Gaur et al. [13] studied the problem of stabbing rectangles by a *minimum number of axis-aligned lines* and gave an LP-based 2-approximation algorithm. Kovaleva and Spieksma [17] considered a weighted generalization of this problem and gave an $O(1)$-approximation algorithm.

Stabbing and HV-Stabbing are related to geometric set cover which is a fundamental geometric optimization problem. Brönnimann and Goodrich [4] in a seminal paper gave an $O(d\log(d\text{OPT}))$-approximation for unweighted geometric set cover where d is the dual VC-dimension of the set system and OPT is the value of the optimal solution. Using ε-*nets*, Aronov et al. [2] gave an $O(\log\log OPT)$-approximation for hitting set for axis-parallel rectangles. Later, Varadarajan [19] developed quasi-uniform sampling and provided sub-logarithmic approximation for weighted set cover where sets are weighted fat triangles or weighted disks. Chan et al. [8] generalized this to any set system with low *shallow cell complexity*. Afterward, Chan and Grant [6] and Mustafa et al. [18] have settled the APX-hardness statuses of all natural weighted geometric set cover problems.

2 Dynamic Program

We present a dynamic program that computes a $(1+\varepsilon)$-approximation to HV-Stabbing for the case where $h_i \geq w_i$ for each rectangle $R_i \in \mathcal{R}$. This implies directly a PTAS for the setting of squares for the same problem, and we will argue that it also yields a PTAS for Stabbing. Later, we will use it as a subroutine to obtain a $(2+\varepsilon)$-approximation for HV-Stabbing and a PTAS for the setting of δ-large rectangles of HV-Stabbing.

For a line segment ℓ, we use the notation $|\ell|$ to represent its length, and for a set of segments $\mathcal{L}$, we use notation $c(\mathcal{L})$ to represent the cost of the set, which is also the total length of the segments contained in it. We use the term OPT interchangeably to refer to the optimal solution to the problem and also to $c(\text{OPT})$, i.e., the cost of the optimal solution.

2.1 Preprocessing Step

First, we show that by some simple scaling and discretization steps we can ensure some simple properties that we will use later. Without loss of generality we assume that $(1/\varepsilon) \in \mathbb{N}$ and we say that a value $x \in \mathbb{R}$ is *discretized* if x is an integral multiple of ε/n.

Lemma 1. *For any positive constant $\varepsilon < 1/3$, by losing a factor $(1+O(\varepsilon))$ in the approximation ratio, we can assume for each $R_i \in \mathcal{R}$ the following properties hold:*

- $\varepsilon/n \leq w_i \leq 1$,
- $x_1^{(i)}, x_2^{(i)}$ *are discretized and within* $[0, n]$,
- $y_1^{(i)}, y_2^{(i)}$ *are discretized and within* $[0, 4n^2]$, *and*
- *each horizontal line segment in the optimal solution has width at most* $1/\varepsilon$.

Henceforth in this paper when we refer to the set of input rectangles $\mathcal{R}$, we are referring to a set $\mathcal{R}'$ that has been obtained after applying the preprocessing from Lemma 1 to the input set $\mathcal{R}$, and when we refer to OPT, we are referring to the optimal solution to the set of rectangles $\mathcal{R}'$, which is a $(1+O(\varepsilon))$ approximation of the optimal solution of the input instance.

2.2 Description of the Dynamic Program

Our algorithm is based on a dynamic program (DP). It has a cell $\text{DP}(S, \mathcal{L})$ for each combination of

- a rectangular region $S \subseteq [0, n] \times [0, 4n^2]$ with discretized coordinates (that is not necessarily equal to an input rectangle in $\mathcal{R}$).
- a set $\mathcal{L}$ of at most $3(1/\varepsilon)^3$ line segments, each of them horizontal or vertical, such that for each $\ell \in \mathcal{L}$ we have that $\ell \subseteq S$, and all coordinates of ℓ are discretized.

This DP-cell corresponds to the subproblem of stabbing all rectangles in $\mathcal{R}$ that are contained in S and that are not already stabbed by the line segments in $\mathcal{L}$. Therefore, the DP stores solution $\text{SOL}(S, \mathcal{L})$ in the cell $\text{DP}(S, \mathcal{L})$ such that $\text{SOL}(S, \mathcal{L}) \cup \mathcal{L}$ stabs all rectangles in $\mathcal{R}$ that are contained in S.

Given a DP-cell $\mathrm{DP}(S,\mathcal{L})$, our DP computes a solution for it as follows. If $\mathcal{L}$ already stabs each rectangle from $\mathcal{R}$ that is contained in S, then we simply define a solution $\mathrm{SOL}(S,\mathcal{L}) := \emptyset$ for the cell $\mathrm{DP}(S,\mathcal{L})$ and do not compute anything further. Another simple case is when there is a line segment $\ell \in \mathcal{L}$ such that $S \setminus \ell$ is divided into two rectangular regions S_1, S_2 (these regions are henceforth referred to as connected components). In this case we define $\mathrm{SOL}(S,\mathcal{L}) := \mathrm{SOL}(S_1,\mathcal{L}\cap S_1) \cup \mathrm{SOL}(S_2,\mathcal{L}\cap S_2) \cup \{\ell\}$, where for any set of line segments $\mathcal{L}'$ and any rectangle S' we define $\mathcal{L}'\cap S' := \{\ell'\cap S' | \ell' \in \mathcal{L}' \wedge \ell'\cap S' \neq \emptyset\}$. In case that there is more than one such line segment $\ell \in \mathcal{L}$ then we pick one according to some arbitrary but fixed global tie-breaking rule. We will later refer to this as *trivial operation.*

Otherwise, we do each of the following operations which produces a set of candidate solutions:

1. *Add operation:* Consider each set $\mathcal{L}'$ of line segments with discretized coordinates such that $|\mathcal{L}| \cup |\mathcal{L}'| \leq 3\varepsilon^{-3}$ and each $\ell \in \mathcal{L}'$ is contained in S and horizontal or vertical. For each such set $\mathcal{L}'$ we define the solution $\mathcal{L}' \cup \mathrm{SOL}(S,\mathcal{L}\cup\mathcal{L}')$ as a candidate solution.
2. *Line operation:* Consider each vertical/horizontal line ℓ with a discretized vertical/horizontal coordinate such that $S \setminus \ell$ has two connected components S_1 and S_2. Let $\mathcal{R}_\ell$ denote the rectangles from $\mathcal{R}$ that are contained in S and that are stabbed by ℓ. For the line ℓ we do the following:
   (a) compute an $O(1)$-approximate solution $\mathcal{L}(\mathcal{R}_\ell)$ for the rectangles in $\mathcal{R}_\ell$ using the polynomial time algorithm in [5].
   (b) produce the candidate solution $\mathcal{L}(\mathcal{R}_\ell) \cup \mathrm{SOL}(S_1,\mathcal{L}\cap S_1) \cup \mathrm{SOL}(S_2,\mathcal{L}\cap S_2)$.

Note that in the line operation we consider entire lines, not just line segments. We define $\mathrm{SOL}(S,\mathcal{L})$ to be the solution of minimum cost among all the candidate solutions produced above and store it in $\mathrm{DP}(S,\mathcal{L})$.

We do the operation above for each DP-cell $\mathrm{DP}(S,\mathcal{L})$. Finally, we output the solution $\mathrm{SOL}([0,n]\times[0,4n^2],\emptyset)$, i.e., the solution corresponding to the cell $\mathrm{DP}([0,n]\times[0,4n^2],\emptyset)$.

We remark that instead of using the $O(1)$-approximation algorithm in [5] for stabbing the rectangles in $\mathcal{R}_\ell$, one could design an algorithm with a better approximation guarantee, using the fact that all rectangles in $\mathcal{R}_\ell$ are stabbed by the line ℓ. However, for our purposes an $O(1)$-approximate solution is good enough.

2.3 Definition of DP-Decision Tree

We want to show that the DP above computes a $(1+\varepsilon)$-approximate solution. For this, we define a tree T in which each node corresponds to a cell $\mathrm{DP}(S,\mathcal{L})$ of the DP and a corresponding solution $\overline{\mathrm{SOL}}(S,\mathcal{L})$ to this cell. The root node of T corresponds to the cell $\mathrm{DP}([0,n]\times[0,4n^2],\emptyset)$. Intuitively, this tree represents doing one of the possible operations above, of the DP in the root problem $\mathrm{DP}([0,n]\times[0,4n^2],\emptyset)$ and recursively one of the possible operations in each resulting DP-cell. The corresponding solutions in the nodes are the solutions obtained by choosing exactly these operations in each DP-cell. Since the DP always picks the solution of minimum total cost this implies that the computed solution has a cost that is at most the cost of the root, $c(\overline{\text{SOL}}([0,n]\times[0,4n^2],\emptyset))$.

Formally, we require T to satisfy the following properties. We require that a node v is a leaf if and only if for the corresponding DP-cell $\text{DP}(S,\mathcal{L})$ the DP directly defined that $\text{DP}(S,\mathcal{L})=\emptyset$ because all rectangles in $\mathcal{R}$ that are contained in S are already stabbed by the segments in $\mathcal{L}$. If a node v for a DP-cell $\text{DP}(S,\mathcal{L})$ has one child then we require that we reduce the problem for $\text{DP}(S,\mathcal{L})$ to the child by applying the add operation, i.e., there is a set $\mathcal{L}'$ of horizontal/vertical line segments with discretized coordinates such that $|\mathcal{L}|\cup|\mathcal{L}'|\leq 3(1/\varepsilon)^3$, the child node of v corresponds to the cell $\text{DP}(S,\mathcal{L}\cup\mathcal{L}')$, and $\overline{\text{SOL}}(S,\mathcal{L})=\overline{\text{SOL}}(S,\mathcal{L}\cup\mathcal{L}')\cup\mathcal{L}'$.

Similarly, if a node v has two children then we require that we can reduce the problem of $\text{DP}(S,\mathcal{L})$ to these two children by applying the trivial operation or the line operation. Formally, assume that the child nodes correspond to the subproblems $\text{DP}(S_1,\mathcal{L}_1)$ and $\text{DP}(S_2,\mathcal{L}_2)$. If there is a segment $\ell\in\mathcal{L}$ such that $S_1\cup S_2\cup\ell=S$, then the applied operation was a trivial operation, and it must also be true that $\mathcal{L}_1\cup\mathcal{L}_2\cup\{\ell\}=\mathcal{L}$ and $\overline{\text{SOL}}(S,\mathcal{L})=\overline{\text{SOL}}(S_1,S_1\cap\mathcal{L})\cup\overline{\text{SOL}}(S_2,S_2\cap\mathcal{L})$. If no such segment exists, then the applied operation was a line operation on a line along the segment ℓ, such that $S_1\cup S_2\cup\ell=S$, $\mathcal{L}_1\cup\mathcal{L}_2=\mathcal{L}$, and $\overline{\text{SOL}}(S,\mathcal{L})=\overline{\text{SOL}}(S_1,S_1\cap\mathcal{L})\cup\overline{\text{SOL}}(S_2,S_2\cap\mathcal{L})\cup\mathcal{L}(\mathcal{R}_\ell)$; where $\mathcal{L}(\mathcal{R}_\ell)$ is a $O(1)$-approximate solution for the set of segments stabbing the set of rectangles intersected by ℓ.

We call a tree T with these properties a *DP-decision-tree*. If there exists a DP-decision-tree with cost $(1+\varepsilon)\text{OPT}$, then our DP computes a solution with at most this cost since the choices in each node of the DP-decision-tree are possible choices of the DP in each node, and in each node the DP makes the choice that minimizes the overall costs.

Lemma 2. *If there is a DP-decision-tree T' for which $c(\overline{\text{SOL}}([0,n]\times[0,4n^2],\emptyset))\leq(1+\varepsilon)\text{OPT}$ then the DP is a $(1+\varepsilon)$-approximation algorithm with a running time of $(n/\varepsilon)^{O(1/\varepsilon^3)}$.*

We define now a DP-decision-tree for which $c(\overline{\text{SOL}}(S,\mathcal{L}))\leq(1+\varepsilon)\text{OPT}$. Assume w.l.o.g. that $1/\varepsilon\in\mathbb{N}$. We start by defining a hierarchical grid of vertical lines. Let $a\in\mathbb{N}_0$ be a random offset to be defined later. The grid lines have levels. For each level $j\in\mathbb{N}_0$, there is a grid line $\{a+k\cdot\varepsilon^{j-2}\}\times\mathbb{R}$ for each $k\in\mathbb{N}$. Note that for each $j\in\mathbb{N}_0$ each grid line of level j is also a grid line of level $j+1$. Also note that any two consecutive lines of some level j are exactly ε^{j-2} units apart.

We say that a line segment $\ell\in\text{OPT}$ is *of level j* if the length of ℓ is in $(\varepsilon^j,\varepsilon^{j-1}]$ (Note that we can have vertical segments which are longer than $1/\varepsilon$, we consider these also to be of level 0). We say that a *horizontal* line segment of some level j is *well-aligned* if both its left and its right x-coordinates lie on a grid line of level $j+3$, i.e., if both of its x-coordinates are of the form $a+k\cdot\varepsilon^{j+1}$. We say that a *vertical* line segment of some level j is *well-aligned* if both its top and bottom y-coordinates are integral multiples of ε^{j+1}. This would be similar to the segment's end points lying on an (imaginary) horizontal grid line of level $j+3$. In order to make a line segment from OPT well-aligned, it suffices to extend it by a factor $1+O(\varepsilon)$, which hence increases the cost by at most this factor.

Lemma 3. *By losing a factor* $1+O(\varepsilon)$*, we can assume that each line segment* $\ell \in \mathrm{OPT}$ *is well-aligned.*

We define the tree T by defining recursively one of the possible operations (trivial operation, add operation, line operation) for each node v of the tree. After applying an operation, we always add children to the processed node v that corresponds to the subproblems that we reduce to, i.e., for a node v corresponding to the subproblem $\mathrm{DP}(S, \mathcal{L})$, if we are applying the trivial (resp. line) operation along a segment (resp. line) ℓ, then we add children corresponding to the DP subproblems $\mathrm{DP}(S_1, S_1 \cap \mathcal{L})$ and $\mathrm{DP}(S_2, S_2 \cap \mathcal{L})$, where S_1 and S_2 are the connected components of $S \backslash \ell$. Similarly if we apply the add operation on v with the set of segments $\mathcal{L}'$ then we add the child node corresponding to the subproblem $\mathrm{DP}(S, \mathcal{L} \cup \mathcal{L}')$.

First level. We start with the root $\mathrm{DP}([0,n] \times [0,4n^2], \emptyset)$. We apply the line operation for each vertical line that corresponds to a (vertical) grid line of level 0. Consider one of the resulting subproblems $\mathrm{DP}(S, \emptyset)$. Suppose that there are more than ε^{-3} line segments (horizontal or vertical) from OPT of level 0 inside S. We want to partition S into smaller rectangles, such that within each of these rectangles S' at most $O(\varepsilon^{-3})$ of these level 0 line segments start or end. This will make it possible for us to guess them. To this end, we consider the line segments from OPT of level 0 inside S, take their endpoints and order these endpoints non-decreasingly by their y-coordinates. Let $p_1, p_2, ..., p_k$ be these points in this order. For each $k' \in \mathbb{N}$ with $k'/\varepsilon^3 \leq k$, we consider the point p_{k'/ε^3}. Let ℓ' be the horizontal line that contains p_{k'/ε^3}. We apply the line operation to ℓ'.

Lemma 4. *Let* $\mathrm{DP}(S', \emptyset)$ *be one of the subproblems after applying the operations above. There are at most* ε^{-3} *line segments* $\mathcal{L}'$ *(horizontal or vertical) from* OPT *of level 0 that have an endpoint inside* S'*.*

In each resulting subproblem $\mathrm{DP}(S', \emptyset)$, for each vertical line segment $\ell \in \mathrm{OPT}$ that crosses S', i.e., such that $S' \setminus \ell$ has two connected components, we apply the line operation for the line that contains ℓ. In each subproblem $\mathrm{DP}(S'', \emptyset)$ obtained after this step, we apply the add operation to the line segments from OPT of level 0 that intersects S'' (or to be more precise, their intersection with S''), i.e., to the set $\mathcal{L}' := \{\ell \cap S'' \mid \ell \in \mathrm{OPT} \wedge \ell \cap S'' \neq \emptyset \wedge \ell \,\text{is of level}\, 0\}$. Claim 4 implies that $|\mathcal{L}'| \leq \varepsilon^{-3}$. In each obtained subproblem, we apply the trivial operation until it is no longer applicable. We say that all these operations correspond to level 0.

Subsequent levels. Next, we do a sequence of operations that correspond to levels $j = 1, 2, 3, ...$ Assume by induction that for some j each leaf in the current tree

T corresponds to a subproblem $\mathrm{DP}(S, \mathcal{L})$ such that $\ell \cap S \in \mathcal{L}$ for each line segment $\ell \in \mathrm{OPT}$ of each level $j' < j$ for which $\ell \cap S \neq \emptyset$. Take one of these leaves and assume that it corresponds to a subproblem $\mathrm{DP}(S, \mathcal{L})$. We apply the line operation for each vertical line that corresponds to a (vertical) grid line of level j.

Consider a corresponding subproblem $\mathrm{DP}(S', \mathcal{L})$. Suppose that there more than ε^{-3} line segments (horizontal or vertical) from OPT of level j that have an endpoint inside S'. Like above, we consider these endpoints and we order them non-decreasingly by their y-coordinates. Let $p_1, p_2, ..., p_k$ be these points in this order. For each $k' \in \mathbb{N}$ with $k'/\varepsilon^3 \leq k$, we consider the point p_{k'/ε^3} and apply the line operation for the horizontal line ℓ' that contains p_{k'/ε^3}. If for a resulting subproblem $\mathrm{DP}(S'', \mathcal{L})$ there is a vertical line segment $\ell \in \mathcal{L}$ of some level $j' < j - 2$ with an endpoint p inside S'', then we apply the line operation for the horizontal line that contains p.

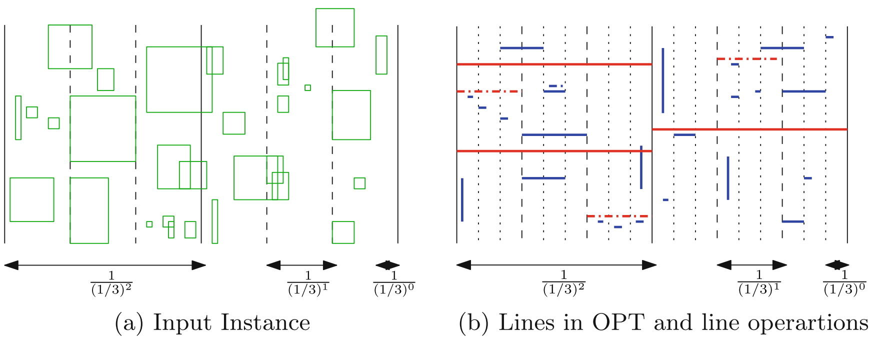


Fig. 2. Horizontal line operations

Lemma 5. *Let* $\mathrm{DP}(S', \mathcal{L})$ *be one of the subproblems after applying the operations above. There are at most* ε^{-3} *line segments* $\mathcal{L}'$ *(horizontal or vertical) from* OPT *of level* j *that have an endpoint inside* S'*.*

Consider a resulting subproblem $\mathrm{DP}(S'', \mathcal{L})$. For each line segment $\ell \in \mathrm{OPT}$ such that ℓ crosses S'', i.e., $S'' \setminus \ell$ has two connected components, we apply the line operation to the line that contains ℓ. We apply the trivial operation until it is no longer applicable. In each subproblem $\mathrm{DP}(S''', \mathcal{L})$ obtained after this step, we apply the add operation to the line segments of level j that have an endpoint in S''', i.e., to the set $\mathcal{L}' := \{\ell \cap S''' \mid \ell \in \mathrm{OPT} \wedge \ell \cap S''' \neq \emptyset \wedge \ell \,\text{is of level}\, j\}$.

As an example, look at Fig. 2, with $\varepsilon = 1/3$. The solid lines in it are of level 0, the dashed lines of level 1, and dotted lines of level 2. Also the black lines are vertical grid lines, the blue lines are (well-aligned) lines in OPT and the red lines are lines along which horizontal line operations are applied. It can be seen from this example that a segment of level j, by virtue of it being well-aligned, will get removed by a trivial operation of level less than $j + 3$.

2.4 Analysis of DP-Decision Tree

We want to prove that the resulting tree T is indeed a DP-decision-tree corresponding to a solution of cost at most $(1+\varepsilon)$OPT. To this end, first we need to show that whenever we apply the add operation to a subproblem DP$(S, \mathcal{L})$ for a set $\mathcal{L}'$ then $|\mathcal{L}| + |\mathcal{L}'| \leq 3\varepsilon^{-3}$. The *key insight* for this is that if we added a line segment $\ell \in$ OPT of some level j, then it will not be included in the respective set $\mathcal{L}$ of later subproblems of level $j+3$ or higher since ℓ is well-aligned. More precisely, if ℓ is horizontal then its x-coordinates are aligned with the grid lines of level $j+3$. Hence, if ℓ or a part of ℓ is contained in a set $\mathcal{L}$ of some subproblem DP$(S, \mathcal{L})$ for some level $j+3$, then we applied the trivial operation to ℓ and thus ℓ "disappeared" from $\mathcal{L}$ (note that here by disappear we mean that the segment does not need to be considered in $\mathcal{L}$ anymore, and gets added to the solution of the DP subproblem). If ℓ is vertical and it appears in a DP$(S, \mathcal{L})$ for some level $j+3$ then we applied the line operation to the horizontal lines that contain the two endpoints of ℓ. Afterwards, we applied the trivial operation to ℓ until ℓ "disappeared" from $\mathcal{L}$.

In particular, for each subproblem DP$(S, \mathcal{L})$ constructed by operations of level j, the set $\mathcal{L}$ can contain line segments of levels $j-2, j-1$, and j; but no line segments of a level j' with $j' < j-2$. Using this, we prove the following lemma.

Lemma 6. *The constructed tree T is a DP-decision-tree.*

We want to show that the cost of the solution corresponding to T is at most $(1+O(\varepsilon))$OPT. In fact, depending on the offset a this might or might not be true. However, we show that there is a choice for a such that this is true (in fact, we will show that for a random choice for a the cost will be at most $(1+O(\varepsilon))$OPT in expectation). Intuitively, when we apply the line operation to a vertical grid line ℓ of some level j then the incurred cost is at most $O(1)$ times the cost of the line segments from OPT of level j or larger that stab at least one rectangle intersected by ℓ. A line segment $\ell' \in$ OPT of level j stabs such a rectangle only if ℓ' is intersected by ℓ (if ℓ' is horizontal) or the x-coordinate of ℓ' is close to ℓ (if ℓ' is vertical). Here we use that $h_i \geq w_i$ for each rectangle $R_i \in$ OPT.

Thus, we want to bound the total cost over all levels j of the line segments from OPT that are in level j and that are intersected or close to grid lines of level j or smaller. We will show that if we choose a randomly then the total cost of such grid lines is at most $\varepsilon \cdot$ OPT in expectation. Hence, by using the constant approximation algorithm from [5] in expectation the total cost due to all line operations for vertical line segments is at most $O(\varepsilon) \cdot$ OPT.

When we apply the line operation for a horizontal line, then the cost of stabbing the corresponding rectangles is at most the width of the rectangle S of the current subproblem DP$(S, \mathcal{L})$. We will charge this cost to the line segments of OPT inside S of the current level or higher levels. We will argue that we can charge each such line operation to line segments from OPT whose total width is at least $1/\varepsilon$ times the width of S. This costs another $O(\varepsilon) \cdot$ OPT in total due to all applications of line operations for horizontal segments.

The add operation yields a cost of exactly OPT and the trivial operation does not cost anything. This yields a total cost of $(1 + O(\varepsilon))$OPT.

Lemma 7. *There is a choice for the offset a such that the solution $\overline{\mathrm{SOL}}([0, n] \times [0, 4n^2], \emptyset)$ in T has a cost of at most $(1 + O(\varepsilon))$OPT.*

Proof. In the tree as defined above, the add operations are only applied on segments from OPT, and hence the cost across all such add operations is at most c(OPT). Similarly, the trivial operations are applied on segments which were 'added' before, and hence their cost is also already accounted for. So we are left with analyzing the cost of stabbing the rectangles which are intersected by the lines along which we apply the line operations. We claim that for a random offset a, this cost is $O(\varepsilon \cdot \mathrm{OPT})$, which gives us the required result.

Let us first consider any line operation of level j that is applied to a horizontal line ℓ. This operation would create 2 cells of width at most ε^{j-2}, one of which either contains ε^{-3} endpoints of segments (horizontal or vertical) from OPT of level j; or contains at least one vertical segment from OPT of level $j' < j - 2$, i.e., the cost of the segments from OPT with at least one endpoint in this cell is at least $\varepsilon^{-3} \cdot \varepsilon^{j} = \varepsilon^{j-3}$. Since a segment of width ε^{j-2} (width of cell) is sufficient to stab all rectangles stabbed by ℓ, we see that this horizontal line takes only ε times the cost of the segments in OPT with at least one endpoint in the cell. We charge the cost of this horizontal segment to these corresponding endpoints. Since each such segment in OPT of level j can be charged at most twice, by summing over all horizontal line operations over all levels we get that the cost of such line operations is at most $2\varepsilon \cdot \mathrm{OPT}$.

Now, let us consider the line operations applied to vertical grid lines. We wish to bound the cost of stabbing all the rectangles intersected or *close to* grid lines (will be formally defined shortly), over all levels j. This can also be stated as bounding the cost, over all levels j, of line segments in level j of OPT (call this set OPT_j) intersected or close to grid lines of level j or smaller. For a horizontal segment $\ell \in \mathrm{OPT}_j$, let I_ℓ be the indicator variable representing the event that a grid line of level j or smaller intersects ℓ ($I_\ell = 0$ for vertical segments). Since $|\ell| \leq \varepsilon^{j-1}$, if we take a random offset a, we obtain that $\mathbb{E}[I_\ell] \leq \varepsilon^{j-1}/\varepsilon^{j-2} = \varepsilon$. For a vertical segment $\ell \in \mathrm{OPT}_j$, let J_ℓ be the indicator variable representing the event that a grid line of level j or smaller intersect the rectangle stabbed by ℓ ($J_\ell = 0$ for horizontal segments). Since for $j > 0$, $|\ell| \leq \varepsilon^{j-1}$, we know that for the rectangle stabbed by ℓ, the dimensions satisfy $w_i \leq h_i \leq \varepsilon^{j-1}$. This means that to stab such a rectangle, ℓ has to lie *close to*, i.e., within $\pm\varepsilon^{j-1}$ of the vertical grid line. So for a random offset a and level $j > 0$ we obtain that $\mathbb{E}[J_\ell] \leq 2\varepsilon^{j-1}/\varepsilon^{j-2} = 2\varepsilon$. For level 0, we note that even though the vertical segments can be very long, the maximum width of a rectangle is at most 1. So ℓ has to lie within ± 1 of the grid line, giving us: $\mathbb{E}[J_\ell] \leq 2/\varepsilon^{-2} = 2\varepsilon^2 \leq 2\varepsilon$. With the expectations computed above, we can upper bound the expected cost of segments in OPT intersected by vertical line operations as:

$$\mathbb{E}\left[\sum_j \sum_{\ell \in \mathrm{OPT}_j} (I_\ell + J_\ell) \cdot |\ell|\right] = \sum_j \sum_{\ell \in \mathrm{OPT}_j} |\ell| \cdot (\mathbb{E}[I_\ell] + \mathbb{E}[J_\ell])$$
$$\leq \sum_j \sum_{\ell \in \mathrm{OPT}_j} |\ell| \cdot (\varepsilon + 2\varepsilon) = 3\varepsilon \cdot \mathrm{OPT}$$

Now, by using the α-approximation algorithm for stabbing from [5], where α is a constant, the solution returned by our algorithm takes an additional cost of $3\alpha \cdot \varepsilon \cdot \mathrm{OPT}$. □

This gives our main theorem.

Theorem 1. *There is a $(1+\varepsilon)$-approximation algorithm for the general rectangle stabbing problem with a running time of $(n/\varepsilon)^{O(1/\varepsilon^3)}$, assuming that $h_i \geq w_i$ for each rectangle $R_i \in \mathcal{R}$.*

Theorem 1 has some direct implications. First, it yields a PTAS for the general square stabbing problem.

Corollary 1. *There is a PTAS for the general square stabbing problem.*

Also, it yields a $(2+\varepsilon)$-approximation algorithm for the general rectangle stabbing problem for arbitrary rectangles: we can simply split the input into rectangles R_i for which $h_i \geq w_i$ holds, and those for which $h_i < w_i$ holds, and output the union of these two solutions.

Corollary 2. *There is a $(2+\varepsilon)$-approximation algorithm for the general rectangle stabbing problem with a running time of $(n/\varepsilon)^{O(1/\varepsilon^3)}$.*

Finally, it yields a PTAS for the horizontal rectangle stabbing problem: we can take the input of that problem and stretch all input rectangles vertically such that it is always very costly to stab any rectangle vertically (so in particular our $(1+\varepsilon)$-approximate solution would never do this). Then we apply the algorithm due to Theorem 1.

Corollary 3. *There is a $(1+\varepsilon)$-approximation algorithm for the horizontal rectangle stabbing problem with a running time of $(n/\varepsilon)^{O(1/\varepsilon^3)}$.*

3 δ-Large Rectangles

We now consider the case of δ-large rectangles for some given constant δ, i.e., where for each input rectangle R_i we assume that $w_i \leq 1$ and $h_i \leq 1$ and additionally $w_i \geq \delta$ or $h_i \geq \delta$. For this case we again give a PTAS in which we use our algorithm due to Theorem 1 as a subroutine.

First, by losing only a factor of $1+\varepsilon$, we divide the instance into independent subproblems which are disjoint rectangular cells. For each cell C_i, we denote by $\mathrm{OPT}(C_i)$ the cells from OPT that are contained in C_i and our routine ensures that $c(\mathrm{OPT}(C_i)) \leq O(1/\varepsilon^3)$. Then for each cell C_i, the number of segments in

$\mathrm{OPT}(C_i)$ with length longer than δ is bounded by $O(1/\delta\varepsilon^3)$. We guess them in polynomial time. Now, the remaining segments in OPT are all of length smaller than δ, and hence they can stab a rectangle only along its shorter dimension. Hence, we can divide the remaining rectangles into two disjoint sets, one with $h_i \geq w_i$ and the other with $w_i > h_i$, and use Theorem 1 to get a $1+\varepsilon$ approximation of the remaining problem.

Theorem 2. *For* HV-Stabbing *with δ-large rectangles, there is a $(1+\varepsilon)$-approximation algorithm with a running time of $(n/\varepsilon)^{O(1/\delta\varepsilon^3)}$.*

4 Conclusion

In this paper, we have settled the Stabbing problem by giving a PTAS for it, and also give a $(2+\varepsilon)$-approximate solution for the HV-Stabbing problem and PTASs for some related special cases of these problems. It is not immediately clear whether these techniques could be extended to obtain a PTAS for the HV-Stabbing problem, or even if such a PTAS exists or not, since the question of the APX-hardness of HV-Stabbing is still open.

Acknowledgement. Arindam Khan was partly supported by Pratiksha Trust Young Investigator Award, Google India Research Award, and Google ExploreCS Award. Andreas Wiese was partially supported by the FONDECYT Regular grant 1200173.

References

1. Adamaszek, A., Wiese, A.: Approximation schemes for maximum weight independent set of rectangles. In: FOCS, pp. 400–409 (2013)
2. Aronov, B., Ezra, E., Sharir, M.: Small-size \eps-nets for axis-parallel rectangles and boxes. SIAM J. Comput. **39**(7), 3248–3282 (2010)
3. Becchetti, L., Marchetti-Spaccamela, A., Vitaletti, A., Korteweg, P., Skutella, M., Stougie, L.: Latency-constrained aggregation in sensor networks. ACM Trans. Algorithms **6**(1), 1–20 (2009)
4. Brönnimann, H., Goodrich, M.T.: Almost optimal set covers in finite VC-dimension. Discret. Comput. Geom. **14**(4), 463–479 (1995). https://doi.org/10.1007/BF02570718
5. Chan, T.M., van Dijk, T.C., Fleszar, K., Spoerhase, J., Wolff, A.: Stabbing rectangles by line segments - how decomposition reduces the shallow-cell complexity. In: Hsu, W., Lee, D., Liao, C. (eds.) ISAAC, vol. 123, pp. 61:1–61:13 (2018)
6. Chan, T.M., Grant, E.: Exact algorithms and APX-hardness results for geometric packing and covering problems. Comput. Geom. **47**(2), 112–124 (2014). https://doi.org/10.1016/j.comgeo.2012.04.001
7. Chan, T.M., Grant, E., Könemann, J., Sharpe, M.: Weighted capacitated, priority, and geometric set cover via improved quasi-uniform sampling. In: Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1576–1585. SIAM (2012)
8. Chan, T.M., Grant, E., Könemann, J., Sharpe, M.: Weighted capacitated, priority, and geometric set cover via improved quasi-uniform sampling. In: Rabani, Y. (ed.) SODA, pp. 1576–1585 (2012)

9. Chvatal, V.: A greedy heuristic for the set-covering problem. Math. Oper. Res. **4**(3), 233–235 (1979)
10. Das, A., Fleszar, K., Kobourov, S., Spoerhase, J., Veeramoni, S., Wolff, A.: Approximating the generalized minimum manhattan network problem. Algorithmica **80**(4), 1170–1190 (2018)
11. Eisenbrand, F., Gallato, M., Svensson, O., Venzin, M.: A QPTAS for stabbing rectangles. CoRR abs/2107.06571 (2021). https://arxiv.org/abs/2107.06571
12. Finke, G., Jost, V., Queyranne, M., Sebő, A.: Batch processing with interval graph compatibilities between tasks. Discret. Appl. Math. **156**(5), 556–568 (2008)
13. Gaur, D.R., Ibaraki, T., Krishnamurti, R.: Constant ratio approximation algorithms for the rectangle stabbing problem and the rectilinear partitioning problem. J. Algorithms **43**(1), 138–152 (2002)
14. Hochbaum, D.S., Maass, W.: Approximation schemes for covering and packing problems in image processing and VLSI. J. ACM **32**(1), 130–136 (1985)
15. Khan, A., Pittu, M.R.: On guillotine separability of squares and rectangles. In: APPROX/RANDOM. vol. 176, pp. 47:1–47:22 (2020)
16. Khan, A., Subramanian, A., Wiese, A.: A PTAS for the horizontal rectangle stabbing problem. CoRR abs/2111.05197 (2021). https://arxiv.org/abs/2111.05197
17. Kovaleva, S., Spieksma, F.: Approximation algorithms for rectangle stabbing and interval stabbing problems. SIAM J. Discret. Math. **20**(3), 748–768 (2006)
18. Mustafa, N.H., Raman, R., Ray, S.: Quasi-polynomial time approximation scheme for weighted geometric set cover on pseudodisks and halfspaces. SIAM J. Comput. **44**(6), 1650–1669 (2015). https://doi.org/10.1137/14099317X
19. Varadarajan, K.R.: Weighted geometric set cover via quasi-uniform sampling. In: STOC, pp. 641–648 (2010)