# On Approximation Schemes for Stabbing Rectilinear Polygons

**Arindam Khan**
Indian Institute of Science, Bengaluru, India

**Aditya Subramanian**
Indian Institute of Science, Bengaluru, India

**Tobias Widmann**
Technical University of Munich, Germany

**Andreas Wiese**
Technical University of Munich, Germany

## Abstract

We study the problem of stabbing rectilinear polygons, where we are given $n$ rectilinear polygons in the plane that we want to stab, i.e., we want to select horizontal line segments such that for each given rectilinear polygon there is a line segment that intersects two opposite (parallel) edges of it. Our goal is to find a set of line segments of minimum total length such that all polygons are stabbed. For the special case of rectangles, there is an $O(1)$-approximation algorithm and the problem is NP-hard [Chan, van Dijk, Fleszar, Spoerhase, and Wolff, 2018]. Also, the problem admits a QPTAS [Eisenbrand, Gallato, Svensson, and Venzin, 2021] and even a PTAS [Khan, Subramanian, and Wiese, 2022]. However, the approximability for the setting of more general polygons, e.g., L-shapes or T-shapes, is completely open.

In this paper, we give conditions under which the problem admits a $(1 + \varepsilon)$-approximation algorithm. We assume that each input polygon is composed of rectangles that are placed on top of each other. We show that if all input polygons satisfy the *hourglass condition*, then the problem admits a quasi-polynomial time approximation scheme. In particular, it is thus unlikely that this case is APX-hard. Furthermore, we show that there exists a PTAS if each input polygon is composed out of rectangles with a bounded range of widths. On the other hand, we prove that the general case of the problem (in which the input polygons may not satisfy these conditions) is APX-hard, already if all input polygons have only eight edges. We remark that all polygons with fewer edges automatically satisfy the hourglass condition. For arbitrary rectilinear polygons we even show a lower bound of $\Omega(\log n)$ for the possible approximation ratio, which implies that the best possible ratio is in $\Theta(\log n)$ since the problem is a special case of Set Cover.

**2012 ACM Subject Classification** Theory of computation → Computational geometry

**Keywords and phrases** Approximation Algorithms, Stabbing, Rectangles, Rectilinear Polygons, QPTAS, APX-hardness

**Digital Object Identifier** 10.4230/LIPIcs.FSTTCS.2024.27

**Related Version** *Full Version*: https://arxiv.org/abs/2402.02412 [34]

**Funding** *Arindam Khan*: Research 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*: Research was supported by the Walmart Center for Tech Excellence.


44th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2024).
Editors: Siddharth Barman and Sławomir Lasota; Article No. 27; pp. 27:1–27:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 1 Introduction

The Stabbing problem is a geometric case of the well-studied Set Cover problem. We are given a set of geometric objects in the plane. The goal is to compute a set of horizontal line segments of minimum total length such that each given object $R$ is *stabbed*, i.e., there is a line segment $\ell$ for which $R \setminus \ell$ consists of two connected components. The problem was introduced by Chan, van Dijk, Fleszar, Spoerhase, and Wolff [9] for the case where each given object is an axis-parallel rectangle. In particular, they argued that this case models a resource allocation problem for frequencies. In this application, the $x$-axis models time and the $y$-axis represents a frequency spectrum. Each given rectangle represents a request for a time window $[t_1, t_2]$ and a frequency band $[f_1, f_2]$ that needs to be fulfilled. Each selected segment $[t'_1, t'_2] \times \{f'\}$ corresponds to opening a communication channel $f'$ during a time interval $[t'_1, t'_2]$ which then serves each request whose time window is contained in $[t'_1, t'_2]$ and for which $f$ is a frequency in its corresponding band $[f_1, f_2]$. Also, Das, Fleszar, Kobourov, Spoerhase, Veeramoni, and Wolff [13] showed a connection to the Generalized Minimum Manhattan Network problem.

The first result for the case of rectangles was a polynomial time $O(1)$-approximation due to Chan et al. [9]. Subsequently, Eisenbrand, Gallato, Svensson, and Venzin improved the approximation ratio to 8 and provided a QPTAS, i.e., a $(1+\varepsilon)$-approximation algorithm that runs in quasi-polynomial time [16]. In particular, this implies that the problem is unlikely to be APX-hard. After that, Khan, Subramanian, and Wiese presented a polynomial time approximation scheme (PTAS) for rectangles [35].

A natural question is the Stabbing problem for geometric shapes that are more general than rectangles. We restrict ourselves to rectilinear polygons. Rectilinear polygons can model more general types of requests in the resource allocation problem. Depending on the resource quality, the requested time period and preprocessing times for jobs may be different. This can be modeled as an instance of our problem, where each job is represented by multiple rectangular regions (each of them corresponds to a particular bandwidth interval and a time period during which the job may be processed), and the aim is to select bandwidths and corresponding time intervals such that each job is served. This corresponds to stabbing one of the rectangular regions corresponding to each job. If the rectangular regions are contiguous (which is quite common due to the locality of bandwidth requirements) they correspond to *$k$-shapes* which motivates studying these objects.

Also, from a theoretical point of view, it is natural to ask which approximation ratios are possible for more general geometric objects. As mentioned above, Stabbing admits a $(1+\varepsilon)$-approximation algorithm when all given objects are rectangles [35]. However, is this also true for slightly more general polygons, e.g., that have the shape of an L or a T, polyominoes, or even for arbitrary rectilinear polygons? If not, under which conditions on the input objects is a $(1+\varepsilon)$-approximation still possible? Also, given that Stabbing is a special case of Set Cover, another natural question is whether it is strictly easier than this problem.

In this paper, we investigate the questions above. We focus on a type of rectilinear polygons that we call *$k$-shapes*. Intuitively, a $k$-shape is formed by $k$ rectangles that are stacked on top of each other such that for any two consecutive rectangles, the top edge of the bottom rectangle is contained in the bottom edge of the top rectangle, or vice versa (see Figures 1 and 2). We denote by $k$-Stabbing the setting of the Stabbing problem in which the input consists of $k$-shapes.

## 1.1 Our contribution

In this paper, we give conditions on $k$-shapes in the input, under which, the $k$-STABBING problem admits a $(1+\varepsilon)$-approximation algorithm in (quasi-)polynomial time, which makes it unlikely that it is APX-hard in these cases. We provide two separate conditions for this. Also, we prove that if the input objects may (slightly) violate these conditions, then the problem becomes APX-hard. For arbitrary $k$-shapes, we prove even that the problem is as difficult as general SET COVER, which yields a lower bound of $\Omega(\log n)$ for the possible approximation ratio.

Our first condition on the input $k$-shapes is the *hourglass condition*. It requires intuitively that the rectangles of each $k$-shape in the input are stacked like an hourglass (see Figure 1 and Definitions 1 and 3). Formally, it states that if we consider the rectangle of each $k$-shape of the smallest width, then the rectangles on top of it are ordered non-decreasingly by width, and an analogous mirrored ordering holds for the rectangles below it. For example, all L-shapes and triominoes fulfill this condition. We prove that this setting admits a $(1+\varepsilon)$-approximation algorithm for any $\varepsilon > 0$ in quasi-polynomial running time, i.e., in time $n^{(\log n/\varepsilon)^{O(1)}}$. In particular, this makes it unlikely that this case is APX-hard. Our algorithm generalizes the known QPTAS for the case of rectangles [16]. However, it is arguably simpler. For example, it does not need an $O(1)$-approximation algorithm for the problem as a subroutine. Instead, we show that the calls to this subroutine can be replaced by suitable guessing steps and by an $O(\log n)$-approximation algorithm for general SET COVER.

Note that when we say *guess*, we mean that there are only a polynomial number of possible options to choose from. So one can iterate over all possible options and find one of the correct options in polynomial time.

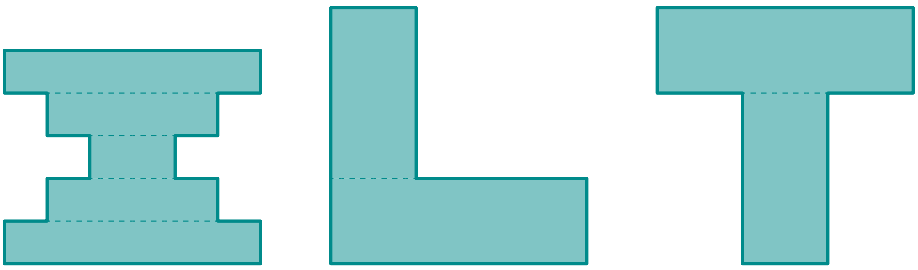

**Figure 1** Examples of $k$-shapes satisfying the hourglass condition.

**Figure 2** A 3-shape not satisfying the hourglass condition (left), and a stack of rectangles that does not form a $k$-shape (right).

Our algorithm is based on a hierarchical decomposition of the plane into smaller and smaller rectangular regions. Intuitively, given such a region $R$, we guess all line segments that are relatively long compared to the width of $R$. Then, we partition $R$ into smaller rectangular regions inside which we will select only shorter line segments. It can happen that a $k$-shape $K$ contained in $R$ is composed of at least one wide rectangle (of similar width as the guessed long line segments) and of at least one narrow rectangle (see Figure 3). If the guessed long line segments do not stab $K$, then it is clear that $K$ needs to be stabbed by a short line segment (that we select in one of the subproblems that we recurse into). Such line segments can stab only the narrow rectangles of $K$. Therefore, in this case we remove the wide rectangle from $K$ and hence make $K$ smaller. The hourglass condition ensures that after this removal, the remainder of $K$ still consists of only one connected component. We crucially need this property in order to ensure that the subproblems of $R$ we recurse into form independent subproblems. This would not be the case if the remainder of $K$ consisted of two connected components such that each of them lies in a different subproblem.

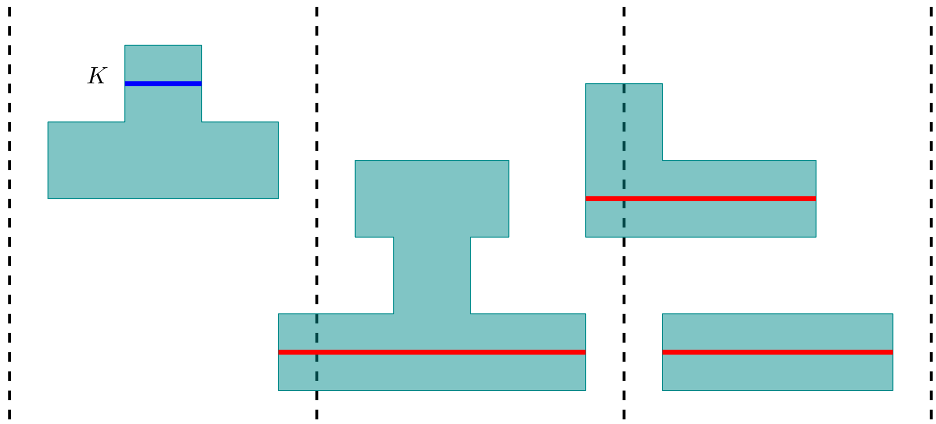


**Figure 3** The guessed (red) long segments do not stab $K$, so it has to be stabbed by a shorter (blue) segment in a future step.

While the hourglass condition is crucial for our algorithm above, it could be that it is not needed in an alternative algorithmic approach that computes a $(1+\varepsilon)$-approximation for, e.g., general $k$-shapes. However, we prove that this is not the case. We show that our problem is APX-hard, already if the input consists only of 3-shapes that do not satisfy the hourglass condition. On the other hand, note that each 2-shape automatically satisfies the hourglass condition by definition.

In our proof of this APX-hardness result, we construct 3-shapes that are composed out of three rectangles whose widths differ a lot. We prove that the latter is necessary in order to prove that our problem is APX-hard. To this end, we show that it admits a polynomial time $(1+\varepsilon)$-approximation algorithm for any constant $k \in \mathbb{N}$ and $\varepsilon > 0$ if each $k$-shape is composed out of rectangles whose widths are in a *constant* range. This yields our second condition under which our problem admits a $(1+\varepsilon)$-approximation. In fact, our result can handle some other classes of polygons which may not even be $k$-shapes, including polyominoes with $O(1)$ number of cells such as trominoes, tetrominoes (shapes that appear in the game *Tetris*), pentominoes, etc. Our algorithm is a generalization of the PTAS for rectangles [35]. One crucial insight is that if the widths of the rectangles of each input $k$-shape differ by at most a constant factor of $1/\delta$, then we can reduce our problem to the setting of rectangles by losing only a factor of $O(k/\delta)$. To do this, we simply replace each $k$-shape $K$ by the smallest rectangle that contains $K$. We use this insight in one step of our algorithm where we need an $O(1)$-approximation algorithm as a black box. More precisely, we again partition the input plane hierarchically into smaller and smaller rectangular regions. In the process, we repeatedly need to compute constant factor approximations for certain sets of $k$-shapes that intuitively admit a solution whose cost is at most $O(k\delta\varepsilon\mathrm{OPT})$; for those, we use the mentioned algorithm. We stab all other $k$-shapes with segments whose total cost is at most $(1+\varepsilon)\mathrm{OPT}$, which yields a PTAS.

We round up our results by showing that for general $k$-shapes and, more generally, even arbitrary rectilinear polygons that are composed of $k$ rectangles each, Stabbing admits a polynomial time $O(k)$-approximation algorithm. A natural question is whether the dependence on $k$ (and the input size) in the approximation ratio can be avoided and there is, e.g., also an $O(1)$-approximation. We show that this is not the case: for arbitrary $k$, we prove that $k$-Stabbing is as difficult as arbitrary instances of Set Cover, which yields a lower bound of $\Omega(\log n)$ for our approximation ratio.

## 1.2 Other related work

As mentioned above, the Stabbing problem is a special case of Set Cover which is NP-hard [23] and which does not admit a $(c \cdot \ln n)$-approximation algorithm for Set Cover for any $c < 1$, assuming that $\mathsf{P} \neq \mathsf{NP}$ [15] (see also [18]). On the other hand, a simple polynomial time greedy algorithm [12] achieves an approximation ratio of $O(\log n)$.

Das, Fleszar, Kobourov, Spoerhase, Veeramoni, and Wolff [13] studied approximation algorithms for the Generalized Minimum Manhattan Network (GMMN) problem, where given a set of $n$ pairs of terminal vertices, the goal is to find a minimum-length rectilinear network such that each pair is connected by a Manhattan path. The currently best known approximation ratio for this problem is $(4+\varepsilon)\log n$, due to Khan, Subramanian, and Wiese [35] by using their PTAS for Stabbing as a subroutine in a variant of the algorithm of Das et al. [13].

Gaur, Ibaraki, and Krishnamurti [24] studied the problem of stabbing rectangles by a minimum number of axis-aligned lines and obtained an LP-based 2-approximation algorithm. Kovaleva and Spieksma [37] studied a weighted generalization of this problem and gave an $O(1)$-approximation algorithm.

Geometric set cover is a related geometric special case of general Set Cover, where the given sets are geometric objects. Brönnimann and Goodrich [5] first gave an $O(d\log(d \cdot \mathrm{OPT}))$-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. Aronov, Ezra, and Sharir [3] utilized $\varepsilon$-nets to design an $O(\log\log \mathrm{OPT})$-approximation algorithm for the hitting set problem involving axis-parallel rectangles. Varadarajan [46] provided an improved approximation algorithm for weighted geometric set cover for fat triangles or disks, and his techniques were extended by Chan, Grant, Könemann, and Sharpe [7] to any set system with low shallow cell complexity. Subsequently, Chan and Grant [6], and Mustafa, Raman, and Ray [42] have settled the APX-hardness statuses of (almost) all natural variants for this problem. Recently, these problems are studied under online and dynamic setting as well [2, 8, 31].

Maximum Independent Set of Rectangles is another related problem. The problem admits a QPTAS [1], and recently a breakthrough $O(1)$-approximation algorithm was given by Mitchell [41]. Subsequently, a $(2+\varepsilon)$-approximation guarantee [21] was achieved.

Rectangle packing and covering problems such as two-dimensional knapsack [19, 28, 32], two-dimensional bin packing [4, 33], strip packing [27, 30] etc. are well-studied in computational geometry and approximation algorithms. We refer the readers to [11] for a survey on the approximation/online algorithms related to rectangles.

Rectilinear polygons appear naturally in the context of circuit design [38], architectural design [44], geometric information systems [10], computer graphics [45], etc. In computational geometry, often problems (for general polygons) are studied in the rectilinear setting, e.g., the art gallery problem [47], rectilinear convex hull [43], and rectilinear steiner tree [22]. Specially, $L$-shape polygons are encountered in many geometric problems as they are the simplest nonconvex rectilinear polygons. These $L$-shapes appear in geometric packing [20, 32], folding [14], VLSI layouts [39], lithography [48], etc. Polyominoes [26, 40] are special type of rectilinear polygons that are formed by joining one or more equal squares edge to edge. They are well-studied in the context of tiling [25], percolation theory and statistical physics [29], polymer chemistry [17], etc. They also appear in many puzzles and board games, including Tetris, Blokus, Rampart, Cathedral, etc.

### 1.3 Organization of this paper

First, in Section 2 we introduce some basic definitions and notation. Then, in Section 3 we present our QPTAS for $k$-shapes satisfying the hourglass condition, and in Section 4 we present our PTAS for $k$-shapes whose rectangles have a bounded ratio of widths. Finally, in Section 5 we present our hardness results.

## 2 Preliminaries

We start with some basic definitions and notations. We represent a given axis-aligned rectangle $R_i$ as the Cartesian product of two given closed and bounded intervals, i.e., $R_i = [x_i^\ell, x_i^r] \times [y_i^b, y_i^t]$ for given coordinates $x_i^\ell, x_i^r, y_i^b, y_i^t \in \mathbb{N}$, where $x_i^\ell \leq x_i^r$ and $y_i^b \leq y_i^t$. The following notation will be useful: we define

- $b(R_i) := [x_i^\ell, x_i^r] \times \{y_i^b\}$ as the *bottom edge* of $R_i$,
- $t(R_i) := [x_i^\ell, x_i^r] \times \{y_i^t\}$ as the *top edge* of $R_i$, and
- $w(R_i) := (x_i^r - x_i^\ell)$ as the *width* of $R_i$.

A *horizontal line segment* $s \subset \mathbb{R}^2$ is a Cartesian product $s = [x^\ell, x^r] \times \{y\}$ with coordinates $x^\ell, x^r, y \in \mathbb{N}$ and $x^\ell \leq x^r$. We say that $s$ *stabs* the rectangle $R_i$ if and only if $R_i \cap s = [x_i^\ell, x_i^r] \times \{y\}$. Also, we define $|s| := x^r - x^\ell$ to be the *length* or the *cost* of $s$. We will study the Stabbing problem in the setting where each given object is a $k$-shape.

▶ **Definition 1** ($k$-shape)**.** *Let $k \in \mathbb{N}$. A* $k$-shape *$K$ is the union of a sequence of at most $k$ axis-aligned rectangles $(R_1, R_2, \ldots, R_k)$ such that $t(R_i) \subseteq b(R_{i+1})$ or $t(R_i) \supseteq b(R_{i+1})$ for each $i \in \{1, \ldots, k-1\}$.*

We say that a $k$-shape $K = R_1 \cup \cdots \cup R_k$ is *stabbed* by a line segment $s$, if there exists an index $i \in \{1, \ldots, k\}$ such that the rectangle $R_i$ is stabbed by $s$. This leads to the following formal definition of the Stabbing problem for $k$-shapes.

▶ **Definition 2.** *Let $k \in \mathbb{N}$. An instance of the $k$-Stabbing problem is a finite set of $k$-shapes $\mathcal{K}$, where the objective is to find a set $\mathcal{S}$ of horizontal line segments of minimum total length, such that every $k$-shape in $\mathcal{K}$ is stabbed by a segment in $\mathcal{S}$.*

In the following section, we shall use the term OPT interchangeably to refer to the optimal solution to the problem, and also to represent its cost, i.e., the total length of segments in the set. Similarly, SOL will be used to represent a solution set and also its cost.

## 3 Quasi-polynomial-time approximation scheme

In this section, we present our QPTAS for $k$-Stabbing. The algorithm is an extension of the QPTAS for Stabbing [16] to the more general case of $k$-shapes; also, we simplify some of its steps.

Let $\varepsilon > 0$ and suppose we are given a set of $k$-shapes $\mathcal{K}$. In this section, we assume that each given $k$-shape $K \in \mathcal{K}$ satisfies the hourglass condition (see Figure 1).

▶ **Definition 3.** *A $k$-shape $K = (R_1, R_2, \ldots, R_k)$ satisfies the* hourglass condition *if there is no value $i \in \{2, \ldots, k-1\}$ such that both $w(R_{i-1}) < w(R_i)$ and $w(R_{i+1}) < w(R_i)$.*

For each given $k$-shape $K$, we define $w_{\max}(K) := \max_{i\in\{1,\ldots,k\}} w(R_i)$ and similarly $w_{\min}(K) := \min_{i\in\{1,\ldots,k\}} w(R_i)$ which are the widths of the widest and most narrow parts of $K$, respectively. For sets of $k$-shapes $\mathcal{K}' \subseteq \mathcal{K}$, we define accordingly $w_{\max}(\mathcal{K}') := \max_{K\in\mathcal{K}'} w_{\max}(K)$, $w_{\min}(\mathcal{K}') := \min_{K\in\mathcal{K}'} w_{\min}(K)$. Moreover, we define $w_{\text{range}}(\mathcal{K}') := \min\{w \mid \exists x \forall K \in \mathcal{K}' : K \subseteq [x, x+w] \times \mathbb{R}\}$ as the width of the most narrow strip that contains all $k$-shapes in $\mathcal{K}'$. Further, we note here that there are $n$ given $k$-shapes and each is described by at most $2k$ distinct points. Therefore, the solution to the instance has only $\binom{2kn}{2}$ combinatorially distinct candidate segments, which is a polynomial in $n$ (we shall use the notation that the number of candidate segments is $\text{poly}(n)$).

▶ **Lemma 4.** *By losing a factor of $1+\varepsilon$ in our approximation ratio, we assume that $\frac{\varepsilon}{n} < w_{\min}(\mathcal{K}) \le w_{\max}(\mathcal{K}) \le \log n$ and $w_{\text{range}}(\mathcal{K}) \le n \log n$.*

Let $\mu := \varepsilon / \log^2 n$. We partition the plane into relatively wide vertical strips of width $w_{\max}(\mathcal{K})/\mu$ each. We do this such that, intuitively, almost all input shapes are contained in one of our strips, and the remaining shapes, which are intersected by the vertical grid lines, can be stabbed very cheaply. To construct this partition, we define vertical grid lines with a spacing of $w_{\max}(\mathcal{K})/\mu$ and give them a random horizontal shift (see Figure 4). Then, each shape in $\mathcal{K}$ intersects one of these grid lines only with very small probability. Therefore, we can show that there exists a specific way to perform the shift of our grid lines such that all input shapes intersecting our grid lines can be stabbed with line segments whose cost is at most $\mu \cdot \text{OPT}$.

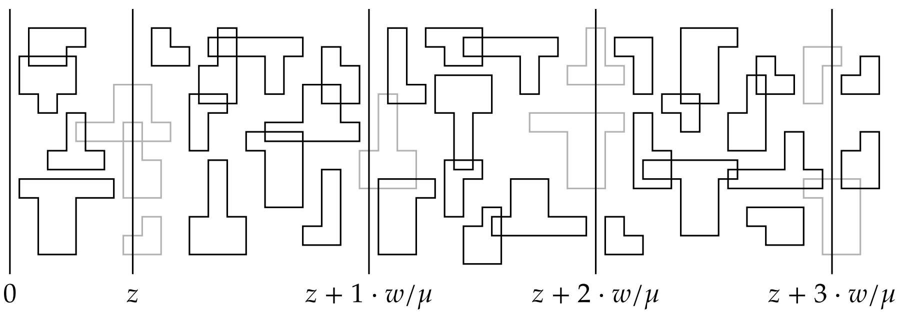


**Figure 4** Partitioning the instance into narrow strips.

Formally, we invoke the following lemma with our choice for $\mu$ defined above. It guesses a set of line segments that yield our desired partition into narrow strips, i.e., it produces a polynomial number of candidate sets such that one of them has the claimed property. Algorithmically, we recurse on each of these polynomially many options and at the end output the returned solution with the smallest total cost.

▶ **Lemma 5** (Partitioning into narrow strips). *Let $\mu > 0$ such that $\mu/n < w_{\min}(\mathcal{K})$. In polynomial time, we can guess a partition of $\mathcal{K}$ into sets $\mathcal{K}_0, \ldots, \mathcal{K}_t$ and one special set $\mathcal{K}_{\text{rest}}$ such that*

**(i)** $\text{OPT} \geq \sum_{\ell=1}^{t} \text{OPT}(\mathcal{K}_\ell)$,

**(ii)** $\text{OPT}(\mathcal{K}_{\text{rest}}) \leq 8\mu \cdot \text{OPT}$, *and*

**(iii)** $w_{\text{range}}(\mathcal{K}_i) \leq w_{\max}(\mathcal{K})/\mu$ *for each* $i \in \{1, \ldots, t\}$.

We compute an $O(\log n)$-approximate solution for stabbing $\mathcal{K}_{\text{rest}}$ by reducing our problem to an instance of Set Cover (see full version [34] for details). By our choice of $\mu$, the resulting cost is at most $O(\log n \cdot \mu \cdot \text{OPT}) = O(\text{OPT} \cdot \varepsilon/\log n)$. Hence, this step is simpler than the corresponding step in the previous QPTAS for Stabbing [16]. In that result, an $O(1)$-approximation algorithm for Stabbing was needed, while we can simply call an arbitrary standard $O(\log n)$-approximation algorithm for Set Cover, e.g., the straight-forward greedy algorithm.

Now let $\mathcal{K}_i$ be one of the sets of $k$-shapes due to Lemma 5. We define $S_i := [a, b] \times \mathbb{R}$ for some values $a, b \in \mathbb{R}$ with $b - a \leq w_{\max}(\mathcal{K})/\mu$ such that each $k$-shape in $\mathcal{K}_i$ is contained in $S_i$. We want to partition $S_i$ along horizontal lines into rectangular pieces such that each resulting piece contains line segments from $\text{OPT}(\mathcal{K}_i)$ of total cost at most $O(w_{\max}(\mathcal{K})/\mu^2)$. To this end, we guess whether the segments in $\text{OPT}(\mathcal{K}_i)$ have a total cost of at most $w_{\max}(\mathcal{K})/\mu^2$. If this is not the case, we guess a line segment $s = [a, b] \times \{h\}$ for some value $h \in \mathbb{N}$ according to the following lemma, which intuitively partitions $S_i$ in a balanced way according to the segments in $\text{OPT}(\mathcal{K}_i)$. We call such a segment $s$ a *balanced horizontal cut*. Also, here our algorithm is simpler than the earlier QPTAS for Stabbing [16]. In the latter algorithm, an $O(1)$-approximate algorithm for the problem was used to find the "correct" horizontal cuts algorithmically. Instead, we show that it is sufficient to simply guess them.

▶ **Lemma 6.** *If* $\text{OPT}(\mathcal{K}_i) > w_{\max}(\mathcal{K})/\mu^2$ *then in polynomial time we can guess a value* $h \in \mathbb{N}$ *and a corresponding line segment* $s = [a, b] \times \{h\}$ *such that each connected component* $C$ *of* $S_i \setminus s$ *contains segments from* $\text{OPT}(\mathcal{K}_i)$ *whose total cost is at least* $\text{OPT}(\mathcal{K}_i)/2 - w_{\max}(\mathcal{K})/\mu$.

**Proof.** Since we use only horizontal segments to stab $k$-shapes, w.l.o.g. (by stretching along the $y$ direction) we can assume that the at most $2kn$ points describing the instance occupy consecutive integral $y$-coordinates, starting at $y = 0$. Note that such a stretching step along the $y$ direction will not affect the length of the horizontal segments used to stab the $k$-shapes.

Consider the segments from $\text{OPT}(\mathcal{K}_i)$. Starting from $y = 0$ and going up, we can start counting the cumulative cost of segments in OPT. Let $h$ be the $y$-coordinate at which this cumulative cost crosses $\text{OPT}(\mathcal{K}_i)/2$, and $s = [a, b] \times \{h\}$ be the corresponding segment. Since the width of $S_i$ is at most $w_{\max}(\mathcal{K})/\mu$, no segment in $\text{OPT}(\mathcal{K}_i)$ is wider than $w_{\max}(\mathcal{K})/\mu$. From this we can infer that the cost of segments from $\text{OPT}(\mathcal{K}_i)$, below (and similarly, above) the segment $s$ should have been at least $\text{OPT}(\mathcal{K}_i)/2 - w_{\max}(\mathcal{K})/\mu$.

Since there are only a polynomial (i.e., $2kn$) number of possible $y$-coordinates, we can guess this value $h$ in polynomial time by enumeration. ◀

We add $s$ to our solution and recurse on each connected component $C$ of $S_i \setminus s$ separately. The resulting subproblem is to stab all input shapes that are contained in $C$. Observe that $s$ stabs all $k$-shapes contained in $S_i$ that intersect both connected components of $S_i \setminus s$. Given $C$, we guess again whether $\text{OPT}(C)$, i.e., the optimal solution for all $k$-shapes contained in $C$, has a total cost of at most $w_{\max}(\mathcal{K})/\mu^2$, and if not, we guess a corresponding horizontal line segment. Note that we stop after at most $O(\log n)$ recursion levels if all guesses are correct, since $\mathrm{OPT}(S_i) \leq \mathrm{OPT} \leq n \log n$ due to our preprocessing in Lemma 4. We enforce that in any case we stop after $O(\log n)$ recursion levels in order to guarantee a quasi-polynomial bound on the running time later.

▶ **Lemma 7.** *If all guesses for the balanced horizontal cuts are correct, then their total cost is bounded by* $3\mu \cdot \mathrm{OPT}(\mathcal{K}_i)$.

**Proof.** After our sequence of (correctly guessed) balanced horizontal cuts, let us assume that there are $t$ connected components, with cost at least $w_{\max}(\mathcal{K}_i)/2\mu^2 - w_{\max}(\mathcal{K})/\mu$. This can happen only if there were $t-1$ such cuts applied. If we charge the cost of every cut $s$ to the cost of segments of $\mathrm{OPT}(\mathcal{K}_i)$ within a cell $C$, we get

$$\frac{|s|}{\mathrm{OPT}(C)} = \frac{w_{\max}(\mathcal{K})/\mu}{w_{\max}(\mathcal{K})/2\mu^2 - w_{\max}(\mathcal{K})/\mu} = \frac{2\mu}{1-2\mu} \leq 3\mu.$$

Where the last inequality follows under the assumption of $\mu \leq \varepsilon < 1/3$. Summing over all such horizontal cuts, we get the total cost to be at most $3\mu \cdot w_{\max}(\mathcal{K}_i)$. ◀

At the end, each resulting subproblem is characterized by a rectangle $C$ of width at most $w_{\max}(\mathcal{K})/\mu$ and for which $\mathrm{OPT}(C) \leq w_{\max}(\mathcal{K})/\mu^2$. We guess all line segments in $\mathrm{OPT}(C)$ whose width is larger than $\varepsilon w_{\max}(\mathcal{K})$. Since $\mathrm{OPT}(C) \leq w_{\max}(\mathcal{K})/\mu^2$ there can be at most $1/\varepsilon\mu^2 = \varepsilon^{-3}\log^4 n$ of them, and for each of them there are only $\mathrm{poly}(n)$ options. Hence, we can guess them in time $n^{O(\varepsilon^{-3}\log^4 n)}$. Let $\mathcal{S}_C$ denote the guessed segments.

Our next step crucially differs from the known (Q)PTASs for stabbing rectangles [16, 35]. In particular, it is not necessary when all input objects are rectangles. Inside $C$, there might be a $k$-shape $K$ that is not stabbed by any segment in $\mathcal{S}_C$ but for which one of its rectangles $R_i$ satisfies that $w(R_i) > \varepsilon w_{\max}(\mathcal{K})$. Since we have guessed all segments in $C$ of width larger than $\varepsilon w_{\max}(\mathcal{K})$ and did not yet stab $K$, we know that the optimal solution does not stab $K$ by stabbing $R_i$ (but by stabbing another rectangle that $K$ is composed of). Therefore, we modify $K$ by removing $R_i$ from $K$. We do this for each rectangle $R_i$ with $w(R_i) > \varepsilon w_{\max}(\mathcal{K})$ that is part of a $k$-shape $K$ that is contained in $C$ but not yet stabbed. Denote by $\mathcal{K}'(C)$ the resulting set of $k$-shapes. Importantly, the hourglass property implies that still each $k$-shape has only one single connected component. This is the reason why we imposed this property.

Observe that for each $K \in \mathcal{K}'(C)$ we have that $w_{\max}(K) \leq \varepsilon \cdot w_{\max}(\mathcal{K})$. Thus, we made progress in the sense that the maximum width of any $k$-shape reduces by a factor of $\varepsilon$. Also, if all our guesses are correct, then our total cost is small, i.e., $O(\mu \cdot \mathrm{OPT})$. Also, the number of guesses is quasi-polynomially bounded since for each guess there are only $n^{O(\varepsilon^{-3}\log^4 n)}$ many options and our recursion depth is only $O(\log n)$.

▶ **Lemma 8.** *If all our guesses are correct, then the total cost for the selected line segments due to Lemma 5 and Lemma 7 is bounded by* $O(\mu \cdot \mathrm{OPT})$. *Also, the total number of (combinations of) guesses is bounded by* $n^{O(\varepsilon^{-3}\log^4 n)}$.

We continue recursively with each resulting subproblem. Since initially $\frac{\varepsilon}{n} < w_{\min}(\mathcal{K}) \leq w_{\max}(\mathcal{K}) \leq \log n$, we stop after applying the algorithm above for $O(\log(n/\varepsilon))$ levels. Each level incurs in total at most $n^{O(\varepsilon^{-3}\log^4 n)}$ guesses, which yields a total running time of $n^{O(\varepsilon^{-4}\log^5 n)}$. Also, our approximation ratio can easily be bounded by $(1+\mu)^{O(\log n)} = 1 + O(\varepsilon)$.

▶ **Theorem 9.** *There is a QPTAS for the k-*Stabbing *problem, if all input k-shapes satisfy the hourglass condition.*

## 4 PTAS if pieces have bounded ratio of widths

In this section, we improve our QPTAS from Section 3 to a PTAS in the special case when $k$ is a constant and when for each given $k$-shape, for any two of its rectangles $R_i, R_j$, it holds that $\delta w(R_j) \le w(R_i) \le w(R_j)/\delta$ for a given constant $\delta > 0$. Our algorithm generalizes the known PTAS for the case when all input objects are rectangles [35].

Let $\alpha$ be a constant for which the problem admits a polynomial time $\alpha$-approximation algorithm. We show in the full version [34] that such an algorithm exists. Without loss of generality, we assume that $\alpha, (1/\varepsilon) \in \mathbb{N}$, and we say that an $x$-coordinate $x \in \mathbb{R}$ is *discrete* if $x$ is an integral multiple of $\varepsilon^d$, where we define $d \in \mathbb{N}$ such that $\varepsilon^3/n < \varepsilon^d \le \varepsilon^2/n$; note that hence $d$ is unique. Similarly, a $y$-coordinate is called *discrete* if it is integral. 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.

▶ **Lemma 10.** *Let $\alpha$ be a constant for which $k$-Stabbing admits an $\alpha$-approximate algorithm and let $\varepsilon > 0$ with $\varepsilon < 1/3$. In polynomial time we can compute a new instance of $k$-Stabbing, in which each $K \in \mathcal{K}$ satisfies,*

**(i)** $\frac{\alpha\varepsilon}{n} < w_{\min}(K) \le w_{\max}(K) \le \alpha$,
**(ii)** *all points defining $K$ are discrete,*
**(iii)** *$K$ lies within a bounding box of $[0, \alpha n] \times [0, (k+1)n]$,*

*and this new instance admits a solution of cost at most $(1 + O(\varepsilon)) \cdot \mathrm{OPT}$ with each segment in the solution being discrete, and having length at most $\alpha/\varepsilon$.*

First, we apply Lemma 10 in order to preprocess our instance. In our algorithm, we intuitively embed the recursion of our QPTAS in Section 3 into a polynomial time dynamic program, such that we can afford to forget most of the balanced horizontal cuts from the higher levels, and only need to remember a constant number of the corresponding line segments. The idea is to construct a DP-table that contains one cell for each possible subproblem of a recursive call. Formally, we introduce one DP cell $\mathrm{DP}(R, \mathcal{S})$ for each combination of

- a closed rectangle $R \subseteq [0, \alpha n] \times [0, (k+1)n]$ with discrete coordinates,
- a set $\mathcal{S}$ of at most $\varepsilon^{-3}$ discrete horizontal line segments, that all intersect $R$.

This DP cell encodes the subproblem of stabbing all input $k$-shapes that are contained in $R$ and that are not already stabbed by the segments in $\mathcal{S}$. Clearly, the DP cell $\mathrm{DP}([0, \alpha n] \times [0, (k+1)n], \emptyset)$ corresponds to our given problem.

Given a DP cell $\mathrm{DP}(R, \mathcal{S})$, we compute its solution as follows. The base case occurs when the line segments in $\mathcal{S}$ already stab all $k$-shapes that are contained in $R$. Then we define $\mathrm{DP}(R, \mathcal{S}) := \emptyset$. Another easy case occurs when there is a line segment $\ell \in \mathcal{S}$ that stabs the interior of $R$, i.e., $R \setminus \ell$ has two connected components $R_1$ and $R_2$. Assume that $\mathcal{S}_1$ and $\mathcal{S}_2$ are parts of the line segments from $\mathcal{S}$ that intersect $R_1$ and $R_2$, respectively. Then we define $\mathrm{DP}(R, \mathcal{S}) := \mathrm{DP}(R_1, \mathcal{S}_1) \cup \mathrm{DP}(R_2, \mathcal{S}_2) \cup \{\ell\}$. We will refer to this later as the *trivial operation.*

Otherwise, we compute a polynomial number of candidate solutions as follows,

1. *Add operation.* For each set $\mathcal{S}'$ of discrete segments contained in $R$ for which $|\mathcal{S}| + |\mathcal{S}'| \le 3\varepsilon^{-3}$ holds, we generate the candidate solution $\mathcal{S}' \cup \mathrm{DP}(R, \mathcal{S} \cup \mathcal{S}')$.
2. *Line operation.* Consider each vertical line $\ell$ that intersects the interior of $R$. Let $\mathcal{K}_\ell$ denote the set of $k$-shapes contained in $R$ that intersect with $\ell$. For each $K \in \mathcal{K}_\ell$ we construct the smallest axis-parallel rectangle that contains $K$, let $\mathcal{R}_\ell$ denote the resulting set of rectangles. We apply the PTAS for stabbing rectangles [35] to $\mathcal{R}_\ell$, let $\mathcal{S}_\ell$ denote the computed set of segments. We will show later that the optimal solution for $\mathcal{R}_\ell$ is by

at most a factor $O(k/\delta)$ more expensive that the optimal solution for $\mathcal{K}_\ell$, and that this approximation ratio is good enough for our purposes in this step. Denote by $R_1$ and $R_2$ the connected components of $R \setminus \ell$ and by $\mathcal{S}_1$ and $\mathcal{S}_2$ the parts of segments from $\mathcal{S}$ that intersect $R_1$ and $R_2$, respectively. We define the candidate solution $\mathcal{S}_\ell \cup \mathrm{DP}(R_1, \mathcal{S}_1) \cup \mathrm{DP}(R_2, \mathcal{S}_2)$.

We store in $\mathrm{DP}(R, \mathcal{S})$ the candidate solution with the smallest cost. Finally, we output the solution stored in the cell $\mathrm{DP}([0, \alpha n] \times [0, 2kn], \emptyset)$.

## Analysis

We first note that all DP subproblems and operations are defined on discrete coordinates, and since there are only a polynomial $\frac{\alpha n}{\varepsilon^d} \times 2kn \leq 2\alpha k \varepsilon^{-3} n^3$ number of discrete points, the running time of the dynamic program is also polynomial.

▶ **Lemma 11.** *The running time of the above dynamic program is* $(kn/\varepsilon)^{O(1/\varepsilon^3)}$.

Our proof for bounding our approximation factor is similar to the analysis of the PTAS for rectangles [35] and our QPTAS in Section 3. We describe here its main structure and highlight the key differences.

The solution computed by our DP corresponds to performing a sequence of trivial, add, and *line* operations, and recursing on the respective subproblems. It is sufficient to argue that there exists a sequence of these operations such that

- there exists a DP cell for each arising subproblem; in particular, the number of line segments in each subproblem is bounded by $3\varepsilon^{-3}$, and
- the total cost of the computed solution is bounded by $(1 + O(\varepsilon))\mathrm{OPT}$.

We now describe this sequence. It is based on a hierarchical grid of vertices lines, shifted by a random offset $r \in \{0, \varepsilon^d, 2\varepsilon^d, \ldots, \alpha\varepsilon^{-2}\}$ that we will fix later. For each level $j \in \mathbb{N}_0$, we define a grid line $\{r + t \cdot \alpha\varepsilon^{j-2}\} \times \mathbb{R}$ for each $t \in \mathbb{Z}$. Note that for all $j \leq d+2$, grid lines of level $j$ have discrete $x$-coordinates. We say that a line segment $\ell \in \mathrm{OPT}$ is of *level* $j$ if the length of $\ell$ is in $(\alpha\varepsilon^j, \alpha\varepsilon^{j-1}]$. We say that a line segment of some level $j$ is *well-aligned* if its left and right endpoint lies on a grid line of level $j+3$, and if the $y$-coordinate of both endpoints is discrete. We can extend each line segment $\ell \in \mathrm{OPT}$ so that it becomes well-aligned, by increasing its length by at most a factor of $1 + O(\varepsilon)$.

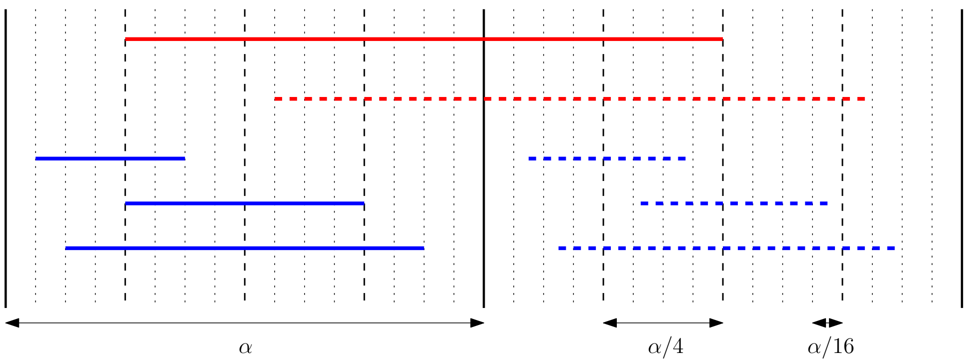


**Figure 5** For $\varepsilon = 1/4$, the figure shows vertical grid lines of level $j = 2, 3, 4$ (solid, dashed and dotted lines respectively). Horizontal segments of level $j = 0, 1$ (red and blue respectively) are shown where the solid segments are well-aligned, and the dashed ones are not.

▶ **Lemma 12.** *For any value of our offset, by losing a factor of* $1 + O(\varepsilon)$ *in our approximation ratio, we can assume that each line segment* $\ell \in \mathrm{OPT}$ *is well-aligned.*

Note that each horizontal segment $\ell \in \mathrm{OPT}$ satisfies that $\alpha\varepsilon/n < |\ell| \le \alpha\varepsilon^{-1}$. By our choice of $d$ we have $\varepsilon^{d-1} \le \varepsilon/n < \varepsilon^{d-2}$ which implies $\alpha\varepsilon^{d-1} < |\ell| \le \alpha\varepsilon^{-1}$. Since a segment is of level $j$ if its length is in the range $(\alpha\varepsilon^{j}, \alpha\varepsilon^{j-1}]$, we can conclude that all segments in OPT belong to levels in the range $\{0, \ldots, d-1\}$. 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.

In our sequence of operations, we first perform one *line operation* for each (vertical) grid line of level $j = 0$. This is similar to partitioning the instance into narrow strips as we did it in Lemma 5. However, now each strip has a width of $\alpha\varepsilon^{-2}$ instead of $w_{\max}(\mathcal{K})/\mu$. In our following operations, we add horizontal line segments to partition each vertical strip, similar to Section 3. Formally, we sort the segments from OPT of level $j = 0$ in increasing order of their $y$-coordinates, and pick every $(\varepsilon^{-3})$-th segment, and do an add operation along the (strip wide) line along it. This leads to a trivial operation immediately after that. Finally, we perform add operations for all line segments of level $j = 0$ in OPT. We call the above operations to be *operations of level* 0.

With the above operations for level $j = 0$ done, in increasing order of level $j = 1, 2, \ldots$ we do *operations of level* $j$ similarly as follows:

- *line operations* on vertical grid lines of level $j$,
- any valid trivial operations (this step is not done for level 0),
- add, and trivial operations to divide the vertical strips into smaller subproblems,
- and finally the add operations on the segments from OPT of level $j$,

mimicking the recursive structure from the analysis of the QPTAS.

▶ **Lemma 13.** *The above sequence of operations always leads to valid DP subproblems.*

We wish to bound the cost of the above operations. Suppose that we perform a line operation with a vertical line $\ell$ and let $\mathcal{K}_\ell$ denote the $k$-shapes that $\ell$ intersects. Recall that for each *line operation*, we compute a solution that stabs all $k$-shapes in $\mathcal{K}_\ell$ (and in fact every rectangle in $\mathcal{R}_\ell$). Note that any horizontal line segment $\ell' \in \mathrm{OPT}$ of some level $j' \ge j$ stabs a $k$-shape in $\mathcal{K}_\ell$ only if the distance between $\ell$ and $\ell'$ is at most $\alpha\varepsilon^{j-1}$. Another key insight is that since the ratio between the widest and the narrowest part of any $K \in \mathcal{K}_\ell$ is $1/\delta$, the solution we compute is also an $O(k/\delta + \varepsilon)$-approximate solution. Using the above facts, we claim that if we choose our offset $r$ uniformly at random from the range $\{0, \varepsilon^d, 2\varepsilon^2 d, \ldots, \alpha\varepsilon^{-2}\}$, then the overall cost of these *line operations* is only $O(\varepsilon) \cdot \mathrm{OPT}$. Further to bound the cost of the add operations, we note that each add operation is either done on a segment in OPT, or is an operation that created a subproblem. We will show that we can charge the latter operations to segments from OPT inside the subproblem thus created, whose total cost is at least $\varepsilon^{-1}$ times the width of the subproblem. We refer to the full version [34] for a formal description of our analysis.

▶ **Lemma 14.** *There is a discrete value for the offset* $r \in \{0, \varepsilon^d, 2\varepsilon^d, \ldots, \varepsilon^{-2}\}$ *such that the described sequence of operations produces a solution of cost at most* $(1 + O(\varepsilon))\mathrm{OPT}$.

▶ **Theorem 15.** *For each constant* $k \in \mathbb{N}$ *there is a PTAS for the* $k$*-*Stabbing *problem when each given* $k$*-shape consists of pieces of a constant range of widths.*

▶ Remark 16. In the proof of our result above, we used that the widths of the rectangles of each input $k$-shape are in a bounded range. Strictly speaking, we used only that the width "spanned" by each input $k$-shape is at most a constant factor larger than the width of the narrowest rectangle of the $k$-shape. Hence, the result will also hold for other polygons, like polyominoes with $O(1)$ number of cells, that are not $k$-shapes but that do satisfy the latter property.

## 5 General case

In this section, we study the general case of stabbing rectilinear polygons. Please refer to the full version [34] for the missing proofs, and details of the results in this section.

### 5.1 APX-hardness

In contrast to the cases studied in Sections 3 and 4, we show that the general case of the stabbing problem does *not* admit a $(1+\varepsilon)$-approximation algorithm, even for only slightly more general types of instances. Formally, we prove that stabbing is APX-hard, already if each input polygon is a 3-shape.

▶ **Theorem 17.** *The stabbing problem for 3-shapes is* APX*-hard.*

On the other hand, any 2-shape satisfies the hourglass property; hence, stabbing is unlikely to be APX-hard for this class of objects since we have a QPTAS for this case.

▶ **Proposition 18.** *Each 2-shape satisfies the hourglass property.*

In the remainder of this subsection, we prove Theorem 17. We give an L-reduction from the vertex cover problem to 3-Stabbing. Note that it is NP-hard to approximate vertex cover with a strictly better approximation factor than $\sqrt{2}$ [36]. We will obtain the same lower bound for stabbing.

Consider a given instance $G = (V, E)$ of vertex cover. Remember that in vertex cover, we are required to select a subset $S \subseteq V$ of smallest size such that for each $e \in E$ one of its end points is in $S$. We construct an instance of $k$-Stabbing corresponding to $G$ as follows. Assume that $V = \{v_1, \ldots, v_n\}$. For each $v_i \in V$ construct a $1 \times 1$ square $s_i$, such that they are all arranged in a column separated by 1 unit distance each (see Figure 6). Formally, for each $v_i \in V$ the top-left corner of the square $s_i$ has the coordinates $(0, 2i-1)$. Note that the squares $s_1, \ldots, s_n$ do *not* belong to our input shapes, but they only help us to construct the latter. For each edge $\{v_i, v_j\} \in E$ we define a 3-shape $r_{i,j}$ as the union of the three rectangles $s_i$, $[0, n+1] \times [2i-1, 2j-2]$ and $s_j$ (see Figure 6).

Note that none of the resulting shapes satisfies the hourglass property, and also for neither of them the widths of its three rectangles are in a constant range. The width of the widest rectangle of each constructed 3-shape is greater than $n$, but there is always a feasible solution with cost $n$ that simply stabs the square $s_i$ for each vertex $v_i \in V$. Thus, in any given solution to the stabbing instance, we can assume w.l.o.g. that no 3-shape is stabbed across its widest rectangle.

▶ **Lemma 19.** *For each $\gamma \in \mathbb{N}$, the given instance of vertex cover has a solution of size $\gamma$ if and only if the corresponding $k$-Stabbing instance has a solution of cost $\gamma$.*

**Proof.** We first show that if there is a vertex cover of size $\gamma$ then there is a solution of cost $\gamma$ to our instance of $k$-Stabbing. Given a solution $S = \{v_1, v_2, \ldots, v_\gamma\}$ to the vertex cover instance, construct a solution to the stabbing instance as follows: for each $v_i \in S$, stab the corresponding $s_i$ along its top edge by a segment of length one. Clearly the cost of this set of segments is $\gamma$. Now we notice that every $k$-shape $r_{i,j}$ corresponds to an edge $e(v_i, v_j)$ in the graph. Since this edge has been covered by one of its adjacent vertices $v_i \in S$, $r_{i,j}$ is also stabbed by the segment that stabs $s_i$. We know that every edge of the graph is covered by some vertex in $S$, and hence every $k$-shape in the instance is also stabbed in the solution we constructed.

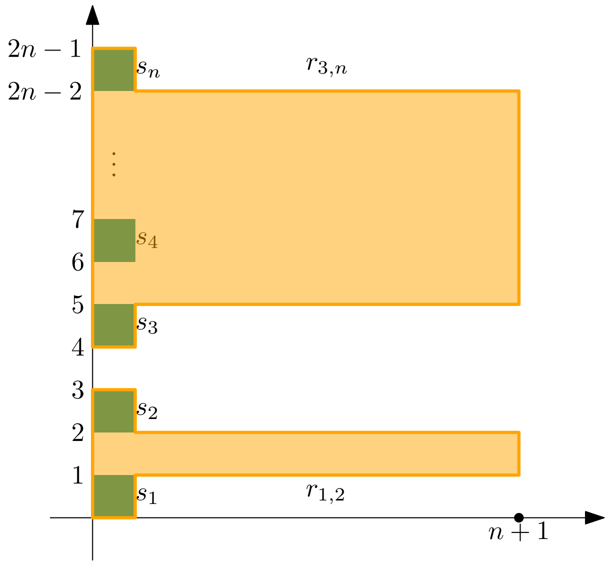


**Figure 6** Construction of $k$-STABBING instance in our reduction from vertex cover.

Next, we argue that a solution of cost $\gamma$ to our instance of $k$-STABBING yields a solution to vertex cover of size at most $\gamma$. Consider any solution to the stabbing instance of cost $\gamma$. We can assume that there are no segments of length greater than one in this solution, since any segment of length at least $n+1$, can be broken down into at most $n$ segments of length 1 stabbing the same set of $k$-shapes, but along their bordering squares; and segments of length in the range $(1, n+1)$ can stab only one $k$-shape, and hence be shortened to length one. Further, segments in any solution can also not be of length less than one, since such a segment cannot stab any $k$-shape. Hence we conclude that all segments in the solution are of length one, and by extension that they stab any $k$-shape along one of its bordering squares.

Now we construct a vertex cover solution by picking the vertices $v_i$, that correspond to any square $s_i$ that has been stabbed by the given (or modified as mentioned above) $k$-STABBING solution. Note that every $k$-shape is stabbed by the given solution, and hence a vertex adjacent to every edge in the vertex cover instance has been picked by us. This shows that the selected set is in fact a valid vertex set, and is of size at most $\gamma$. ◀

This yields the proof of Theorem 17.

## 5.2 Set Cover hardness

We further show that $k$-STABBING for arbitrary $k$-shapes cannot be approximated with a ratio of $o(\log n)$, unless $\mathsf{P} = \mathsf{NP}$. In fact, we show that the problem is as hard as general instances of SET COVER.

▶ **Theorem 20.** *The $k$-STABBING problem does not admit an $o(\log n)$-approximation algorithm, unless $\mathsf{P} = \mathsf{NP}$.*

The proof of the above theorem is a generalization of the proof of Theorem 17, and its details can be found in the full version [34].

## 5.3 Approximation algorithm

We show that there is a polynomial time $O(k)$-approximation algorithm for $k$-Stabbing. Given an instance $\mathcal{K}$ of $k$-Stabbing with $n := |\mathcal{K}|$, we first show that we can restrict ourselves to a polynomial number of line segments which we construct using the following lemma.

▶ **Lemma 21.** *In polynomial time, we can construct a set $\mathcal{C}$ of line segments with the following properties:*

- *$\mathcal{C}$ contains $O((kn)^3)$ segments,*
- *$\mathcal{C}$ contains no redundant segments, where a segment is* redundant *if it stabs exactly the same $k$-shapes as another segment, or no $k$-shapes at all, and*
- *$\mathcal{K}$ admits an optimal solution using only the segments from $\mathcal{C}$.*

Using $\mathcal{C}$, we define a linear program that corresponds to $\mathcal{K}$.

$$\begin{aligned} \min \sum_{s\in\mathcal{C}} |s|\cdot z_s & & \\ \text{s.t.} \sum_{s\in\mathcal{C}:\ s \text{ stabs } K} z_s \geq 1 & \qquad \forall K\in\mathcal{K} & \qquad (1)\\ z_s \geq 0 & \qquad \forall s\in\mathcal{C}. \end{aligned}$$

If each $k$-shape $K \in \mathcal{K}$ is a rectangle, then it was shown by Chan et al. [9] that this LP has a constant integrality gap. We prove that for arbitrary $k$-shapes it has an integrality gap of $O(k)$, and we give a corresponding polynomial time rounding algorithm, in which we use the result by Chan et al. [9] as a black-box.

▶ **Theorem 22** ([9])**.** *If each $k$-shape $K \in \mathcal{K}$ is a rectangle, then there is a constant $\alpha$ such that for any solution $z$ to LP* (1)*, in polynomial time we can compute an integral solution to* (1) *whose cost is at most $\alpha \sum_{s\in\mathcal{F}} |s|\dot{z}_s$.*

Using Theorem 22, we construct now an $(\alpha \cdot k)$-approximation algorithm for arbitrary $k$-shapes. Suppose we are given an optimal solution $z^*$ to the LP (1). We define a new solution $\tilde{z}$ by setting $\tilde{z}_s := k \cdot z^*_s$ for each segment $s \in \mathcal{F}$. Each $k$-shape $K \in \mathcal{K}$ is composed out of at most $k$ rectangles. Thus, for each $k$-shape $K \in \mathcal{K}$ there is one of these rectangles $R$ for which $\sum_{s\in\mathcal{F}:\ s \text{ stabs } R} z^*_s \geq 1/k$ and, therefore, $\sum_{s\in\mathcal{F}:\ s \text{ stabs } R} \tilde{z}_s \geq 1$. Let $\mathcal{R}$ denote the set of all these rectangles for all $k$-shapes in $\mathcal{K}$. We apply Theorem 22 on $\tilde{z}_s$ and $\mathcal{R}$ which yields a set of segments $\tilde{\mathcal{S}}$ whose cost is at most $\alpha\cdot\sum_{s\in\mathcal{F}} |s|\cdot\tilde{z}_s = \alpha k\cdot\sum_{s\in\mathcal{F}} |s|\cdot z^*_s \leq \alpha k\cdot\mathrm{OPT}$. Since $\tilde{\mathcal{S}}$ stabs $\mathcal{R}$, it also stabs $\mathcal{K}$. Hence, $\tilde{\mathcal{S}}$ yields an $O(k)$-approximation to our problem.

▶ **Theorem 23.** *There is a polynomial time $O(k)$-approximation algorithm for $k$-Stabbing.*

We remark that our algorithm extends also to the setting in which each given shape consists of at most $k$ rectangles that are not necessarily connected, but such that still at least one of them needs to be stabbed.

## References

1 Anna Adamaszek, Sariel Har-Peled, and Andreas Wiese. Approximation schemes for independent set and sparse subsets of polygons. *J. ACM*, 66(4):29:1–29:40, 2019. doi:10.1145/3326122.

2 Pankaj K. Agarwal, Hsien-Chih Chang, Subhash Suri, Allen Xiao, and Jie Xue. Dynamic geometric set cover and hitting set. *ACM Trans. Algorithms*, 18(4):40:1–40:37, 2022. doi:10.1145/3551639.

3 Boris Aronov, Esther Ezra, and Micha Sharir. Small-size $\varepsilon$-nets for axis-parallel rectangles and boxes. *SIAM J. Comput.*, 39(7):3248–3282, 2010. doi:10.1137/090762968.

4 Nikhil Bansal and Arindam Khan. Improved approximation algorithm for two-dimensional bin packing. In *SODA*, pages 13–25, 2014. doi:10.1137/1.9781611973402.2.

5 Hervé Brönnimann and Michael T. Goodrich. Almost optimal set covers in finite VC-dimension. *Discret. Comput. Geom.*, 14(4):463–479, 1995. doi:10.1007/BF02570718.

6 Timothy M. Chan and Elyot Grant. Exact algorithms and APX-hardness results for geometric packing and covering problems. *Comput. Geom.*, 47(2):112–124, 2014. doi:10.1016/J.COMGEO.2012.04.001.

7 Timothy M. Chan, Elyot Grant, Jochen Könemann, and Malcolm Sharpe. Weighted capacitated, priority, and geometric set cover via improved quasi-uniform sampling. In *SODA*, pages 1576–1585, 2012. doi:10.1137/1.9781611973099.125.

8 Timothy M. Chan, Qizheng He, Subhash Suri, and Jie Xue. Dynamic geometric set cover, revisited. In *SODA*, pages 3496–3528, 2022. doi:10.1137/1.9781611977073.139.

9 Timothy M. Chan, Thomas C. van Dijk, Krzysztof Fleszar, Joachim Spoerhase, and Alexander Wolff. Stabbing rectangles by line segments - how decomposition reduces the shallow-cell complexity. In *ISAAC*, pages 61:1–61:13, 2018. doi:10.4230/LIPICS.ISAAC.2018.61.

10 Kang-Tsung Chang. *Introduction to geographic information systems (4. ed.).* McGraw-Hill, 2008.

11 Henrik I. Christensen, Arindam Khan, Sebastian Pokutta, and Prasad Tetali. Approximation and online algorithms for multidimensional bin packing: A survey. *Computer Science Review*, 24:63–79, 2017. doi:10.1016/J.COSREV.2016.12.001.

12 Vasek Chvátal. A greedy heuristic for the set-covering problem. *Math. Oper. Res.*, 4(3):233–235, 1979. doi:10.1287/MOOR.4.3.233.

13 Aparna Das, Krzysztof Fleszar, Stephen G. Kobourov, Joachim Spoerhase, Sankar Veeramoni, and Alexander Wolff. Approximating the generalized minimum manhattan network problem. *Algorithmica*, 80(4):1170–1190, 2018. doi:10.1007/S00453-017-0298-0.

14 Emily Dinan, Alice Nadeau, and Isaac Odegard. Folding concave polygons into convex polyhedra: The L-shape. *Rose-Hulman Undergraduate Mathematics Journal*, 16(1):13, 2015.

15 Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *STOC*, pages 624–633, 2014. doi:10.1145/2591796.2591884.

16 Friedrich Eisenbrand, Martina Gallato, Ola Svensson, and Moritz Venzin. A QPTAS for stabbing rectangles. *arXiv*, 2021. arXiv:2107.06571.

17 Adeel Farooq, Mustafa Habib, Abid Mahboob, Waqas Nazeer, and Shin Min Kang. Zagreb polynomials and redefined zagreb indices of dendrimers and polyomino chains. *Open Chemistry*, 17(1):1374–1381, 2019. doi:10.1515/chem-2019-0144.

18 Uriel Feige. A threshold of ln $n$ for approximating set cover. *J. ACM*, 45(4):634–652, 1998. doi:10.1145/285055.285059.

19 Waldo Gálvez, Fabrizio Grandoni, Salvatore Ingala, Sandy Heydrich, Arindam Khan, and Andreas Wiese. Approximating geometric knapsack via L-packings. *ACM Trans. Algorithms*, 17(4):33:1–33:67, 2021. doi:10.1145/3473713.

20 Waldo Gálvez, Fabrizio Grandoni, Arindam Khan, Diego Ramírez-Romero, and Andreas Wiese. Improved approximation algorithms for 2-dimensional knapsack: Packing into multiple L-shapes, spirals, and more. In *SoCG*, pages 39:1–39:17, 2021. doi:10.4230/LIPICS.SOCG.2021.39.

21 Waldo Gálvez, Arindam Khan, Mathieu Mari, Tobias Mömke, Madhusudhan Reddy Pittu, and Andreas Wiese. A $(2+\epsilon)$-approximation algorithm for maximum independent set of rectangles. *arXiv*, 2021. arXiv:2106.00623.

22 M. R. Garey and David S. Johnson. The rectilinear steiner tree problem is NP complete. *SIAM Journal of Applied Mathematics*, 32:826–834, 1977. doi:10.1137/0132071.

23 M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979. doi:10.5555/574848.

**24** Daya Ram Gaur, Toshihide Ibaraki, and Ramesh Krishnamurti. Constant ratio approximation algorithms for the rectangle stabbing problem and the rectilinear partitioning problem. *J. Algorithms*, 43(1):138–152, 2002. doi:10.1006/JAGM.2002.1221.

**25** Solomon W. Golomb. Tiling with polyominoes. *Journal of Combinatorial Theory*, 1(2):280–296, 1966. doi:10.1016/S0021-9800(66)80033-9.

**26** Solomon W. Golomb. *Polyominoes: puzzles, patterns, problems, and packings*, volume 111. Princeton University Press, 1996. doi:10.2307/j.ctv10vm1sc.

**27** Rolf Harren, Klaus Jansen, Lars Prädel, and Rob van Stee. A $(5/3 + \epsilon)$-approximation for strip packing. *Comput. Geom.*, 47(2):248–267, 2014. doi:10.1016/J.COMGEO.2013.08.008.

**28** Klaus Jansen, Arindam Khan, Marvin Lira, and K. V. N. Sreenivas. A PTAS for packing hypercubes into a knapsack. In *ICALP*, pages 78:1–78:20, 2022. doi:10.4230/LIPICS.ICALP.2022.78.

**29** Iwan Jensen and Anthony J. Guttmann. Statistics of lattice animals (polyominoes) and polygons. *Journal of Physics A: Mathematical and General*, 33(29):L257, 2000. doi:10.1088/0305-4470/33/29/102.

**30** Arindam Khan, Aditya Lonkar, Arnab Maiti, Amatya Sharma, and Andreas Wiese. Tight approximation algorithms for two-dimensional guillotine strip packing. In *ICALP*, pages 80:1–80:20, 2022. doi:10.4230/LIPICS.ICALP.2022.80.

**31** Arindam Khan, Aditya Lonkar, Saladi Rahul, Aditya Subramanian, and Andreas Wiese. Online and dynamic algorithms for geometric set cover and hitting set. In *SoCG*, pages 46:1–46:17, 2023. doi:10.4230/LIPICS.SOCG.2023.46.

**32** Arindam Khan, Arnab Maiti, Amatya Sharma, and Andreas Wiese. On guillotine separable packings for the two-dimensional geometric knapsack problem. In *SoCG*, pages 48:1–48:17, 2021. doi:10.4230/LIPICS.SOCG.2021.48.

**33** Arindam Khan and Eklavya Sharma. Tight approximation algorithms for geometric bin packing with skewed items. *Algorithmica*, 85(9):2735–2778, 2023. doi:10.1007/S00453-023-01116-0.

**34** Arindam Khan, Aditya Subramanian, Tobias Widmann, and Andreas Wiese. On approximation schemes for stabbing rectilinear polygons. *arXiv*, 2024. doi:10.48550/arXiv.2402.02412.

**35** Arindam Khan, Aditya Subramanian, and Andreas Wiese. A PTAS for the horizontal rectangle stabbing problem. In *IPCO*, pages 361–374, 2022. doi:10.1007/978-3-031-06901-7_27.

**36** Subhash Khot, Dor Minzer, and Muli Safra. Pseudorandom sets in grassmann graph have near-perfect expansion. In *FOCS*, pages 592–601, 2018. doi:10.1109/FOCS.2018.00062.

**37** Sofia Kovaleva and Frits C. R. Spieksma. Approximation algorithms for rectangle stabbing and interval stabbing problems. *SIAM J. Discret. Math.*, 20(3):748–768, 2006. doi:10.1137/S089548010444273X.

**38** Jens Lienig and Juergen Scheible. *Fundamentals of layout design for electronic circuits*. Springer, 2020. doi:10.1007/978-3-030-39284-0.

**39** Mario Alberto López and Dinesh P. Mehta. Efficient decomposition of polygons into L-shapes with application to VLSI layouts. *ACM Trans. Design Autom. Electr. Syst.*, 1(3):371–395, 1996. doi:10.1145/234860.234865.

**40** George Martin. *Polyominoes: A guide to puzzles and problems in tiling*. Cambridge University Press, 1991. doi:10.1080/00029890.1993.11990425.

**41** Joseph S. B. Mitchell. Approximating maximum independent set for rectangles in the plane. In *FOCS*, pages 339–350, 2021. doi:10.1109/FOCS52979.2021.00042.

**42** Nabil H. Mustafa, Rajiv Raman, and Saurabh Ray. Settling the apx-hardness status for geometric set cover. In *FOCS*, pages 541–550, 2014. doi:10.1109/FOCS.2014.64.

**43** Thomas Ottmann, Eljas Soisalon-Soininen, and Derick Wood. On the definition and computation of rectilinear convex hulls. *Information Sciences*, 33(3):157–171, 1984. doi:10.1016/0020-0255(84)90025-2.

**44** Helmut Pottmann, Andreas Asperl, Michael Hofer, Axel Kilian, and Daril Bentley. *Architectural geometry*, volume 724. Bentley Institute Press Exton, 2007. doi:10.1016/j.cag.2014.11.002.

45 Peter Shirley, Michael Ashikhmin, and Steve Marschner. *Fundamentals of computer graphics*. AK Peters/CRC Press, 2009. doi:10.5555/1628957.

46 Kasturi R. Varadarajan. Weighted geometric set cover via quasi-uniform sampling. In *STOC*, pages 641–648, 2010. doi:10.1145/1806689.1806777.

47 Chris Worman and J. Mark Keil. Polygon decomposition and the orthogonal art gallery problem. *Int. J. Comput. Geom. Appl.*, 17(02):105–138, 2007. doi:10.1142/S0218195907002264.

48 Bei Yu, Jhih-Rong Gao, and David Z. Pan. L-shape based layout fracturing for e-beam lithography. In *ASP-DAC*, pages 249–254, 2013. doi:10.1109/ASPDAC.2013.6509604.