



# A PTAS for the horizontal rectangle stabbing problem

Arindam Khan<sup>1</sup> · Aditya Subramanian<sup>1</sup> · Andreas Wiese<sup>2</sup>

Received: 12 July 2022 / Accepted: 3 June 2024 / Published online: 13 June 2024

© Springer-Verlag GmbH Germany, part of Springer Nature and Mathematical Optimization Society 2024

## Abstract

We study rectangle stabbing problems in which we are given  $n$  axis-aligned rectangles in the plane that we want to *stab*, that is, 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 the *horizontal–vertical stabbing problem* (HV-STABBING), the goal is to find a set of rectilinear (that is, either vertical or horizontal) line segments of minimum total length such that all rectangles are stabbed. Both variants are NP-hard. Chan et al. (ISAAC, 2018) initiated the study of these problems by providing constant approximation algorithms. Recently, Eisenbrand et al. (A QPTAS for stabbing rectangles, 2021) have presented a QPTAS and a polynomial-time 8-approximation algorithm for STABBING, but it was 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 PTASs 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  $\delta$ -large, that is, have at least one edge whose length is at least  $\delta$ , while all edge lengths are at most 1. Our result also implies improved approximations for other problems such as *generalized minimum Manhattan network*.

A preliminary version of this work appeared in the 23rd Conference on Integer Programming and Combinatorial Optimization, 2022 [33]. Arindam Khan was partly supported by Pratiksha Trust Young Investigator Award, Google India Research Award, SERB Core Research Grant (CRG/2022/001176) on “Optimization under Intractability and Uncertainty”, and Walmart Center for Tech Excellence. Aditya Subramanian is partly supported by Kotak IISc AI-ML Center (KIAC) PhD fellowship, and by the Walmart Center for Tech Excellence.

✉ Aditya Subramanian  
adityasubram@iisc.ac.in

Arindam Khan  
arindamkhan@iisc.ac.in

Andreas Wiese  
andreas.wiese@tum.de

<sup>1</sup> Indian Institute of Science, Bengaluru, India

<sup>2</sup> Technical University of Munich, Munich, Germany

**Keywords** Approximation algorithms · Geometric optimization · Rectangles · Stabbing

**Mathematics Subject Classification** 68W25 · 68W20 · 68Q25 · 52C15 · 90C27

## 1 Introduction

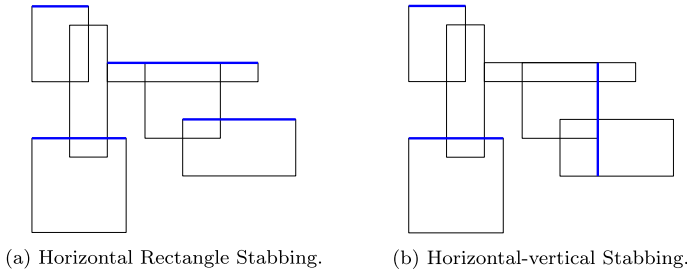
Rectangle stabbing problems are natural geometric optimization problems. Here, we are given a set of  $n$  axis-parallel rectangles  $\mathcal{R}$  in the two-dimensional plane. For each rectangle  $R_i \in \mathcal{R}$ , we are given points  $(x_1^{(i)}, y_1^{(i)})$ ,  $(x_2^{(i)}, y_2^{(i)}) \in \mathbb{R}^2$  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. We assume that  $w_i > 0$  and  $h_i > 0$  for each rectangle  $R_i \in \mathcal{R}$ . Our goal is to compute a set of line segments  $\mathcal{L}$  that stab all input rectangles. We call a rectangle  $R_i \in \mathcal{R}$  *stabbed* if a segment  $\ell \in \mathcal{L}$  intersects both edges of  $R_i$  that are perpendicular to  $\ell$ . 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 *horizontal-vertical 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 (see Fig. 1). The *horizontal-vertical 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 [6, 8, 14].

Note that STABBING and HV-STABBING are special cases of weighted geometric set cover, 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 [13] for HV-STABBING and STABBING.

Chan et al. [8] initiated the study of STABBING. They proved STABBING to be NP-hard via a reduction from planar vertex cover. Also, they presented a constant<sup>1</sup> factor approximation using *decomposition techniques* and the *quasi-uniform sampling* method [39] for weighted geometric set cover. In particular, they showed that HV-STABBING instances can be decomposed into two disjoint *laminar* set cover instances of small shallow cell complexity for which the *quasi-uniform sampling* yields a constant approximation using techniques introduced by Chan et al. [10].

Recently, Eisenbrand et al. [16] presented a quasi-polynomial time approximation scheme (QPTAS) for STABBING. This shows that STABBING is not APX-hard unless  $\text{NP} \subseteq \text{DTIME}(2^{\text{poly} \log n})$ . The QPTAS relies on *the shifting technique* by Hochbaum and Maass [24], 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.

<sup>1</sup> The constant is not explicitly stated, and it depends on a not explicitly stated constant in a work by Chan et al. [10].



**Fig. 1** A solution to an instance of STABBING and HV-STABBING

They also gave a polynomial time exact algorithm, based on dynamic programming, for laminar instances of STABBING (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 PTASs for several special cases: when all input rectangles are squares, more generally when for each rectangle its width is at most its height, and finally for  $\delta$ -large rectangles, that is, 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).

To obtain an algorithm for STABBING we first give a PTAS to solve the HV-STABBING problem, when the input consists of rectangles whose width is at most their height. The high level idea is 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 the earlier QPTAS [16], 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 advanced 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 afterward these line segments *vanish*. This allows us to bound the number of arising subproblems (and hence of the DP cells) by a polynomial.

The algorithm above directly implies a PTAS for SQUARE- STABBING and can also be used to obtain a PTAS for STABBING, and a polynomial time  $(2 + \varepsilon)$ -approximation for HV- STABBING—thus improving on the constant approximation by Chan et al. [8].

Then, we extend our techniques above to the setting of  $\delta$ -large rectangles of HV- STABBING. This is an important subclass of rectangles and is well-studied for other geometric problems [1, 30]. To this end, we first reduce the problem to the setting in which all input rectangles are contained in rectangular boxes, and the instance corresponding to any such box 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  $\delta$ -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 previously known algorithm [14].

The paper is organized as follows: the PTAS for STABBING is given in Sect. 2, which is followed by the PTAS for  $\delta$ -large rectangles in Sect. 3. Finally in Sect. 4 we present our results for the generalized minimum Manhattan network problem.

## 1.2 Further related work

Finke et al. [17] 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. [14] 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<sup>2</sup> problem, a special case of 2D-GMMN, and a  $(6 + \varepsilon)(\log n)$ -approximation algorithm for 2-D GMMN.

Gaur et al. [22] 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 [34] 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. In a seminal paper, Brönnimann and Goodrich [7] gave an  $O(d \log(dOPT))$ -approximation algorithm 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  $\varepsilon$ -nets, Aronov et al. [3] gave an  $O(\log \log OPT)$ -approximation algorithm for hitting set for axis-parallel rectangles. Later, Varadarajan

<sup>2</sup> Refer to Sect. 4 for formal problem definitions.

[39] developed quasi-uniform sampling and provided sub-logarithmic approximation for weighted set cover where sets are fat triangles or disks. Chan et al. [10] generalized this to any set system with low *shallow cell complexity*. Bansal and Pruhs [5] reduced several scheduling problems to a particular geometric set cover problem for anchored rectangles and obtained  $O(1)$ -approximation using quasi-uniform sampling. Afterward, Chan, and Grant [9] and Mustafa et al. [38] have settled the APX-hardness statuses of all natural weighted geometric set cover problems. Online and dynamic variants of geometric set cover have also recently received some attention [2, 11, 28].

Another related problem is the maximum independent set of rectangles (MISR). Adamaszek and Wiese [1] gave a QPTAS for MISR using *balanced cuts* with small complexity that intersect only a few rectangles in the solution. Recently, Mitchell [35] obtained the first polynomial time constant approximation algorithm for the problem, followed by a  $(2 + \varepsilon)$ -approximation by Gálvez et al. [21].

Many other rectangle packing problems are also well-studied [12], such as geometric knapsack [19, 20, 25, 29], geometric bin packing [4, 31, 32], strip packing [15, 18, 23, 27], storage allocation problem [26, 36, 37], etc.

## 2 Dynamic program

We present a dynamic program that computes a  $(1 + \varepsilon)$ -approximation to HV-STABBING, for any given constant  $\varepsilon > 0$ , for the case when for each input rectangle  $R_i \in \mathcal{R}$  its width is at most its height, that is,  $w_i \leq h_i$ . 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. Also, we will use it later as a subroutine to obtain a  $(2 + \varepsilon)$ -approximation for HV-STABBING and a PTAS for the setting of  $\delta$ -large rectangles of HV-STABBING.

For a line segment  $\ell$ , we use the notation  $|\ell|$  to represent its length, and for a set of segments  $\mathcal{L}$ , we use the 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  $c(\text{OPT})$ , that is, the cost of the optimal solution. Similarly, the term SOL is interchangeably used to represent a solution returned by the algorithm, and its cost.

### 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 *discrete* if  $x$  is an integral multiple of  $\varepsilon^d$ , where we define  $d \in \mathbb{N}$  such that  $\varepsilon^3/n < \varepsilon^d \leq \varepsilon^2/n$ ; note that hence  $d$  is unique. A point is called *discrete* if its  $x$  and  $y$  coordinates are discrete, and similarly a segment or a rectangle is said to be *discrete* if both of its end points, or both of its diagonally opposite corners are discrete. Note that solving a scaled version of the problem is equivalent to solving the problem, so we uniformly scale the width and

height of each rectangle in  $\mathcal{R}$  so that the largest width of a rectangle is  $1 - 2\varepsilon$ . Let this be the instance under consideration, with  $\text{OPT}$  being the optimal solution for it.

**Lemma 1** *For any positive constant  $\varepsilon < 1/3$ , in polynomial time we can compute a new instance  $\mathcal{R}'$  of HV-STABBING, which satisfies*

- (i) for each  $R_i \in \mathcal{R}'$ ,  $\varepsilon/n < w_i \leq 1$ ,
- (ii) for each  $R_i \in \mathcal{R}'$ ,  $x_1^{(i)}, x_2^{(i)}$  are discrete and within  $[0, n]$ ,
- (iii) for each  $R_i \in \mathcal{R}'$ ,  $y_1^{(i)}, y_2^{(i)}$  are discrete and within  $[0, 4n^2]$ ,
- (iv) The instance admits a valid solution of cost at most  $(1 + O(\varepsilon)) \cdot \text{OPT}$ , in which each horizontal line segment has a width of at most  $1/\varepsilon$ .

Further, a solution to this instance can be modified in polynomial time, to obtain a valid solution to the original instance of cost  $(1 + O(\varepsilon)) \cdot c(\text{OPT})$ .

**Proof** See Fig. 2 for a visual representation of these properties. We start with  $\mathcal{R}'' = \mathcal{R}$ , and remove all rectangles with width at most  $\varepsilon/n$  from  $\mathcal{R}''$ . So for each rectangle in  $\mathcal{R}''$  we have  $\varepsilon/n < w_i \leq 1 - 2\varepsilon$ .

Now we incorporate a *discretization* step. Since each rectangle  $R_i \in \mathcal{R}''$  has width  $w_i \leq 1 - 2\varepsilon$ , the instance either has some  $x$ -coordinate that is not covered by any rectangle—in which case we can split the instance into smaller subproblems around this  $x$ -coordinate—or the total width of the instance is less than  $n(1 - 2\varepsilon)$ . Hence, all the  $x$ -coordinates of the rectangles can be assumed to be between 0 and  $n(1 - 2\varepsilon)$ . Further, we extend the rectangles on both sides to make their  $x$ -coordinates align with the next nearest multiple of  $\varepsilon^d$ . This ensures Property (ii). This step involves an extension by at most  $2\varepsilon^d < 2\varepsilon/n$  to the width of every rectangle, which hence bounds the total cost of an optimal solution  $\text{OPT}''$  to  $\mathcal{R}''$ , by at most  $(1 + 2\varepsilon) \cdot \text{OPT}$ . Thus,  $\varepsilon/n < w_i \leq 1$  which ensures Property (i).

Next, we show Property (iii) using a *stretching* step. Since we have  $h_i \geq w_i$  for the input instance, the heights of the rectangles could still be arbitrarily large. But after scaling the rectangles, as  $w_i \leq 1$ , we know that  $c(\text{OPT}'')$  is clearly less than  $n$ . So, there cannot be any vertical line segment in  $\text{OPT}''$  which is longer than  $n$ . We exploit this fact to scale the heights of the rectangles in  $\mathcal{R}''$ , to obtain a set  $\mathcal{R}'$  as follows: of the at most  $2n$  distinct  $y$ -coordinates of the rectangles, if the distance between any consecutive coordinates is more than  $2n$ , we shrink it down to the nearest multiple of  $\varepsilon^d$  which is less than  $2n$ . Since all the  $y$ -coordinates are separated by at most  $2n$ , they can be assumed to be within  $[0, 4n^2]$ . This operation does not affect the cost and validity of  $\text{OPT}''$ , since given any such solution, there cannot be a segment that spans one of these stretched vertical sections (as it would add cost at least  $2n - \varepsilon^d$ ).

Now, similar to the  $x$ -coordinates, we extend the rectangles by at most  $\varepsilon^d$  on each side along the  $y$ -coordinate, which gives us that an optimal solution  $\text{OPT}'$  to  $\mathcal{R}'$ , is of cost at most  $(1 + 2\varepsilon) \cdot \text{OPT}''$ .

We now consider the final property. We have shown the cost of  $\text{OPT}'$  to exceed the cost of the original optimal solution  $\text{OPT}$  by at most a factor of  $1 + O(\varepsilon)$ . Now, consider any horizontal segment  $\ell \in \text{OPT}'$  that is longer than  $1/\varepsilon$ . From the left endpoint, we divide the segment into consecutive smaller segments of length  $1/\varepsilon - 2$  each, with one potential last piece being smaller than  $1/\varepsilon - 2$ . Now, for each smaller segment,

we extend it on both sides in such a way that it completely stabs the rectangles that it intersects (that is, that it intersects before the extension). Since the maximum width of a rectangle is 1, we extend each such segment by at most 2 units. We denote by  $|\ell|$  the length of  $\ell$  and conclude that we increase the length of  $\ell$  by at most a factor of

$$\begin{aligned} \left( |\ell| + \left\lceil \frac{|\ell|}{1/\varepsilon - 2} \right\rceil \cdot 2 \right) \cdot \frac{1}{|\ell|} &\leq 1 + \frac{2}{|\ell|} \cdot \left\lceil \frac{\varepsilon|\ell|}{1 - 2\varepsilon} \right\rceil \\ &\leq 1 + \frac{2}{|\ell|} \cdot \left( 1 + \frac{\varepsilon|\ell|}{1 - 2\varepsilon} \right) \\ &\leq 1 + 2\varepsilon + 2\varepsilon \cdot \frac{1}{1 - 2\varepsilon} \quad (\text{since } |\ell| > 1/\varepsilon) \\ &\leq 1 + 2\varepsilon + 2\varepsilon \cdot \frac{1}{1 - 2/3} \quad (\text{since } \varepsilon < 1/3) \\ &\leq 1 + 8\varepsilon. \end{aligned}$$

We note here that we have only made the rectangles larger, and constructed a solution of cost  $(1 + O(\varepsilon)) \cdot \text{OPT}$  that stabs these rectangles. This solution hence already stabs all rectangles of width larger than  $\varepsilon/n$  in the original instance  $\mathcal{R}$ . But, the rectangles that have been removed from  $\mathcal{R}$  can be stabbed greedily using segments of total length at most  $\varepsilon$ . Since the width of the widest rectangle is at least  $1 - 2\varepsilon$ , we also get that  $c(\text{OPT}') \geq c(\text{OPT}) \geq 1 - 2\varepsilon$ . So, if  $\varepsilon < 1/3$ , the cost of including these greedily picked segments increases the cost of a solution by a factor of at most

$$\frac{\text{OPT}' + \varepsilon}{\text{OPT}'} \leq 1 + \frac{\varepsilon}{1 - 2\varepsilon} \leq 1 + \varepsilon \cdot \frac{1}{1 - 2(1/3)} = 1 + 3 \cdot \varepsilon.$$

This completes the proof of the lemma. □

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  $\text{OPT}$ , we are abusing notation by referring to the valid solution to the set of discretized rectangles  $\mathcal{R}'$  obtained from Lemma 1, which is a  $(1 + O(\varepsilon))$ -approximation of the optimal solution to the input instance.

### 2.2 Description of the dynamic program

Our algorithm is based on a dynamic program. 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 discrete coordinates (that is not necessarily equal to an input rectangle in  $\mathcal{R}$ ).
- a set  $\mathcal{L}$  of at most  $3(1/\varepsilon)^3$  axis-aligned line segments, such that for each  $\ell \in \mathcal{L}$  we have that  $\ell \subseteq S$ , and  $\ell$  is discrete.

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  $\mathcal{L}$ . There-

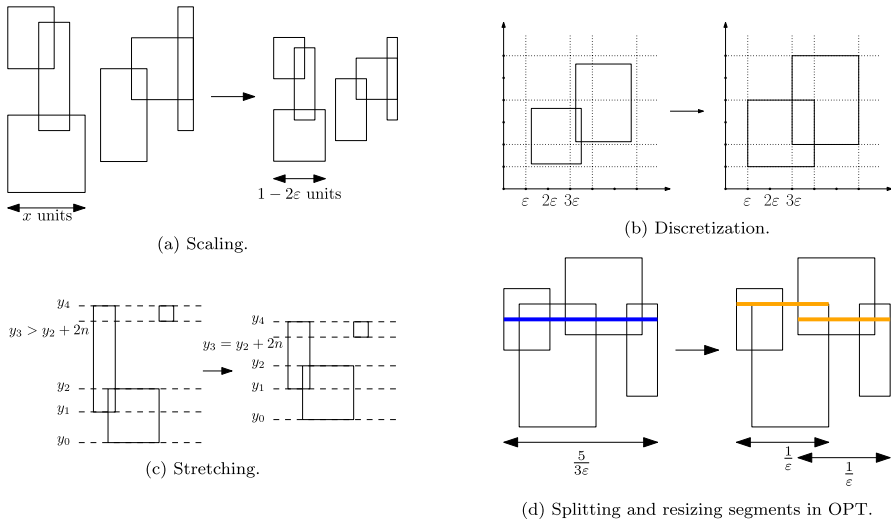


Fig. 2 Pre-processing steps

fore, the DP stores a near-optimal solution  $SOL(S, \mathcal{L})$  in the cell  $DP(S, \mathcal{L})$  such that  $SOL(S, \mathcal{L}) \cup \mathcal{L}$  stabs all rectangles in  $\mathcal{R}$  that are contained in  $S$ .

Given a DP cell  $DP(S, \mathcal{L})$ , our dynamic program 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 the solution  $SOL(S, \mathcal{L}) := \emptyset$  for the cell  $DP(S, \mathcal{L})$  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. Note that these components need not be closed rectangles). In this case, we define  $SOL(S, \mathcal{L}) := SOL(S_1, \mathcal{L} \cap S_1) \cup SOL(S_2, \mathcal{L} \cap S_2) \cup \{\ell\}$ , where for any set of line segments  $\mathcal{L}'$  and any rectangle  $S'$  we define  $\mathcal{L}' \cap S' := \{\ell' \cap S' \mid (\ell' \in \mathcal{L}') \wedge (\ell' \cap S' \neq \emptyset)\}$ . In case 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 a *trivial operation*.

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

1. *Add operation:* Consider each set  $\mathcal{L}'$  of line segments with discrete coordinates such that  $|\mathcal{L} \cup \mathcal{L}'| \leq 3\epsilon^{-3}$  and each  $\ell \in \mathcal{L}'$  is axis-aligned and contained in  $S$ . For each such set  $\mathcal{L}'$  we define the solution  $\mathcal{L}' \cup SOL(S, \mathcal{L} \cup \mathcal{L}')$  as a candidate solution.
2. *Line operation:* Consider each axis-aligned line  $\ell$  passing through some discrete point 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  $\ell$ . For the line  $\ell$  we do the following:
  - a. Compute an  $O(1)$ -approximate solution  $\mathcal{L}(\mathcal{R}_\ell)$  for the rectangles in  $\mathcal{R}_\ell$  using a polynomial time algorithm [8].
  - b. Produce the candidate solution  $\mathcal{L}(\mathcal{R}_\ell) \cup SOL(S_1, \mathcal{L} \cap S_1) \cup SOL(S_2, \mathcal{L} \cap S_2)$ .



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

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

We remark that instead of using the existing  $O(1)$ -approximation algorithm [8] 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  $\ell$ . However, for our purposes an  $O(1)$ -approximate solution is good enough.

### 2.3 DP-decision trees

We want to show that the DP above computes a  $(1 + \epsilon)$ -approximate solution. For this, we define a tree  $T$  in which each node corresponds to a cell  $DP(S, \mathcal{L})$  of the DP and a corresponding solution  $\overline{SOL}(S, \mathcal{L})$  to this cell. The root node of  $T$  corresponds to the cell  $DP([0, n] \times [0, 4n^2], \emptyset)$ . Intuitively, the outgoing edges (when considered in root to leaf direction) from a node represent application of some DP operation (as described in the section above) to the parent node, and the child nodes are the subproblems obtained due to application of said operation. The corresponding solution stored at a node is the solution obtained by choosing certain operations, corresponding to the edges in the subtree at that node. This structure now allows us to represent a valid sequence of operations by the algorithm as a tree. 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,  $c(\overline{SOL}([0, n] \times [0, 4n^2], \emptyset))$  of any such tree. Formally, we require  $T$  to satisfy the following properties:

1. We require that a node  $v$  is a leaf if and only if for the corresponding DP cell  $DP(S, \mathcal{L})$  the DP directly defined that  $SOL(S, \mathcal{L}) = \emptyset$ . This is because all rectangles in  $\mathcal{R}$  that are contained in  $S$  are already stabbed by the segments in  $\mathcal{L}$ .
2. If a node  $v$  for a DP cell  $DP(S, \mathcal{L})$  has exactly one child, then we require that we reduce the problem for  $DP(S, \mathcal{L})$  to the child by applying the add operation, that is, there is a set  $\mathcal{L}'$  of axis-aligned line segments with discrete coordinates such that  $|\mathcal{L}| \cup |\mathcal{L}'| \leq 3(1/\epsilon)^3$ , the child node of  $v$  corresponds to the cell  $DP(S, \mathcal{L} \cup \mathcal{L}')$ , and  $\overline{SOL}(S, \mathcal{L}) = \overline{SOL}(S, \mathcal{L} \cup \mathcal{L}') \cup \mathcal{L}'$ .
3. Similarly, if a node  $v$  has two children then we require that we can reduce the problem of  $DP(S, \mathcal{L})$  to these two children by applying the trivial operation or the line operation. Formally, assume that the child nodes correspond to the subproblems  $DP(S_1, \mathcal{L}_1)$  and  $DP(S_2, \mathcal{L}_2)$ .
  - 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  $\overline{SOL}(S, \mathcal{L}) = \overline{SOL}(S_1, S_1 \cap \mathcal{L}) \cup \overline{SOL}(S_2, S_2 \cap \mathcal{L}) \cup \{\ell\}$ .
  - If no such segment exists, then the applied operation was a line operation along the line  $\ell$ , such that  $S_1 \cup S_2 \cup \ell = S$ ,  $\mathcal{L}_1 \cup \mathcal{L}_2 = \mathcal{L}$ , and  $\overline{SOL}(S, \mathcal{L}) =$

$\overline{\text{SOL}}(S_1, S_1 \cap \mathcal{L}) \cup \overline{\text{SOL}}(S_2, S_2 \cap \mathcal{L}) \cup \mathcal{L}(\mathcal{R}_\ell)$ ; where  $\mathcal{L}(\mathcal{R}_\ell)$  is an  $O(1)$ -approximate solution to the set of rectangles intersected by  $\ell$ .

We call a tree  $T$  with these properties a *DP-decision tree*.

**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  $(n/\varepsilon)^{O(1/\varepsilon^3)}$ .*

**Proof** We know that the DP always picks a solution that corresponds to the DP-decision tree with the minimum cost. Since  $T'$  is a valid DP-decision tree, the solution picked by the DP is of cost at most  $c(\overline{\text{SOL}}([0, n] \times [0, 4n^2], \emptyset))$ .

Now let us consider the running time of the algorithm. Since a DP problem is defined on a discrete rectangular cell, there are at most

$$\frac{n}{\varepsilon^d} \times \frac{4n^2}{\varepsilon^d} = \frac{4n^3}{\varepsilon^{2d}} \leq \frac{4n^5}{\varepsilon^6} \quad (\text{since } \varepsilon^3/n < \varepsilon^d \leq \varepsilon^2/n)$$

possibilities for a corner vertex of a rectangle, and hence  $\binom{4n^5/\varepsilon^6}{2} = O(n^{10}/\varepsilon^{12})$  possible rectangles.

Similarly, the subproblem definition also includes a set of segments  $\mathcal{L}$  of size at most  $3\varepsilon^{-3}$ . Since the segments are discrete, we can count the number of vertical segments by picking two points (corresponding to the starting and ending points) from the available discrete points on a vertical line, and then fixing its  $x$ -coordinate. We repeat a similar process for horizontal lines giving us,

$$\binom{4n^2/\varepsilon^d}{2} \times \frac{n}{\varepsilon^d} + \binom{n/\varepsilon^d}{2} \times \frac{4n^2}{\varepsilon^d} \leq \binom{4n^3/\varepsilon^3}{2} \times \frac{n^2}{\varepsilon^3} + \binom{n^2/\varepsilon^3}{2} \times \frac{4n^3}{\varepsilon^3}$$

that is,  $O(n^8/\varepsilon^9)$  possible segments.<sup>3</sup> So there are at most  $(n/\varepsilon)^{O(1/\varepsilon^3)}$  subsets of segments  $\mathcal{L}$  of size at most  $3\varepsilon^{-3}$ , and at most  $(n/\varepsilon)^{O(1/\varepsilon^3)}$  valid DP cells.

For each DP cell, we have to consider all possible candidate solutions and select the minimum. There are at most  $n^2/\varepsilon^3 + 4n^3/\varepsilon^3 = O(n^3/\varepsilon^3)$  possible line operations and  $(n/\varepsilon)^{O(1/\varepsilon^3)}$  possible add operations. Note that if the DP performs a trivial operation, then there is no choice to make here, but the trivial operation is selected automatically.

Hence, the total number of possible operations for a given DP cell is  $(n/\varepsilon)^{O(1/\varepsilon^3)}$ . For each line operation we call the  $O(1)$ -approximation algorithm which also runs in polynomial time [8]. Since we have  $(n/\varepsilon)^{O(1/\varepsilon^3)}$  DP cells, our overall running time is bounded by  $(n/\varepsilon)^{O(1/\varepsilon^3)}$ . □

We now define a DP-decision tree for which  $c(\overline{\text{SOL}}(S, \mathcal{L})) \leq (1 + \varepsilon)\text{OPT}$ . Recall that  $1/\varepsilon \in \mathbb{N}$ . We start by defining a hierarchical grid of vertical lines. Let  $a \in [0, n]$  be a discrete random offset to be defined later. The grid lines have levels. For each

<sup>3</sup> Note that we allow all possible discrete segments. Hence, the simple bound of  $O(n^3)$  on the number of segments used in [16] is not valid.

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  $j + 1$ , and that for all  $j \leq d + 2$ , the grid lines of level  $j$  have discrete  $x$ -coordinates. We also note that any two consecutive lines of some level  $j$  are exactly  $\varepsilon^{j-2}$  units apart.

We say that a line segment  $\ell \in \text{OPT}$  is *of level  $j$*  if the length of  $\ell$  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  $j + 3$ , that is, if both of its  $x$ -coordinates are of the form  $a + k \cdot \varepsilon^{j+1}$ , for some  $k \in \mathbb{Z}$ . 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  $\varepsilon^{j+1}$ . This would be similar to the segment's end points lying on an (imaginary) horizontal grid line of level  $j + 3$ . Recall that by Lemma 1, each horizontal segment  $\ell \in \text{OPT}$  satisfies that  $\varepsilon/n < |\ell| \leq \varepsilon^{-1}$ . By our choice of  $d$  we have  $\varepsilon^{d-1} \leq \varepsilon/n < \varepsilon^{d-2}$  which implies  $\varepsilon^{d-1} < |\ell| \leq \varepsilon^{-1}$ . Since a segment is of level  $j$  if its length is in the range  $(\varepsilon^j, \varepsilon^{j-1}]$ , we can conclude that all segments in  $\text{OPT}$  belong to levels in the range  $\{0, \dots, d - 1\}$ . From this we can infer that any well-aligned horizontal segment is aligned to a vertical grid line of level at most  $d + 2$ , which as we noted earlier has discrete  $x$ -coordinates.

**Lemma 3** *There exists a solution  $\text{SOL}'$  to the problem that consists only of well-aligned segments, and the cost of this solution is  $(1 + O(\varepsilon)) \cdot \text{OPT}$ .*

**Proof** We show the existence of such a solution by considering an optimal solution, and extending its segments on both sides, such that they get well-aligned. A segment  $\ell \in \text{OPT}$  in some level  $j$  will be of length in  $(\varepsilon^j, \varepsilon^{j-1}]$ . To align it to a grid line of level  $j + 3$  we would need to extend it by at most  $\varepsilon^{j+1}$  on each side. The new segment  $\ell'$  thus obtained is of length

$$|\ell'| \leq |\ell| + 2\varepsilon^{j+1} \leq |\ell| \left(1 + \frac{2\varepsilon^{j+1}}{|\ell|}\right) < |\ell| \left(1 + \frac{2\varepsilon^{j+1}}{\varepsilon^j}\right) = |\ell| \cdot (1 + 2\varepsilon).$$

Therefore, the sum of weights over all the well-aligned segments is

$$\sum_{\ell \in \text{OPT}} |\ell'| \leq \sum_{\ell \in \text{OPT}} |\ell| \cdot (1 + 2\varepsilon) = (1 + 2\varepsilon) \cdot \text{OPT}.$$

□

We note that since the input rectangles have been discretized, any vertical (respectively horizontal) segment in the well aligned solution  $\text{SOL}'$  can be shifted vertically (respectively horizontally) to the nearest grid line without changing the set of rectangles it intersects. This fact, in conjunction with their well-alignment (from Lemma 3) ensures that the end-points of segments in  $\text{SOL}'$  are discrete points. Since  $\text{SOL}'$  is only a  $(1 + O(\varepsilon))$  factor more in cost than  $\text{OPT}$ , it is a permissible solution when constructing a PTAS. We shall hence assume that there exists an optimal solution with only discrete segments that are also 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, that is, for a node  $v$  corresponding to the subproblem  $\text{DP}(S, \mathcal{L})$ , if we are applying the trivial (respectively line) operation along a segment (respectively line)  $\ell$ , then we add children corresponding to the DP subproblems  $\text{DP}(S_1, S_1 \cap \mathcal{L})$  and  $\text{DP}(S_2, S_2 \cap \mathcal{L})$ , where  $S_1$  and  $S_2$  are the connected components of  $S \setminus \ell$ . 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  $\text{DP}(S, \mathcal{L} \cup \mathcal{L}')$ .

*First level ( $j = 0$ ).* Consider the line operations corresponding to vertical grid lines of level 0, in some arbitrary order. We start by applying the first operation to the root  $\text{DP}([0, n] \times [0, 4n^2], \emptyset)$ . We then continue by applying the following operations, in sequence, to the appropriate child node obtained from the earlier operation. Consider one of the resulting subproblems  $\text{DP}(S, \emptyset)$ . Suppose that there are more than  $\varepsilon^{-3}$  line segments (horizontal or vertical) from  $\text{OPT}$  of level 0 inside  $S$ . We want to partition  $S$  into smaller rectangles, such that within each of these rectangles  $S'$  at most  $\varepsilon^{-3}$  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 horizontal and vertical line segments from  $\text{OPT}$  of level 0 inside  $S$ , take (the multiset of) their endpoints and order these endpoints non-decreasingly by their  $y$ -coordinates. Let  $p_1, p_2, \dots, p_k$  be these points in this order. For each  $k' \in \mathbb{N}$  with  $k'/\varepsilon^3 \leq k$ , we consider the point  $p_{k'/\varepsilon^3}$ . Let  $\ell'$  be the horizontal line that contains  $p_{k'/\varepsilon^3}$ . We apply the line operation to  $\ell'$ .

**Proposition 1** *Let  $\text{DP}(S', \emptyset)$  be one of the subproblems after applying the operations above. There are at most  $\varepsilon^{-3}$  line segments  $\mathcal{L}'$  (horizontal or vertical) from  $\text{OPT}$  of level 0 that have an endpoint inside  $S'$ .*

In each resulting subproblem  $\text{DP}(S', \emptyset)$ , for each vertical line segment  $\ell \in \text{OPT}$  that crosses  $S'$ , that is, such that  $S' \setminus \ell$  has two connected components, we apply the line operation for the line that contains  $\ell$ . (As earlier, we do the operations in arbitrary order, by applying a line operation to the appropriate child node obtained after previous operations.) In each subproblem  $\text{DP}(S'', \emptyset)$  obtained after this step, we apply the add operation to the line segments from  $\text{OPT}$  of level 0 that intersects  $S''$  (or to be more precise, their intersection with  $S''$ ), that is, to the set  $\mathcal{L}' := \{\ell \cap S'' \mid (\ell \in \text{OPT}) \wedge (\ell \cap S'' \neq \emptyset) \wedge (\ell \text{ is of level } 0)\}$ . Proposition 1 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  $j = 1, 2, 3, \dots$ . Assume by induction that for some  $j$  each leaf in the current tree  $T$  corresponds to a subproblem  $\text{DP}(S, \mathcal{L})$  such that for any line segment  $\ell \in \text{OPT}$  of level  $j' \in [j - 3, j]$  with  $\ell \cap S \neq \emptyset$ , we have  $\ell \cap S \in \mathcal{L}$ . (It is easy to see from the following construction that all such segments from  $\text{OPT}$  will be added to the subproblem. We shall formally show in Lemma 4, that segments from the smaller levels will in fact also get removed.) Consider one of these leaves and suppose that it corresponds to a subproblem  $\text{DP}(S, \mathcal{L})$ . We apply the line operation for each vertical line that corresponds to a (vertical) grid line of level  $j$  intersecting  $S$ .

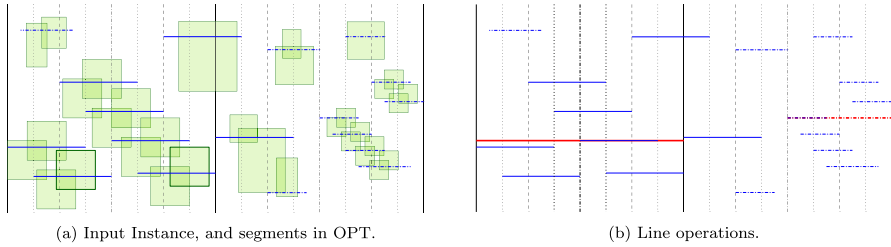


Fig. 3 Horizontal line operations

Consider a resulting subproblem  $DP(S', \mathcal{L})$ . Suppose that there are more than  $\varepsilon^{-3}$  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  $p_1, p_2, \dots, p_k$  be these points in this order. For each  $k' \in \mathbb{N}$  with  $k'/\varepsilon^3 \leq k$ , we consider the point  $p_{k'/\varepsilon^3}$  and apply the line operation for the horizontal line  $\ell'$  that contains  $p_{k'/\varepsilon^3}$ . If for a resulting subproblem  $DP(S'', \mathcal{L})$  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  $\ell$  operation for the horizontal line that contains  $p$ .

**Proposition 2** *Let  $DP(S'', \mathcal{L})$  be one of the subproblems after applying the operations above. There are at most  $\varepsilon^{-3}$  line segments  $\mathcal{L}'$  (horizontal or vertical) from OPT of level  $j$  that have an endpoint inside  $S''$ .*

Consider a resulting subproblem  $DP(S'', \mathcal{L})$ . For each line segment  $\ell \in \text{OPT}$  such that  $\ell$  crosses  $S''$ , that is,  $S'' \setminus \ell$  has two connected components, we apply the line operation to the line that contains  $\ell$ . In each subproblem  $DP(S''', \mathcal{L})$  obtained after this step, we intuitively apply the add operation to the line segments of level  $j$  that intersect  $S'''$ . Formally, we apply them to the line segments in the set  $\mathcal{L}' := \{\ell \cap S''' \mid (\ell \in \text{OPT}) \wedge (\ell \cap S''' \neq \emptyset) \wedge (\ell \text{ is of level } j)\}$ . We will prove later that  $|\mathcal{L}'|$  is sufficiently small such that  $|\mathcal{L} \cup \mathcal{L}'| \leq 3\varepsilon^{-3}$  and, hence, we are allowed to apply the add operation to  $|\mathcal{L}'|$ . Also, we will show that after this operation, the resulting set of line segments  $\mathcal{L} \cup \mathcal{L}'$  contains  $\ell \cap S'''$  for each line segment  $\ell \in \text{OPT}$  of level at least  $j - 3$  that intersects  $S'''$ .

We now apply the trivial operation until it is no longer applicable. We say that the above series of operations correspond to level  $j$ .

As an example, look at Fig. 3, with  $\varepsilon = 1/2$ . The horizontal and vertical solid segments in it are of level 0, the horizontal and vertical dashed-dotted segments are of level 1, the vertical dashed segments are of level 2, and the vertical dotted segments are of level 3. Also, the black segments are vertical grid lines, the blue segments are (well-aligned) segments in OPT and the red segments are lines along which horizontal line operations are applied. Here, say for level 0, we first apply line operations along the vertical grid lines of that level to get two subproblems. Then in the left subproblem, we start counting end points of segments in OPT, and at every eighth point, we do a horizontal line operation along its  $y$ -coordinate. Then we continue doing operations for the further levels. It can also be seen here that a well-aligned segment of level 0, always

starts and ends at grid lines of level 3. So when we do operations corresponding to level 3, it will get removed by a trivial operation (we prove this formally in Lemma 4).

## 2.4 Analysis of the 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  $(1 + \varepsilon)\text{OPT}$ . To this end, first we need to show that whenever we apply the add operation to a subproblem  $\text{DP}(S, \mathcal{L})$  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  $\ell \in \text{OPT}$  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  $\ell$  is well-aligned. More precisely, if  $\ell$  is horizontal then its  $x$ -coordinates are aligned with the grid lines of level  $j + 3$ . Hence, if  $\ell$  or a part of  $\ell$  is contained in a set  $\mathcal{L}$  of some subproblem  $\text{DP}(S, \mathcal{L})$  of level  $j + 3$ , then we applied the trivial operation to  $\ell$  (since it is well-aligned) and thus  $\ell$  “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  $\ell$  is vertical and it appears in a  $\text{DP}(S, \mathcal{L})$  of level  $j + 3$  then we applied the line operation to the horizontal lines that contain the two endpoints of  $\ell$ . Afterward, we applied the trivial operation to  $\ell$  until  $\ell$  “disappeared” from  $\mathcal{L}$ .

In particular, for each subproblem  $\text{DP}(S, \mathcal{L})$  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  $j' < j - 2$ . We make this and some other technicalities formal in the proof of the next lemma.

**Lemma 4** *The constructed tree  $T$  is a DP-decision tree.*

**Proof** We first notice that any reduction in the tree corresponding to a trivial or line operation on  $\text{DP}(S, \mathcal{L})$  can only lead to subproblems of the form  $\text{DP}(S', \mathcal{L}')$  with  $\mathcal{L}' \subseteq \mathcal{L}$ , and not lead to any violations of the DP-decision tree properties.

We now consider the add operations to ensure that any newly created node, by applying the add operation on the set  $\mathcal{L}' := \{\ell \cap S' \mid (\ell \in \text{OPT}) \wedge (\ell \cap S' \neq \emptyset) \wedge (\ell \text{ is of level } j)\}$  to the subproblem  $\text{DP}(S, \mathcal{L})$ , maintains the property that  $|\mathcal{L}| \cup |\mathcal{L}'| \leq 3\varepsilon^{-3}$ . We do this by induction,

**Hypothesis:** We assume that all add operations in the sequence of operations corresponding to level  $j - 1$ , satisfied the property that  $|\mathcal{L}| + |\mathcal{L}'| \leq 3\varepsilon^{-3}$ .

**Base case:** For the sequence of operations of level  $j < 3$ , we know by Propositions 1 and 2 that the ‘added’ set  $\mathcal{L}'$  always satisfies  $|\mathcal{L}'| \leq \varepsilon^{-3}$ . Since these are the only added sets in the first 3 levels, we can be sure that  $|\mathcal{L}| + |\mathcal{L}'| \leq 3\varepsilon^{-3}$ .

**Induction:** Consider an add operation for segments in  $\text{OPT}$  of level  $j$ . Any such operation was preceded by a series of line operations along grid lines of level  $j$  and all viable trivial operations. The grid lines of level  $j$  are separated by  $\varepsilon^{j-2}$  units, and hence any horizontal segments from  $\text{OPT}$  in  $\mathcal{L}$  of level  $j' < j - 2$  (which have length  $\geq \varepsilon^{j-2}$  and being well aligned, completely cut across any cell they intersect) can have the trivial operation applied to them, and be removed from  $\mathcal{L}$ . Similarly, for any

vertical segment in OPT of level  $j' < j - 2$ , we explicitly apply the line operation to the horizontal lines that contain its endpoints, and hence again this also gets removed from  $\mathcal{L}$  when we apply all possible trivial operations. Thus, we are left only with segments added in operations of levels  $j - 1$  and  $j - 2$  in  $\mathcal{L}$ . But we know from Proposition 2 that at each level we add at most  $\varepsilon^{-3}$  segments. Thus, the application of an add operation at this stage ensures that  $|\mathcal{L}| + |\mathcal{L}'| \leq 3\varepsilon^{-3}$ .

□

We want to show that the cost of the solution corresponding to  $T$  is at most  $(1 + O(\varepsilon))\text{OPT}$ . 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  $(1 + O(\varepsilon))\text{OPT}$  in expectation). Intuitively, when we apply the line operation to a vertical grid line  $\ell$  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  $\ell$ . A line segment  $\ell' \in \text{OPT}$  of level  $j$  stabs such a rectangle only if  $\ell'$  is intersected by  $\ell$  (if  $\ell'$  is horizontal) or the  $x$ -coordinate of  $\ell'$  is close to  $\ell$  (if  $\ell'$  is vertical). Here we use that  $h_i \geq w_i$  for each rectangle  $R_i \in \mathcal{R}$ .

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 levels, but this need not be separately considered since all grid lines of smaller levels are also grid lines of level  $j$ ). We will show that if we choose  $a$  randomly then the total cost of such grid lines is at most  $\varepsilon \cdot \text{OPT}$  in expectation. Hence, by using a constant approximation algorithm [8], in expectation the total cost due to all line operations for vertical line segments is at most  $O(\varepsilon) \cdot \text{OPT}$ .

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  $\text{DP}(S, \mathcal{L})$ . We will charge this cost to the line segments of OPT inside  $S$  of the current level or higher levels (that is, levels of higher numerical value). 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  $O(\varepsilon) \cdot \text{OPT}$  in total due to all applications of line operations for horizontal line segments.

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

Formally, we prove this in the following lemma.

**Lemma 5** *There is a discrete value for the offset  $a$  such that  $a \in \{0, \varepsilon^d, 2\varepsilon^d, \dots, \varepsilon^{-2}\}$  and the solution  $\text{SOL}([0, n] \times [0, 4n^2], \emptyset)$  in  $T$  has a cost of at most  $(1 + O(\varepsilon))\text{OPT}$ .*

**Proof** In the tree as defined above, the add operations are only applied on segments from OPT (or their parts), and hence the cost across all such add operations is at most  $c(\text{OPT})$ . Similarly, the trivial operations are applied on segments that 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 discretized random offset



$a \in \{0, \varepsilon^d, 2\varepsilon^d, \dots, \varepsilon^{-2}\}$ , the expected cost is  $O(\varepsilon \cdot \text{OPT})$ , which would give us the required result.

Let us first consider any line operation of level  $j$  that is applied to a horizontal line  $\ell$ . This operation would create two cells of width at most  $\varepsilon^{j-2}$  (since vertical grid lines of level  $j$  are separated by  $\varepsilon^{j-2}$ ), one of which either contains at least  $\varepsilon^{-3}$  endpoints of segments (horizontal or vertical) from OPT of level  $j$ ; or contains at least one vertical segment from OPT of level  $j' < j - 2$ , that is, the cost of the segments from OPT with at least one endpoint in this cell is at least  $\varepsilon^{j-3}$  (in the former case, each of the segments is of length at least  $\varepsilon^j$ , and in the latter the single segment is of the required length). Since a segment of width  $\varepsilon^{j-2}$  (width of the cell) is sufficient to stab all rectangles stabbed by  $\ell$ , we see that this horizontal line takes only  $\varepsilon$  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  $2\varepsilon \cdot \text{OPT}$ .

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 as mentioned above can also be stated as bounding the cost, over all levels  $j$ , of line segments of level  $j$  in OPT (call this set  $\text{OPT}_j$ ) intersected or close to grid lines of level  $j$ . For a horizontal segment  $\ell \in \text{OPT}_j$ , let  $I_\ell$  be the indicator variable representing the event that a grid line of level  $j$  intersects  $\ell$  ( $I_\ell = 0$  for vertical segments). Now, since any two consecutive grid lines of level  $j$  are separated by  $\varepsilon^{j-2}$ , there are  $\varepsilon^{j-2}/\varepsilon^d$  possible shifts for these grid lines due to our offset, and each of these shifts has the same probability. Similarly, since  $|\ell| \leq \varepsilon^{j-1}$  (by virtue of being well-aligned), at most  $(\varepsilon^{j-1}/\varepsilon^d) + 1$  of the valid offsets will cause a grid line to intersect  $\ell$ . So, if we take a random discrete offset  $a \in \{0, \varepsilon^d, 2\varepsilon^d, \dots, \varepsilon^{-2}\}$ , we have that

$$\mathbb{E}[I_\ell] \leq \frac{(\varepsilon^{j-1}/\varepsilon^d) + 1}{\varepsilon^{j-2}/\varepsilon^d} = \varepsilon + \varepsilon^{2+(d-j)} \leq 2\varepsilon \quad (\text{since } j \in [0, d - 1].)$$

For a vertical segment  $\ell \in \text{OPT}_j$ , let  $J_\ell$  be the indicator variable representing the event that a grid line of level  $j$  intersects a rectangle stabbed by  $\ell$  ( $J_\ell = 0$  for horizontal segments). Since for  $j > 0$ ,  $|\ell| \leq \varepsilon^{j-1}$ , we know that for each rectangle  $R_i$  stabbed by  $\ell$ , the dimensions satisfy  $w_i \leq h_i \leq \varepsilon^{j-1}$ . This means that to stab such a rectangle,  $\ell$  has to lie *close to* (that is, within  $\pm\varepsilon^{j-1}$  of) the vertical grid line, further implying that at most  $(2\varepsilon^{j-1}/\varepsilon^d) + 1$  of the discrete offsets are valid for event  $J_\ell$ . So for a discrete random offset  $a \in \{0, \varepsilon^d, 2\varepsilon^d, \dots, \varepsilon^{-2}\}$  and level  $j > 0$  we have that:

$$\mathbb{E}[J_\ell] \leq \frac{(2\varepsilon^{j-1})/\varepsilon^d + 1}{\varepsilon^{j-2}/\varepsilon^d} = 2\varepsilon + \varepsilon^{2+(d-j)} \leq 3\varepsilon.$$

For level  $j = 0$ , we note that even though the vertical segments can be very long, the maximum width of a rectangle is at most 1. So  $\ell$  has to lie within  $\pm 1$  of the grid line.



So if we take a random offset  $a \in \{0, \varepsilon^d, 2\varepsilon^d, \dots, \varepsilon^{-2}\}$  we have that

$$\mathbb{E}[J_\ell] \leq \frac{2/\varepsilon^d + 1}{\varepsilon^{-2}/\varepsilon^d} = 2\varepsilon^2 + \varepsilon^{d+2} \leq 3\varepsilon.$$

With the expectations computed above, we can upper bound the expected cost of segments in OPT intersected by, or *close to* vertical grid lines as:

$$\begin{aligned} \mathbb{E} \left[ \sum_j \sum_{\ell \in \text{OPT}_j} (I_\ell + J_\ell) \cdot |\ell| \right] &= \sum_j \sum_{\ell \in \text{OPT}_j} \mathbb{E} [(I_\ell + J_\ell) \cdot |\ell|] \\ &= \sum_j \sum_{\ell \in \text{OPT}_j} |\ell| \cdot (\mathbb{E}[I_\ell] + \mathbb{E}[J_\ell]) \\ &\leq \sum_j \sum_{\ell \in \text{OPT}_j} |\ell| \cdot (2\varepsilon + 3\varepsilon) \\ &= 5\varepsilon \cdot \text{OPT}. \end{aligned}$$

Recall that the bound above gives us a bound on the total cost of vertical line operations.

Now, by using an  $\alpha$ -approximation algorithm for stabbing [8], where  $\alpha$  is a constant, the solution returned by our algorithm takes an additional cost of  $5\alpha \cdot \varepsilon \cdot \text{OPT}$ .  $\square$

For a fixed value of our random offset  $a$ , the running time of our algorithm is polynomial. There are only  $\varepsilon^{-2}/\varepsilon^d = \varepsilon^{-d-2} < n/\varepsilon^5$  possible choices for  $a$ . Therefore, we can derandomize our algorithm by simply trying all of these possible choices and returning the best solution obtained in this way.

Now we prove our main theorem.

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

**Proof** We gave a DP algorithm in Sect. 2.2 which was shown to have the required running time in Lemma 2. Further in Lemma 5 we showed the correctness and that the solution computed by the DP is actually a  $1 + O(\varepsilon)$  approximation of the solution.  $\square$

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

**Corollary 1** *There is a PTAS for the horizontal–vertical square stabbing problem.*

Also, it yields a  $(2 + \varepsilon)$ -approximation algorithm for the horizontal–vertical 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 horizontal–vertical 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 $\delta$ -large rectangles

We now consider the case of  $\delta$ -large rectangles for some given constant  $\delta$ , that is, 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  $h_i \geq \delta$ . In other words, none of the rectangles are small (less than  $\delta$ ) in both dimensions at the same time. 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  $\text{OPT}(C_i)$  the segments from  $\text{OPT}$  that are contained in  $C_i$  and our routine ensures that  $c(\text{OPT}(C_i)) \leq O(1/\varepsilon^3)$ . Then for each cell  $C_i$ , the number of segments in  $\text{OPT}(C_i)$  with a length longer than  $\delta$  is bounded by  $O(1/(\delta\varepsilon^3))$ . We guess them in polynomial time. Now, the remaining segments in  $\text{OPT}$  are all of length smaller than  $\delta$ , 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. In the following, we describe our algorithm in detail.

#### 3.1 Guessing long segments in the solution

In the following, we will call a coordinate *discrete* if it is an integral multiple of  $\varepsilon/n$ . We first start by discretizing the input in a similar way as in Lemma 1,

**Lemma 6** *By losing a factor  $1 + O(\varepsilon)$  in the approximation ratio, we can assume for each  $R_i \in \mathcal{R}$  that the following properties hold:*

- (i)  $x_1^{(i)}, x_2^{(i)}, y_1^{(i)}, y_2^{(i)}$  are discrete and within  $[0, n]$ ,
- (ii) each line segment in  $\text{OPT}$  has length of at most  $1/\varepsilon$ .

**Proof** We first prove Property (i). Since each rectangle  $R_i \in \mathcal{R}$  has a width  $w_i \leq 1$ , the instance either has some  $x$ -coordinate that is not covered by any rectangle—in which case we can split the instance into smaller subproblems around this  $x$ -coordinate—or the total width of the instance is at most  $n$ . Hence, all the  $x$ -coordinates of the rectangles can be assumed to be between 0 and  $n$ . Further, we extend the rectangles on both sides to make their  $x$ -coordinates align with the next nearest multiple of  $\varepsilon/n$  (discretization). Clearly this involves extension by at most  $2\varepsilon/n$  to the width of every rectangle, and at most a  $2\varepsilon$  addition to the cost of the solution. We can use the same argument to show that the heights are also discretized and between 0 and  $n$ , since the

heights of all rectangles are also less than 1. The height discretization similarly adds another  $2\varepsilon$  to the cost of the solution.

Now we look at Property (ii). Consider any horizontal (respectively vertical) segment  $\ell \in \text{OPT}$ , that is longer than  $1/\varepsilon$ . From the left (respectively bottom) endpoint, we divide the segment into consecutive smaller segments of length  $1/\varepsilon - 2$  each, with one potential last piece being smaller than  $1/\varepsilon - 2$ . Now, for each smaller segment, we extend it on both sides in such a way that it completely stabs the rectangles that it intersects. Since the maximum width (respectively height) of a rectangle is 1, we extend each such segment by at most 2 units.

With the same calculation as in the proof of Lemma 1, we can bound the increase of our cost by

$$\left( |\ell| + \left\lceil \frac{|\ell|}{1/\varepsilon - 2} \right\rceil \cdot 2 \right) \cdot \frac{1}{|\ell|} \leq 1 + 8\varepsilon.$$

□

Then we continue by constructing vertical grid lines at each  $x$ -coordinate  $a + k \cdot \varepsilon^{-2}$  with  $k \in \mathbb{N}$  and a random offset  $a \in \mathbb{N}_0$ . By removing the rectangles that intersect with these grid lines, we divide the input instance into strips of width  $1/\varepsilon^2$  (and height  $n$ ) each. Next, we show that the removed rectangles can be stabbed at very low total cost.

**Lemma 7** *Consider the input rectangles that are intersected by vertical grid lines. They can all be stabbed by a set of segments of total expected cost  $O(\varepsilon) \cdot \text{OPT}$ .*

**Proof** Similar to Lemma 5, we wish to bound the cost of stabbing all the rectangles intersected or *close to* grid lines, which can also be stated as bounding the cost, of line segments of OPT intersected or close to the grid lines. For a horizontal segment  $\ell \in \text{OPT}$ , let  $I_\ell$  be the indicator variable representing the event that a grid line intersects  $\ell$  ( $I_\ell = 0$  for vertical segments). Since  $|\ell| \leq 1/\varepsilon$  (by Lemma 6), if we take a random offset  $a$ , we have that,

$$\mathbb{E}[I_\ell] \leq \frac{\varepsilon^{-1}}{\varepsilon^{-2}} = \varepsilon.$$

For a vertical segment  $\ell \in \text{OPT}$ , let  $J_\ell$  be the indicator variable representing the event that a grid line intersects a rectangle stabbed by  $\ell$  ( $J_\ell = 0$  for horizontal segments). Since the width of all rectangles is less than 1, to stab a rectangle,  $\ell$  has to lie *close to* (that is, within  $\pm 1$  of) the vertical grid line. So for a random offset  $a$  we have that,

$$\mathbb{E}[J_\ell] \leq \frac{2}{\varepsilon^{-2}} = 2\varepsilon^2 \leq 2\varepsilon.$$

With the expectations computed above, we can upper bound the expected cost of segments in OPT intersected by vertical line operations as,

$$\mathbb{E} \left[ \sum_{\ell \in \text{OPT}} (I_\ell + J_\ell) \cdot |\ell| \right] = \sum_{\ell \in \text{OPT}} \mathbb{E} [(I_\ell + J_\ell) \cdot |\ell|]$$

$$\begin{aligned}
&= \sum_{\ell \in \text{OPT}} |\ell| \cdot (\mathbb{E}[I_\ell] + \mathbb{E}[J_\ell]) \\
&\leq \sum_{\ell \in \text{OPT}} |\ell| \cdot (\varepsilon + 2\varepsilon) \\
&= 3\varepsilon \cdot \text{OPT}
\end{aligned}$$

Now, by using the  $\alpha$ -approximate algorithm for HV-STABBING[8], where  $\alpha$  is a constant, we can find a set of segments that stabs the set of rectangles intersected by the vertical grid lines with an additional cost of  $3\alpha \cdot \varepsilon \cdot \text{OPT}$ .  $\square$

Now we divide these vertical strips into smaller rectangular cells such that the optimal solution for each cell is  $O(1/\varepsilon^3)$ .

**Lemma 8** *In time  $n^{O(1)}$  we can compute a set of horizontal line segments of total cost  $O(\varepsilon) \cdot \text{OPT}$  that, together with the vertical grid lines, divides the input instance into rectangular cells containing independent subproblems, each with an optimal solution of cost  $O(1/\varepsilon^3)$ .*

**Proof** We will consider each vertical strip of width at most  $1/\varepsilon^2$  separately as they constitute independent subproblems. Consider one such strip. Since all rectangles are contained in  $[0, n] \times [0, n]$ , we consider only the portion of the strip contained in  $[0, n] \times [0, n]$ . We sweep a horizontal line from bottom to top, and at each discrete  $y$ -coordinate compute a constant approximate solution (using a known  $O(1)$ -approximation algorithm [8] for HV-STABBING) for the subproblem of the rectangles that are completely contained in the area of the strip below the line down to the bottom of the strip, or the end of a previous cell, whichever is closer. (Note here that we are only computing these solutions, and not adding all of them to our final answer.) At the smallest  $y$ -coordinate  $y_0$ , at which the cost of the solution exceeds  $1/\varepsilon^3$ , we take the line  $[0, 1/\varepsilon^2] \times y_0$  as a part of our solution and discard the rectangles already stabbed by it. Note that if we had moved the line up by another  $\varepsilon/n$ , the cost of the solution of the considered area could have increased by at most  $1/\varepsilon^2$  (by also considering the segment  $[0, 1/\varepsilon^2] \times y_0$  as part of the solution of the considered area), therefore the cost of segments from OPT in the considered area is between  $1/(\alpha\varepsilon^3)$  and  $1/\varepsilon^3 + 1/\varepsilon^2$ , which is  $\Omega(1/\varepsilon^3)$ .

Thus, every time we perform a step as described above, we add a segment of length  $1/\varepsilon^2$  to the solution, and create a subproblem that has an optimal solution of cost  $\Omega(1/\varepsilon^3)$ . Hence the cost of the added segment is at most  $O(\varepsilon^{-2}/\varepsilon^{-3}) = O(\varepsilon)$  times the cost of the solution to the subproblem that it creates. Since an optimal solution to all such subproblems can be at most the cost of the original instance, the cost of the horizontal segments that we have added is at most  $O(\varepsilon) \cdot \text{OPT}$ .  $\square$

Now, in any optimal solution to a subproblem contained in a cell as described above, there can be at most  $O(1/(\delta\varepsilon^3))$  long segments, that is, segments of length at least  $\delta$ . Since there are only a polynomial number of discrete candidate segments we can guess the set of long segments in the solution in polynomial time (the exact analysis is shown later).

### 3.2 Computing short segments in the solution

For each guess made for the set of long segments in the solution, we compute a  $(1 + \epsilon)$ -approximation for the *short segments*, that is, segments of length at most  $\delta$  in the solution using Theorem 1.

Note that any rectangle with both height and width at least  $\delta$  will have to be stabbed by a segment of length at least  $\delta$ . However, since all such long segments in the solution have already been guessed, each remaining segment can stab a  $\delta$ -large rectangle only along its shorter dimension. So the rectangles in the instance that have not yet been stabbed can be partitioned into *two* independent instances, one which has rectangles with width less than  $\delta$  (will be stabbed horizontally) and one which has rectangles with height less than  $\delta$  (will be stabbed vertically). We apply Theorem 1 on each of these independent instances and combine the solutions to get a  $(1 + \epsilon)$ -approximate solution for the remaining problem.

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

**Proof** First, we analyze the running time. In the first phase of the algorithm, we divide the given instance into  $n/(1/\epsilon^2) = n\epsilon^2$  strips, and run the constant factor  $n^{O(1)}$  approximation algorithm [8] for each discrete  $y$ -coordinate. This requires a running time of  $n\epsilon^2 \times n/(\epsilon/n) \times n^{O(1)} = \epsilon \cdot n^{O(1)}$ . Now in a strip there are at most  $(n/(\epsilon/n)) \times ((1/\epsilon^2)/(\epsilon/n)) = n^3/\epsilon^4$  possible discrete endpoints for segments, which means there are less than  $\binom{n^3/\epsilon^4}{2}$ , that is,  $O(n^6/\epsilon^8)$  possible candidate segments in a strip. Hence, we can enumerate over all of their subsets of size  $O(1/(\delta\epsilon^3))$  in  $(n/\epsilon)^{O(1/(\delta\epsilon^3))}$  time. Following this we have two invocations of Theorem 1 to solve the independent subproblems which would together take  $(n/\epsilon)^{O(1/\epsilon^3)}$ . This yields an overall running time of  $(n/\epsilon)^{O(1/(\delta\epsilon^3))}$ .

Now to show correctness, let us partition the OPT set into three sets,  $OPT_\delta$  consisting of segments of length at least  $\delta$ ,  $OPT_x$  consisting of horizontal segments shorter than  $\delta$ , and  $OPT_y$  consisting of vertical segments shorter than  $\delta$ . We can correctly guess  $OPT_\delta$  (up to a factor of  $1 + O(\epsilon)$  by Lemmas 6 and 8), and hence identify the rectangles that are stabbed by  $OPT_x$  but not by  $OPT_\delta$  (since these are the remaining rectangles which have width at most  $\delta$ ) and by  $OPT_y$  but not by  $OPT_\delta$ . Now by Theorem 1 we can compute  $(1 + \epsilon)$ -approximations for these two sets, and hence get a solution of cost

$$SOL \leq (1 + O(\epsilon))OPT_\delta + (1 + \epsilon)OPT_x + (1 + O(\epsilon))OPT_y \leq (1 + O(\epsilon))OPT.$$

Now by appropriately adjusting the value of  $\epsilon$ , we can obtain the claimed result.  $\square$

### 4 Generalized minimum Manhattan network

In the generalized minimum Manhattan network problem (GMMN), we are given a set  $R$  of  $n$  unordered terminal pairs, and the goal is to find a minimum-length rectilinear

network such that every pair in  $R$  is connected by a Manhattan-path, which is a path consisting of axis-parallel segments whose total length is the Manhattan distance of the pair under consideration.

For the special case of 2D-GMNN, where all terminals lie on a 2-dimensional plane, Das et al. [14] gave an  $O(\log n)$ -approximate solution using a  $(6 + \varepsilon)$ -approximate solution for  $x$ -separated 2D-GMMN instances. An instance of GMMN is called  $x$ -separated if there is a vertical line that intersects every rectangle formed by a pair of terminals as the diagonally opposite corners of the rectangle. Let  $\text{OPT}_{\text{ver}}$  and  $\text{OPT}_{\text{hor}}$  refer to the cost of the vertical and horizontal segments respectively in an optimal solution. The algorithm solves an  $x$ -separated instance by giving a solution of the form  $N = A_{\text{up}} \cup A_{\text{down}} \cup S$ , where  $S$  is a set of horizontal segments stabbing all the rectangles in the instance and  $A_{\text{up}} \cup A_{\text{down}}$  is a set of segments of cost bounded by  $(2 + 2\varepsilon) \cdot (\text{OPT} + \text{OPT}_{\text{ver}})$ . Using a 4-approximation algorithm for finding the set of segments  $S$ , horizontally stabbing the rectangles, a bound of  $(6 + \varepsilon) \cdot \text{OPT}$  is obtained for the cost of segments in  $N$ . But instead of the 4-approximate algorithm we can instead use the PTAS from Corollary 3 to obtain a better bound of  $(4 + \varepsilon) \cdot \text{OPT}$  on the cost as follows,

$$\begin{aligned} c(N) &= c(A_{\text{up}}) + c(A_{\text{down}}) + c(S) \\ &\leq (2 + 2\varepsilon) \cdot (\text{OPT} + \text{OPT}_{\text{ver}}) + (1 + \varepsilon) \cdot \text{OPT}_{\text{hor}} \\ &\leq (2 + 2\varepsilon) \cdot \text{OPT} + (2 + 2\varepsilon) \cdot (\text{OPT}_{\text{ver}} + \text{OPT}_{\text{hor}}) \\ &= (4 + \varepsilon') \cdot \text{OPT}. \end{aligned}$$

## 5 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 PTASes 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. Further, since the question of the APX-hardness of HV- STABBING is still open, it is not even clear whether such a PTAS exists.

## References

1. Adamaszek, A., Wiese, A.: Approximation schemes for maximum weight independent set of rectangles. In: FOCS, pp. 400–409 (2013)
2. Agarwal, P.K., Chang, H., Suri, S., Xiao, A., Xue, J.: Dynamic geometric set cover and hitting set. *ACM Trans. Algorithms* **18**(4), 40:1–40:37 (2022)
3. Aronov, B., Ezra, E., Sharir, M.: Small-size  $\varepsilon$ -nets for axis-parallel rectangles and boxes. *SIAM J. Comput.* **39**(7), 3248–3282 (2010)
4. Bansal, N., Khan, A.: Improved approximation algorithm for two-dimensional bin packing. In: SODA, pp. 13–25 (2014)
5. Bansal, N., Pruhs, K.: The geometry of scheduling. *SIAM J. Comput.* **43**(5), 1684–1698 (2014)
6. Becchetti, L., Marchetti-Spaccamela, A., Vaitelli, A., Korteweg, P., Skutella, M., Stougie, L.: Latency-constrained aggregation in sensor networks. *ACM Trans. Algorithms* **6**(1), 13:1–13:20 (2009)

7. Brönnimann, H., Goodrich, M.T.: Almost optimal set covers in finite VC-dimension. *Discrete Comput. Geom.* **14**(4), 463–479 (1995)
8. 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: ISAAC, pp. 61:1–61:13 (2018)
9. Chan, T.M., Grant, E.: Exact algorithms and apx-hardness results for geometric packing and covering problems. *Comput. Geom.* **47**(2), 112–124 (2014)
10. Chan, T.M., Grant, E., Könemann, J., Sharpe, M.: Weighted capacitated, priority, and geometric set cover via improved quasi-uniform sampling. In: SODA, pp. 1576–1585. SIAM (2012)
11. Chan, T.M., He, Q., Suri, S., Xue, J.: Dynamic geometric set cover, revisited. In: SODA, pp. 3496–3528. SIAM (2022)
12. Christensen, H.I., Khan, A., Pokutta, S., Tetali, P.: Approximation and online algorithms for multidimensional bin packing: a survey. *Comput. Sci. Rev.* **24**, 63–79 (2017)
13. Chvátal, V.: A greedy heuristic for the set-covering problem. *Math. Oper. Res.* **4**(3), 233–235 (1979)
14. Das, A., Fleszar, K., Kobourov, S.G., Spoerhase, J., Veeramoni, S., Wolff, A.: Approximating the generalized minimum manhattan network problem. *Algorithmica* **80**(4), 1170–1190 (2018)
15. Deppert, M.A., Jansen, K., Khan, A., Rau, M., Tutas, M.: Peak demand minimization via sliced strip packing. In: APPROX/RANDOM, pp. 21:1–21:24 (2021)
16. Eisenbrand, F., Gallato, M., Svensson, O., Venzin, M.: A QPTAS for stabbing rectangles. *CoRR arxiv:2107.06571* (2021)
17. Finke, G., Jost, V., Queyranne, M., Sebö, A.: Batch processing with interval graph compatibilities between tasks. *Discrete Appl. Math.* **156**(5), 556–568 (2008)
18. Gálvez, W., Grandoni, F., Ameli, A.J., Jansen, K., Khan, A., Rau, M.: A tight  $(3/2+\epsilon)$  approximation for skewed strip packing. In: APPROX/RANDOM, pp. 44:1–44:18 (2020)
19. Gálvez, W., Grandoni, F., Heydrich, S., Ingala, S., Khan, A., Wiese, A.: Approximating geometric knapsack via l-packings. In: FOCS, pp. 260–271 (2017)
20. Gálvez, W., Grandoni, F., Khan, A., Ramírez-Romero, D., Wiese, A.: Improved approximation algorithms for 2-dimensional knapsack: Packing into multiple l-shapes, spirals, and more. In: SoCG, pp. 39:1–39:17 (2021)
21. Gálvez, W., Khan, A., Mari, M., Mömke, T., Pittu, M.R., Wiese, A.: A  $(2+\epsilon)$ -approximation algorithm for maximum independent set of rectangles. *CoRR arxiv:2106.00623* (2021)
22. 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)
23. Harren, R., Jansen, K., Prädél, L., van Stee, R.: A  $(5/3 + \epsilon)$ -approximation for strip packing. *Comput. Geom.* **47**(2), 248–267 (2014)
24. Hochbaum, D.S., Maass, W.: Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM* **32**(1), 130–136 (1985)
25. Jansen, K., Khan, A., Lira, M., Sreenivas, K.V.N.: A PTAS for packing hypercubes into a knapsack. In: ICALP, pp. 78:1–78:20 (2022)
26. Kar, D., Khan, A., Wiese, A.: Approximation algorithms for round-ufp and round-sap. In: Chechik, S., Navarro, G., Rotenberg, E., Herman, G. (eds.) ESA, pp. 71:1–71:19 (2022)
27. Khan, A., Lonkar, A., Maiti, A., Sharma, A., Wiese, A.: Tight approximation algorithms for two-dimensional guillotine strip packing. In: ICALP, pp. 80:1–80:20 (2022)
28. Khan, A., Lonkar, A., Rahul, S., Subramanian, A., Wiese, A.: Online and dynamic algorithms for geometric set cover and hitting set. In: SoCG, pp. 46:1–46:17 (2023)
29. Khan, A., Maiti, A., Sharma, A., Wiese, A.: On guillotine separable packings for the two-dimensional geometric knapsack problem. In: SoCG, pp. 48:1–48:17 (2021)
30. Khan, A., Pittu, M.R.: On guillotine separability of squares and rectangles. In: APPROX/RANDOM, pp. 47:1–47:22 (2020)
31. Khan, A., Sharma, E.: Tight approximation algorithms for geometric bin packing with skewed items. In: APPROX/RANDOM, pp. 22:1–22:23 (2021)
32. Khan, A., Sharma, E., Sreenivas, K.V.N.: Geometry meets vectors: approximation algorithms for multidimensional packing. In: FSTTCS, pp. 23:1–23:22 (2022)
33. Khan, A., Subramanian, A., Wiese, A.: A PTAS for the horizontal rectangle stabbing problem. In: IPCO, pp. 361–374 (2022)
34. Kovaleva, S., Spieksma, F.C.R.: Approximation algorithms for rectangle stabbing and interval stabbing problems. *SIAM J. Discrete Math.* **20**(3), 748–768 (2006)

35. Mitchell, J.S.B.: Approximating maximum independent set for rectangles in the plane. In: FOCS, pp. 339–350 (2021)
36. Mömke, T., Wiese, A.: A  $(2 + \epsilon)$ -approximation algorithm for the storage allocation problem. In: ICALP, pp. 973–984 (2015)
37. Mömke, T., Wiese, A.: Breaking the barrier of 2 for the storage allocation problem. In: ICALP, pp. 86:1–86:19 (2020)
38. 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)
39. Varadarajan, K.R.: Weighted geometric set cover via quasi-uniform sampling. In: STOC, pp. 641–648 (2010)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.