**Lines:** 486

# Approximating the Maximum Independent Set of Convex Polygons with a Bounded Number of Directions

**Fabrizio Grandoni**
IDSIA, USI-SUPSI, Lugano, Switzerland

**Edin Husić**
IDSIA, USI-SUPSI, Lugano, Switzerland

**Mathieu Mari**
LIRMM, University of Montpellier, CNRS, Montpellier, France

**Antoine Tinguely**
IDSIA, USI-SUPSI, Lugano, Switzerland

## Abstract

In the maximum independent set of convex polygons problem, we are given a set of $n$ convex polygons in the plane with the objective of selecting a maximum cardinality subset of non-overlapping polygons. Here we study a special case of the problem where the edges of the polygons can take at most $d$ fixed directions. We present an $8d/3$-approximation algorithm for this problem running in time $O((nd)^{O(d4^d)})$. The previous-best polynomial-time approximation (for constant $d$) was a classical $n^\varepsilon$ approximation by Fox and Pach [SODA'11] that has recently been improved to a $\mathrm{OPT}^\varepsilon$-approximation algorithm by Cslovjecsek, Pilipczuk and Węgrzycki [SODA '24], which also extends to an arbitrary set of convex polygons.

Our result builds on, and generalizes the recent constant factor approximation algorithms for the maximum independent set of axis-parallel rectangles problem (which is a special case of our problem with $d = 2$) by Mitchell [FOCS'21] and Gálvez, Khan, Mari, Mömke, Reddy, and Wiese [SODA'22].

**2012 ACM Subject Classification** Theory of computation → Packing and covering problems; Theory of computation → Computational geometry

**Keywords and phrases** Approximation algorithms, packing, independent set, polygons

**Related Version** *Full version*: https://arxiv.org/abs/2402.07666

**Funding** *Fabrizio Grandoni*: Partially supported by the Swiss National Science Foundation (SNSF) Grant 200021 200731/1.
*Edin Husić*: Supported by the Swiss National Science Foundation (SNSF) Grant 200021 200731/1.
*Antoine Tinguely*: Supported by the Swiss National Science Foundation (SNSF) Grant 200021 200731/1.

## 1 Introduction

The Maximum Independent Set of Convex Polygons problem (MISP) is a natural geometric packing problem with many applications in map labeling [13, 40], cellular networks [35], unsplittable flow [6], chip manufacturing [28], or data mining [18, 34]. Given a set of $n$ convex polygons in the plane, the goal is to select a maximum number of them such that the polygons are pairwise non-overlapping.

MISP is NP-hard [16, 29], hence it makes sense to design approximation algorithms for it. Disappointingly, the best (polynomial-time) approximation ratio for MISP (more precisely for $k$-intersecting curves) has been $n^\varepsilon$ [17], for any fixed constant $\varepsilon > 0$. This ratio has recently been improved to $\mathrm{OPT}^\varepsilon$ [12].


40th International Symposium on Computational Geometry (SoCG 2024).
Editors: Wolfgang Mulzer and Jeff M. Phillips; Article No. XX; pp. XX:1–XX:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

**Approximation Schemes.** Interestingly, there is a quasi-polynomial time approximation scheme (QPTAS) for MISP [1]. Thus, the problem is *not* APX-hard, assuming NP $\nsubseteq$ DTIME($2^{\text{polylog}(n)}$), suggesting that it should be possible to obtain a polynomial time approximation scheme (PTAS) for the problem.

If we assume that we are allowed to shrink the polygons by a factor $1-\delta$ for an arbitrarily small constant $\delta$, then there is a PTAS for the problem [41]. Note that here the output is compared to the optimal solution without shrinking.

When the input polygons are fat, e.g., regular polygons, then PTASes are known [9, 15].

**Axis-parallel rectangles.** A prominent special case of MISP that has attracted a lot of attention over the years is the maximum independent set of axis-parallel rectangles (MISR), where all the polygons are rectangles with their edges parallel with the axes. An $O(\log n)$ approximation for MISR was given in [31, 39]. This was slightly improved to $O(\log n/\log\log n)$ in [10], and substantially improved to $O(\log\log n)$ in [7]. In a recent breakthrough result, Mitchell [37] presented the first constant factor approximation algorithm with approximation ratio 10, and later $3+\varepsilon$ in an updated version [38] with a considerably shorter case analysis. Subsequently, his approach was simplified and improved to a $(2+\varepsilon)$-approximation algorithm by Gálvez, Khan, Mari, Mömke, Reddy, and Wiese [21, 22]. These approaches rely on a dynamic program that considers all the partitions of a bounding box containing the instance into a number of containers with constant complexity (constant number of line segments).

**Our contribution.** With the goal of better understanding the approximability of MISP, in this paper, we consider the following natural special case of MISP: $d$-MISP is the special case of MISP where the edges of the input polygons are parallel to a given set $\mathcal{D}$ of $d = |\mathcal{D}|$ directions. Notice that MISR is equivalent to 2-MISP. Our main result is a constant approximation for $d$-MISP when $d$ is a constant.

▶ **Theorem 1.** *There exists an* $8d/3$*-approximation algorithm for* $d$*-MISP running in time* $O((nd)^{O(d4^d)})$.

Our result builds on the approaches in [21, 22, 38], however we have to face several additional complications. In particular, already for $d = 3$ the algorithm and its analysis deviates substantially from the known (polynomial-time) results in the literature about axis-aligned rectangles. An overview of our approach is given in Section 3.

**Related Work.** One can consider a natural weighted version of MISP, where each convex polygon has a positive weight, and the goal is to find an independent set of maximum total weight. The weighted version of MISR was studied in the literature, and the current-best polynomial time approximation factor is $O(\log\log n)$ [8]. We remark that our approach, likewise the approaches in [21, 22, 37], does not seem to extend to the weighted case. In particular, finding a constant approximation for weighted MISR remains a challenging open problem. We remark that the QPTAS in [1] extends to the weighted case, hence suggesting that the weighted version of MISP might also admit a PTAS.

MISR was also studied in terms of parameterized algorithms. Marx [36] proved that the problem is W[1]-hard, which rules out the existence of an EPTAS. A parameterized approximation scheme for the problem is given in [24].

A rectangle packing problem related to MISR is the 2D Knapsack problem. Here we are given an axis-parallel square (the *knapsack*) and a collection of axis-parallel rectangles. The goal is to pack a maximum cardinality (or weight) subset of rectangles in the knapsack

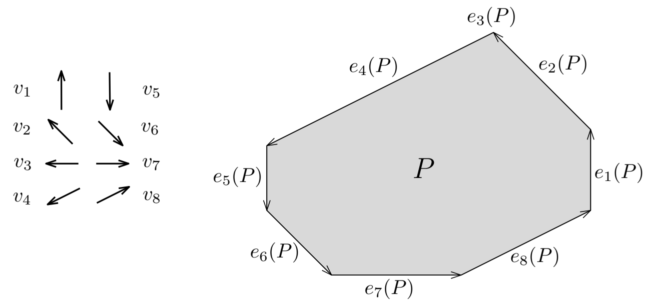



**Figure 1** A convex polygon in 4 directions. The edge $e_3(P)$ is degenerate.

(without rotations). 2D Knapsack admits a QPTAS [2] and a few constant approximation algorithms are known [19, 20, 30]. Here as well, finding a PTAS is a challenging open problem.

Bonsma et al. [6] established an intriguing connection between MISR and the Unsplittable Flow on a Path problem. A PTAS for the latter problem was recently obtained [25], closing a very long line of research (see, e.g., [3, 4, 5, 6, 26, 27]).

## 2 Preliminaries

In this paper, a (possibly closed) *curve* is always assumed to be a polygonal chain (or a singleton point) and a *polygon* $S$ is a bounded set with non-empty interior $\text{int}(S)$ and whose *boundary* $\partial S$ is a closed curve. We denote the *closure of* $S$ as $\bar{S}$, so $\bar{S} = \partial S \cup \text{int}(S)$. We say that two polygons $S, T$ (with non-empty interior) *touch* if $\text{int}(S) \cap \text{int}(T) = \emptyset$ but $\partial S \cap \partial T \neq \emptyset$ and *intersect* if $\text{int}(S) \cap \text{int}(T) \neq \emptyset$. A curve $f$ *touches* $S$ if $f \cap \text{int}(S) = \emptyset$ but $f \cap \partial S \neq \emptyset$.

A line segment or curve is called *degenerate* if it is a singleton point. A line segment or curve is assumed to be non-degenerate unless we explicitly state the opposite. For an (oriented) line segment $e = \overline{ww'}$ (resp. curve $\gamma = w_1 w_2 \cdots w_k$) we define the *head of* $e$ *(of* $\gamma$*)* as $h(e) = w'$ ($h(\gamma) = w_k$) and the *tail of* $e$ *(of* $\gamma$*)* as $t(e) = w$ ($t(\gamma) = w_1$) and the *interior of* $e$ *(of* $\gamma$*)* as $\text{int}(e) = e \setminus \{h(e), t(e)\}$ ($\text{int}(\gamma) = \gamma \setminus \{h(\gamma), t(\gamma)\}$). For a degenerate line segment (resp. curve), the head and the tail coincide with the line segment (resp. curve).

For a vector $v = (x, y)$, let $v^\perp := (y, -x)$ (which is $v$ rotated clockwise orthogonally). For a positive integer $k$, let $[k] := \{1, \ldots, k\}$.

**Input.** For a fixed positive integer $d$, the input of our problem is given by a set of (pairwise linearly independent) $d$ *direction defining vectors* $\mathcal{D} = \{v_1, \ldots, v_d\} \subseteq \mathbb{Z}^2$ and a set $\mathcal{I}$ of $n$ convex polygons with edges oriented along the directions given in $\mathcal{D}$. Polygons of this type are sometimes called *d-discrete orientation polytopes (d-DOPs)* [32]. In this paper, we will more casually refer to them as (input) polygons; the significance of the word "polygon" will be clear from context. Without loss of generality, assume $v_1 = (0, 1)$ and that $v_2, \ldots, v_d$ point to the left and are ordered by decreasing slope, see Figure 1. For $i \in \{d+1, \ldots, 2d\}$, let $v_i := -v_{i-d}$. The indices of the directions are counted modulo $2d$, i.e., $i = i + 2d = i - 2d$. More explicitly, each polygon $P \in \mathcal{I}$ is encoded by $2d$ integers $p_1(P), \ldots, p_{2d}(P)$ as $P = \{x \in \mathbb{R}^2 : \ x^\intercal v_i^\perp < p_i(P), \ \forall i \in [2d]\}$; and thus $\bar{P} = \{x \in \mathbb{R}^2 : x^\intercal v_i^\perp \leq p_i(P), \ \forall i \in [2d]\}$. We assume that those linear inequalities are all tight, including redundant ones[1], i.e., $e_i(P) \coloneqq \bar{P} \cap \{x : x^\intercal v_i^\perp = p_i(P)\} \neq \emptyset$ for every $i \in [2d]$. $e_i(P)$ is called the *edge of $P$ in direction $v_i$*. Then, for every $i \in [2d]$, $e_i(P)$ and $e_{i+1}(P)$ are incident and $h(e_i(P)) = t(e_{i+1}(P))$. Note that $e_1(P)e_2(P)\cdots e_{2d}(P)$ forms a positively oriented closed curve.

**Grid.** Let $\mathcal{L}_1$ be the set of all lines in directions $v_1, \ldots, v_d$ passing through the vertices of the input polygons. In particular, all the edges (including the degenerate ones) of all the polygons in the input lie on the lines in $\mathcal{L}_1$. Notice that $|\mathcal{L}_1| \leq 2d^2n$. We recursively define $\mathcal{V}_k$, for $k \in [2d]$ and $\mathcal{L}_k$, for $k \in \{2, \ldots, 2d\}$ as follows: $\mathcal{V}_k$ is the set of intersection points of any two (non-parallel) lines in $\mathcal{L}_k$, and $\mathcal{L}_k$ is the set of all lines in directions $\mathcal{D}$ passing through points in $\mathcal{V}_{k-1}$. We define the grid $\mathcal{G}_k = (\mathcal{L}_k, \mathcal{V}_k)$. Since $|\mathcal{V}_k| \leq |\mathcal{L}_k|^2$ and $|\mathcal{L}_k| \leq |\mathcal{V}_{k-1}| \cdot d$, it follows that $|\mathcal{V}_k| \leq (2d^3n)^{2^k}$. The grid $\mathcal{G}_{2d}$ form the coordinate system of our algorithm: every geometric object appearing in the algorithm and the analysis lies on $\mathcal{G}_{2d}$. A line segment $s$ *lies on* $\mathcal{G}_k$ if $s$ lies on some line in $\mathcal{L}_k$ and the extreme points of $s$ lie on $\mathcal{V}_k$. Similarly, a curve or polygon lies on $\mathcal{G}_k$ if all of its line segments do so.

**Container.** Consider the grid $\mathcal{G}_1$. Let $C^* \in \mathcal{G}_1$ be a parallelogram that encloses all polygons in $\mathcal{I}$; we call $C^*$ the *bounding box*.[2] A *container* (see Figure 2(a)) is a polygon on $\mathcal{G}_{2d}$ with positively oriented boundary $s_1 f_1 s_2 f_2 \ldots s_\kappa f_\kappa$ where $2 \leq \kappa \leq 5$, such that:

- $s_1, s_2, \ldots, s_\kappa$ are disjoint and possibly degenerate *parallel* line segments on $\mathcal{G}_{2d}$ (these will later be called *cutting lines*).
- For all $j \in [\kappa]$, $f_j$ is a simple curve on $\mathcal{G}_{2d}$ consisting of at most $2d+1$ line segments and $t(f_j) = h(s_j)$ and $h(f_j) = t(s_{j+1})$ for every $j \in [\kappa]$ (where $s_{\kappa+1} = s_1$).
- For all $j \in [\kappa]$, $\mathrm{int}(s_j)$ does not intersect with any other part of the boundary of the container.
- For all $i, j \in [\kappa], i \neq j$, the curves $f_i$ and $f_j$ might touch but do not cross (defined below).

In particular, a container has at most $10d+10$ line segments. Let $\mathcal{C}$ be the set of all containers $C$ with $\mathrm{int}(C) \subseteq \mathrm{int}(C^*)$. In particular, $C^*$ is a container and $C^* \in \mathcal{C}$. A *bipartition of $C \in \mathcal{C}$* is a pair $\{C_1, C_2\} \subseteq \mathcal{C}$ such that $C_1, C_2$ split up $C$, i.e., $\mathrm{int}(C) \setminus (\partial C_1 \cup \partial C_2) = \mathrm{int}(C_1) \cup \mathrm{int}(C_2)$ and $C_1$ and $C_2$ may touch but not intersect.

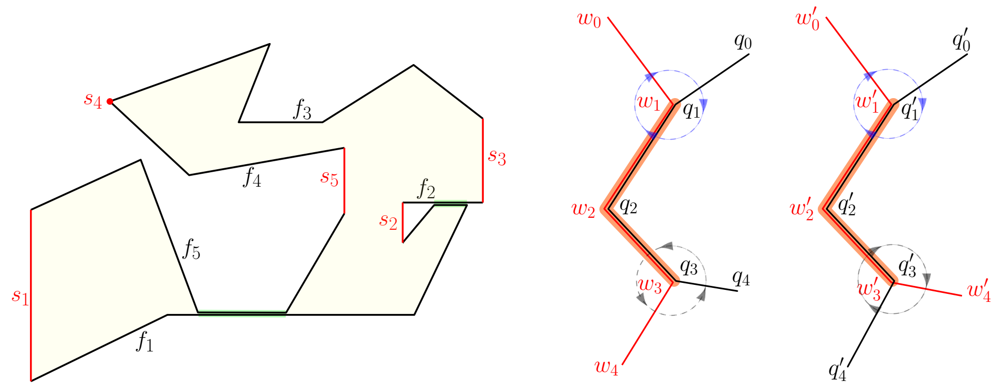
**Crossing curves.** Two curves *cross* (see also Figure 2(b)) if each one of them contains a connected subcurve $w_0 w_1 \cdots w_k$ and $q_0 q_1 \cdots q_k$, respectively, which form a *crossing*, i.e., if $w_0 \neq q_0$, $w_k \neq q_k$, $w_i = q_i$ for $1 \leq i \leq k-1$ and the (non-collinear) triangles $w_0 q_0 w_2$ and $w_t q_t w_{t-2}$ have the same orientation (i.e., are either both positively or both negatively oriented).[3] For two curves formed by at most $k$ line segments in total, it can be decided in time $O(k^3)$ whether there exists a crossing among them or not [11]. With this definition, it is guaranteed that every container has a well-defined interior [11].

The proofs and details which are omitted due to space constraints will appear in the full version of the paper (see also [23]).

[1] An inequality is *redundant* if we can remove it from the definition of $P$ without affecting $P$.

[2] It can, for example, be chosen as a parallelogram delimited by the leftmost and rightmost vertical lines and the top and bottom $v_2$-oriented lines in $\mathcal{G}_1$ (i.e., the extension of $e_2(P')$ where $P' = \arg\max_{P \in \mathcal{I}} p_2(P)$ and the extension of $e_{d+2}(P'')$ where $P'' = \arg\max_{P \in \mathcal{I}} p_{d+2}(P)$).

[3] Any container is thus *weakly simple* according to the definitions in [14, Box 5.1] and [33]. The concept of weakly simple polygons is extensively discussed in [11].




**(a)** A container with $\kappa = 5$. The line segment $s_4$ on the boundary of the container is degenerate. The curves $f_1$ and $f_5$, as well as $f_1$ and $f_2$, respectively, touch on the green segments but do not cross.

**(b)** The curves on the left touch without crossing: the triangles $w_0q_0w_2$ and $w_4q_4w_2$ have negative and positive orientation, respectively. The curves on the right cross: the triangles $w'_0q'_0w'_2$ and $w'_4q'_4w'_2$ are both negatively orientated.

**Figure 2** A container with $\kappa = 5$. An illustration of crossing and non-crossing.

# 3 Our Approach

First, we present the algorithm in Section 3.1, and give an overview of the analysis in Sections 3.2 and 3.3. The detailed analysis and proofs are given in the later sections.

## 3.1 The algorithm

Our algorithm is a dynamic program that generalizes the algorithm in [21]. Each cell of the dynamic program corresponds to a container $C \in \mathcal{C}$. For each container, the dynamic program computes a set of disjoint polygons $\mathrm{Dyn}(C) \subseteq \mathcal{I}$ as follows. If $C$ encloses no polygon in $\mathcal{I}$, set $\mathrm{Dyn}(C) = \emptyset$. If $C$ encloses exactly one polygon $P \in \mathcal{I}$, set $\mathrm{Dyn}(C) = \{P\}$. Otherwise, the dynamic program goes through all bipartitions of $C$ and chooses the bipartition $\{C_1, C_2\}$ that maximizes $|\mathrm{Dyn}(C_1)| + |\mathrm{Dyn}(C_2)|$ and sets $\mathrm{Dyn}(C) = \mathrm{Dyn}(C_1) \cup \mathrm{Dyn}(C_2)$. The final output of the algorithm is $\mathrm{Dyn}(C^*)$.

▶ **Lemma 2** (Running time). *Let $N = |\mathcal{V}_{2d}|$ be the number of points in the grid $\mathcal{G}_{2d}$. $\mathrm{Dyn}(C^*)$ can be computed in time $O\big(N^{20d+20}\big) = O((nd)^{O(d4^d)})$.*

**Proof.** The boundary of each container can be identified by a sequence of $10d + 10$ line segments in $\mathcal{G}_{2d}$. There are therefore at most $O\big(N^{10d+10}\big)$ containers in $\mathcal{C}$. As argued in [21], any bipartition $\{C_1, C_2\}$ of $\mathcal{C}$ is determined by the boundary between $C_1$ and $C_2$, i.e., $\partial C_1 \cap \partial C_2$, which is composed of at most $10d + 10$ line segments. Thus, to compute $\mathrm{Dyn}(C)$, the dynamic program does not consider more that $O\big(N^{10d+10}\big)$ bipartitions. This gives a total running time $O\big(N^{20d+20}\big)$. The lemma follows since $N = O((2d^3n)^{4^d})$, see Section 2. ◀

It is not hard to see that the output $\mathrm{Dyn}(C^*)$ is indeed an independent set, so we will focus on showing that the algorithm has the claimed approximation guarantee.

## 3.2 Analysis

By construction, the output solution $\mathrm{Dyn}(C^*)$ is the union of the solutions of two smaller containers, and so on. We represent this structure by a binary tree called *recursive partition* defined below. We argue that $\mathrm{Dyn}(C^*)$ is the best solution among all the solutions representable by a recursive partition. Then, we show the existence of a recursive partition that respects the approximation factor claimed in Theorem 1.

▶ **Definition 3.** *For a set $\mathcal{R} \subseteq \mathcal{I}$, a* recursive partition *of $\mathcal{R}$ is a rooted tree $T$ with vertex set $V$ such that*

- *every node $u \in V$ corresponds to a pair $(C_u, \mathrm{Pr}(C_u))$ where $C_u \in \mathcal{C}$ is a container, and $\mathrm{Pr}(C_u)$ is the set of* protected *polygons of $\mathcal{R}$ contained in $C_u$,*
- *the root $r$ of $T$ corresponds to $(C^*, \emptyset)$, i.e., $C_r = C^*$ and $\mathrm{Pr}(C_r) = \emptyset$;*
- *every internal node has two children $u_1, u_2$ such that: $C_{u_1}$ and $C_{u_2}$ form a bipartition of $C_u$, and $\mathrm{Pr}(C_u) \subseteq \mathrm{Pr}(C_{u_1}) \cup \mathrm{Pr}(C_{u_2})$;*
- *for every leaf $u$ of $T$, $C_u$ contains exactly one polygon $P_u \in \mathcal{R}$ or no polygon in $\mathcal{R}$ at all;*
- *for every $P \in \mathcal{R}$, there exists a leaf $u$ of $T$ such that $P$ lies in $C_u$.*

Clearly, if $\mathcal{R} \subseteq \mathcal{I}$ admits a recursive partition, it must be an independent set. It is easy to show by induction on the height of the tree that the output $\mathrm{Dyn}(C^*)$ admits a recursive partition, which leads to the following lemma.

▶ **Lemma 4** ([21, Lemma 2.2])**.** *If $\mathcal{R} \subseteq \mathcal{I}$ admits a recursive partition, then $|\mathrm{Dyn}(C^*)| \geq |\mathcal{R}|$.*

Therefore, Theorem 1 is a consequence of Lemma 2 and the following proposition.

▶ **Proposition 5.** *Let* OPT *be an optimal solution of an instance of MISP. There exists a recursive partition for some set $\mathcal{R} \subseteq \mathrm{OPT}$ such that $|\mathcal{R}| \geq \frac{3}{8d}|\mathrm{OPT}|$.*

## 3.3 Informal overview of the proof of Proposition 5

Intuitively, we construct the set $\mathcal{R}$ by starting from an optimal solution OPT contained in the initial container (the bounding box) $C_r = C^*$ and $\mathrm{Pr}(C_r) = \emptyset$. Then, we will recursively partition the current container $C_u$ into two containers $C_{u_1}$ and $C_{u_2}$. $\mathcal{R}$ is then defined as the set of polygons of OPT that are fully contained in the leaf containers. For a polygon $P \in \mathrm{OPT}$ contained in $C_u$, we say that $P$ is *lost (at $C_u$)* if it is neither contained in $C_{u_1}$ nor in $C_{u_2}$.

Below, one of the $d$ directions in $\mathcal{D}$ plays a special role: without loss of generality, we assume that this direction is vertical/vertical-up ($v_1$). The exact choice will be made later.

**Accountable polygons.** We prove that there exists a subset $\mathrm{ACC} \subseteq \mathrm{OPT}$ (the *accountable* polygons) with at least $\frac{3}{4d}|\mathrm{OPT}|$ polygons, such that for each polygon $P \in \mathrm{ACC}$ lost during partitioning of some $C_u$ into $C_{u_1}$ and $C_{u_2}$ we can *charge* an unique polygon $P' \in \mathrm{OPT}$ and $P'$ lies in a leaf container of the recursive partition.

We next describe in more details the set of accountable polygons ACC and how protected polygons are defined. For technical reasons, we replace each original polygon $P \in \mathrm{OPT}$ with a new polygon $\mathrm{ext}(P)$ lying on $\mathcal{G}_{2d}$ that contains $P$ (see Figures 3 and 4). The new set of polygons remains independent, and we will simply denote it by OPT in the following.

Let $P \in \mathrm{OPT}$ and consider its edge $e_1(P)$ in direction vertical-up. Let $P' \in \mathrm{OPT}$ and consider its edge $e_{d+1}(P')$ in direction vertical-down. We say that $P$ *sees* $P'$ if $e_1(P)$ is non-degenerate and $h(e_{d+1}(P')) \in \mathrm{int}(e_1(P)) \cup \{t(e_1(P))\}$, see Figure 4. We let the set ACC of accountable polygons be the polygons $P \in \mathrm{OPT}$ such that $P$ sees some $P' \in \mathrm{OPT}$. It is easy to show that each polygon is seen by at most one other polygon in OPT.

**Partitioning.** For $C \in \mathcal{C}$, let OPT($C$) be the set of polygons in OPT that lie on int($C$). Our construction is guided by a partitioning lemma which is stated later. Roughly speaking, let $C$ be a container with $|\operatorname{OPT}(C)| \geq 2$, and let Pr($C$) be the set of protected polygons in $C$. The partitioning lemma states that $C$ can be bipartitioned by a curve $\Gamma$ into two smaller containers $C_1$ and $C_2$ such that

**(P1)** $\Gamma$ contains a vertical line segment $\ell$ that intersects all the polygons in OPT($C$) that are intersected by $\Gamma$.

**(P2)** $\Gamma$ does not intersect any polygon in Pr($C$),

**(P3)** $\operatorname{Pr}(C) \subseteq \operatorname{Pr}(C_1) \cup \operatorname{Pr}(C_2)$.

We stress that the lemma does *not* hold for an arbitrary set Pr($C$) (e.g., if we take $\operatorname{Pr}(C) = \operatorname{OPT}(C)$). The set of *protected* polygons in a container is defined below.

**Charging and protecting.** The recursive partition which determines $\mathcal{R}$ is defined by repeatedly applying the partitioning lemma. During the construction of the recursive partition, we need to guarantee that the vertical line segments given by (P1) do not intersect too many polygons from OPT; this is the only possibility of "losing" some polygons. For this, we use the set of accountable polygons $\operatorname{ACC} \subseteq \operatorname{OPT}$. Whenever we apply the partitioning lemma, the line $\ell$ intersects some polygons in ACC. For each $P \in \operatorname{ACC}$ that is intersected by $\ell$, i.e., for each lost polygon $P \in \operatorname{ACC}$, we charge exactly one polygon $P'$ seen by $P$. By (P1), if $\ell$ intersects $P$, then $\Gamma$ does not intersect $P'$. If $P'$ is not already an element of Pr($C$) and thus an element of $\operatorname{Pr}(C_1) \cup \operatorname{Pr}(C_2)$, then we add the polygon $P'$ to either $\operatorname{Pr}(C_1)$ if $P' \in C_1$ or to $\operatorname{Pr}(C_2)$ if $P' \in C_2$. Moreover, if there is a polygon $P'' \in \operatorname{OPT}(C)$ that sees $P$, then $P''$ is also added to either $\operatorname{Pr}(C_1)$ or $\operatorname{Pr}(C_2)$.

By (P3), adding $P'$ to one of $\operatorname{Pr}(C_1)$ and $\operatorname{Pr}(C_2)$ means that the charged polygon $P'$ will remain protected. By (P2), $P'$ will not be intersected by the curves in the following applications of the partitioning lemma. Therefore $P'$ will be an element in $\mathcal{R}$ (our intended recursive partition). Adding $P''$ to one of $\operatorname{Pr}(C_1)$ and $\operatorname{Pr}(C_2)$ is also necessary, because the polygon $P$ is already lost and if we were to lose $P''$ in one of the following steps, there might not be a polygon which we could charge the loss of $P''$ to.

We conclude that for every polygon $P \in \operatorname{ACC}$ lost in the partitioning of a container, we can guarantee that a unique polygon $P'$ seen by $P$ is charged, and it will become the protected polygon in a leaf. At least half of the polygons in ACC are either lost or not, so there are at least $\frac{1}{2}|\operatorname{ACC}|$ polygons in the leaves. Proposition 5 follows since $|\operatorname{ACC}| \geq \frac{3}{4d}|\operatorname{OPT}|$.

## 3.4 Comparison with previous work on MISR

Overall, we follow the same high level approach as the papers on MISR [21, 22, 38]. Yet, to generalize the results on MISR to MISP, we encounter several technical difficulties. We discuss a few of the more prominent ones below.

To define the set ACC, we need the following property (later referred as (E3)): for every $P \in \operatorname{OPT}$ and every non-degenerate edge $e$ of $P$, int($e$) touches either another polygon $P' \in \operatorname{OPT}$ or the boundary of the bounding box. This property can be obtained by "maximally extending" OPT as in [21, 38]. The difficulty here, unlike in the case of rectangles, is that naively extending the polygons can result in a grid of exponential size in $n$.

For MISR [21, 38], the accountable polygons correspond to the non-nested polygons (both vertical and horizontal). It is essentially trivial to show that the number of non-nested rectangles is at least half of the optimal number of rectangles. In case of convex polygons, we require a more careful argument to show that there are at least $\frac{3}{4d}|\operatorname{OPT}|$ accountable polygons.

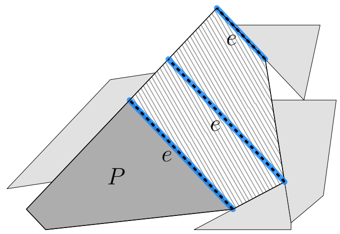



**Figure 3** Illustration of the process of extending a polygon $P$. We extend $P$ by moving the edge $e$ of $P$ until $\text{int}(e)$ touches another polygon in OPT.

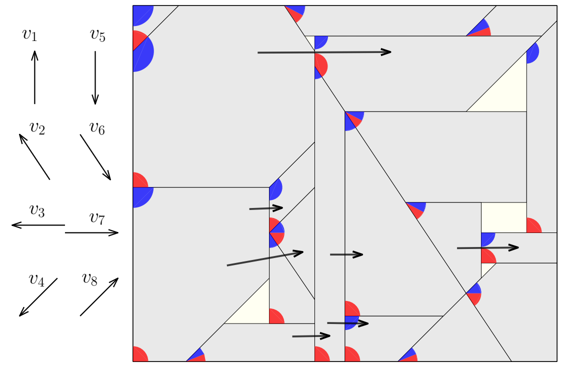



**Figure 4** A black arrow from $P$ to $P'$ indicates that $P$ sees $P'$ with respect to the option $(v_1, t)$, i.e., direction vertical-up and tail. The blue (resp. red) corners represent the tails (resp. head) of all edges with direction vertical-down ($v_5$). Thus, a polygon $P$ sees a polygon $P'$ if the vertical-up edge of $P$ is touching the red corner of $P'$.

To obtain the partitioning lemma, we follow the same idea as in the case of axis-parallel rectangles but we need to work with significantly more complex objects. Firstly, the containers we work with have $O(d)$-times more line segments. Secondly, the containers that appear in our construction might not be simple (since some parts of the boundary may touch other parts of the boundary). These difficulties require more elaborate and more technical arguments.

## 4 Charging options and accountable polygons

Like the papers [21, 38] on MISR, first, we extend an optimum solution OPT.

▶ **Definition 6.** *Let* OPT *be an optimal solution of a MISP instance. We say that* $\text{OPT}'$ *is a* maximal extension *of* OPT *if:*

**(E1)** $\text{OPT}'$ *is an independent set of (convex) polygons on* $\mathcal{G}_{2d}$ *and enclosed in* $C^*$*.*

**(E2)** *There exists a bijection* $\text{ext} : \text{OPT} \to \text{OPT}'$ *such that* $P \subseteq \text{ext}(P)$ *for every* $P \in \text{OPT}$.

**(E3)** *For every* $P \in \text{OPT}'$ *and every non-degenerate edge* $e$ *of* $P$*,* $\text{int}(e)$ *touches either another polygon* $P' \in \text{OPT}'$ *or* $\partial C^*$*.*

On a high level, a maximal extension is constructed as follows: starting with OPT, one direction $v_i$ at a time, as long as there is a polygon $P \in \text{OPT}$ with $e_i(P)$ being non-degenerate but not satisfying (E3), we extend $P$ by moving the edge $e_i(P)$ "outside" (i.e., by steadily increasing $p_i(P)$), see Figure 3. After the extension in the $k$-th direction, the edges of polygons in OPT lie on $\mathcal{G}_k$, so the maximal extension lies on the grid $\mathcal{G}_{2d}$.

By (E2) and (E1), it suffices to prove Proposition 5 for a maximal extension of OPT. (In particular, (E1) implies that the polygons in $\text{OPT}'$ have edges in the given $d$ directions.) The purpose of a maximal extension is to guarantee (E3), which is helpful to bound the number of accountable polygons. For the rest of the paper, we assume that OPT is already "maximally extended" and thus satisfies (E3), and we work with the grid $\mathcal{G}_{2d}$.

In the rest of this section, by the term *direction* we mean a direction $v_i$ where $i \in [2d]$, and say that edge $e$ is of direction $v_i$ if the points of the edge $e$ correspond to $t(e) + \lambda \cdot v_i$, with $\lambda \geq 0$. A *charging option* is specified by a direction $v_i$, $i \in [2d]$ and a choice between $t$ and $h$. Let $\mathcal{O} = \{v_i\}_{i \in [2d]} \times \{t, h\}$ be the set of the $2d \cdot 2 = 4d$ charging options. We show the existence of a charging option and a subset $\mathrm{ACC} \subseteq \mathrm{OPT}$ of accountable polygons with respect to this option such that (essentially) $|\mathrm{ACC}| \geq \frac{3}{4d}|\mathrm{OPT}|$.

▶ **Definition 7.** *Let $P \in \mathrm{OPT}$ and let $e$ be the edge of $P$ in direction $v = v_i$, $i \in [2d]$.*

- *Let $P' \in \mathrm{OPT}$ and $e'$ be the (possibly degenerate) edge of $P'$ of direction $-v$. For $a \in \{t, h\}$, we say that $P$* sees *$P'$ with respect to $(v, a)$ if $e$ is non-degenerate and if $\neg a(e') \in \mathrm{int}(e) \cup \{a(e)\}$, where $\neg t = h$ and $\neg h = t$. (See Figure 4.)*
- *Whenever there exists $P' \in \mathrm{OPT}$ and a charging option $(v, a)$, such that $P$ sees $P'$ for $(v, a)$ then we say that $P$ is* accountable for *$(v, a)$.*

▶ **Lemma 8.** *Let $(v, a) \in \mathcal{O}$ be a charging option. Any polygon $P' \in \mathrm{OPT}$ is seen by at most one other polygon $P \in \mathrm{OPT}$ with respect to $(v, a)$.*

**Proof.** Assume that $P'$ is seen by $P_1, P_2 \in \mathrm{OPT}$ with respect to $(v, a)$. Let $e_1$ and $e_2$ be the edge in direction $v$ of $P_1$ and $P_2$, respectively. Then we have $\neg a(e') \in (\mathrm{int}(e_1) \cup \{a(e_1)\}) \cap (\mathrm{int}(e_2) \cup \{a(e_2)\})$. Since $\mathrm{int}(e_1) \neq \emptyset$ and $\mathrm{int}(e_2) \neq \emptyset$, it follows that $\mathrm{int}(e_1) \cap \mathrm{int}(e_2) \neq \emptyset$. This implies that $P_1$ and $P_2$ intersect, thus $P_1 = P_2$. ◀

We say that a polygon $P \in \mathrm{OPT}$ is a *corner polygon* in the bounding box $C^*$, if all but one of the edges of $P$ are contained in the boundary of $C^*$. In particular, $P$ is a corner polygon if $P = C^*$. Similarly, if $C^*$ is partitioned into two convex polygons, then both are corner polygons. Let $Z \subseteq \mathrm{OPT}$ be the set of corner polygons in $C^*$. Since $C^*$ is a parallelogram, we have $|Z| \leq 4$, and the polygon $C' = C^* \setminus (\bigcup Z)$ is convex.

▶ **Lemma 9** (Good charging option)**.** *Assume that $\mathrm{OPT}$ satisfies (E3). Then, there exists a charging option $(v, a) \in \mathcal{O}$ such that at least $\frac{3}{4d}|\mathrm{OPT} \setminus Z|$ polygons in $\mathrm{OPT} \setminus Z$ are accountable with respect to $(v, a)$.*

**Proof.** Let $P \in \mathrm{OPT}$ and $c$ be a vertex of $P$. Let $e, e'$ be the two non-degenerate edges incident to $c$ where $c = h(e) = t(e')$. Denote with $v$ (resp. $v'$) the direction of $e$ (resp. $e'$).

▷ Claim 10. Suppose that $e$ or $e'$ (or both) does not lie on the boundary of $C^*$. Then, $P$ is accountable with respect to $(v, h)$ or $(v', t)$.

Proof. By (E3), each non-degenerate edge of $P$ not contained in the boundary of the bounding box, must touch some other polygon of OPT in its interior. By assumption either $e$ or $e'$ does not lie on the boundary of $C^*$, without loss of generality, say $e$. Then $P$ touches some $P_1 \in \mathrm{OPT}$ on $\mathrm{int}(e)$, i.e., $\mathrm{int}(e) \cap e_1 \neq \emptyset$, where $e_1$ is the edge of $P_1$ in direction $-v$ ($e_1$ could be degenerate). See Figure 5. If $P$ sees $P_1$ with respect to $(v, h)$, i.e., $t(e_1) \in \mathrm{int}(e) \cup \{h(e)\}$ then the claim is true, so assume that $t(e_1) \notin \mathrm{int}(e) \cup \{h(e)\}$. This however implies $c \in \mathrm{int}(e_1)$.

Since $c \in \mathrm{int}(e_1)$ and $C^*$ is convex, it follows that $e'$ is not on the boundary of $C^*$. Then, by (E3), there exists $P_2 \in \mathrm{OPT}$ that touches $P$ on $\mathrm{int}(e')$, i.e., $\mathrm{int}(e') \cap e_2 \neq \emptyset$, where $e_2$ is the edge of $P_2$ in direction $-v'$. If $P$ does not see $P_2$ with respect to $(v', t)$, then $c \in \mathrm{int}(e_2)$ by the same argument as before. So $\mathrm{int}(e_1)$ and $\mathrm{int}(e_2)$ intersect in $c$ and thus $P_1$ and $P_2$ intersect (as $e_1$ and $e_2$ have different direction) which is a contradiction. Therefore, $P$ must see $P_2$ with respect to $(v', t)$. ◁

Consider $P \in \mathrm{OPT} \setminus Z$. Since $P$ is not a corner polygon in $C^*$, it has at least two consecutive non-degenerate edges such that neither of them lies on $\partial C^*$. By Claim 10, every vertex of $P$ incident to one or both of these edges, provides a charging option for which $P$ is accountable. Thus, the total number of pairs $(P, (v, a))$ with $P \in \mathrm{OPT} \setminus Z$ and $(v, a) \in \mathcal{O}$ such that $P$ is accountable with respect to $(v, a)$ is at least $3|\mathrm{OPT} \setminus Z|$. Since $|\mathcal{O}| = 4d$, there

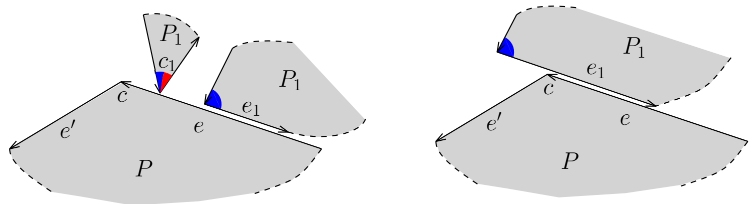



**Figure 5** Claim 10: the blue (red) corners represents the tail (head) of the edges in direction $-v$.

exists an option $(v, a)$ for which the number of accountable polygons in $\mathrm{OPT} \setminus Z$ is at least $\frac{3}{4d}|\mathrm{OPT} \setminus Z|$.[4] ◀

## 5 Recursive partitioning

Without loss of generality (by rotating and mirroring the initial instance if necessary), we assume that the option $(v, a)$ satisfying Lemma 9 is vertical-up and tail, i.e., $(v_1, t)$. In other words, for any $P \in \mathrm{OPT}$, if $e_1(P)$ is non-degenerate and if there is a $P' \in \mathrm{OPT}$ such that $h(e_{d+1}(P')) \in \mathrm{int}(e_1(P)) \cup t(e_1(P))$, then we say that $P$ *sees* $P'$ (and $P'$ *is seen by* $P$) and that $P$ is *accountable*. Lemma 9 states that there exists a subset $\mathrm{ACC} \subseteq \mathrm{OPT} \setminus Z$ of accountable polygons such that $|\mathrm{ACC}| \geq \frac{3}{4d}|\mathrm{OPT} \setminus Z|$, consequently $|Z| + |\mathrm{ACC}| \geq \frac{3}{4d}|\mathrm{OPT}|$.

We will construct a recursive partition for a specific subset $\mathcal{R} \subseteq \mathrm{OPT}$, such that $|\mathcal{R}| \geq |Z| + \frac{1}{2}|\mathrm{ACC}|$, which proves Proposition 5. Recall that $\mathrm{OPT}(C)$ denotes the set of polygons in OPT that lie on $\mathrm{int}(C)$. Moreover, all of the polygons in OPT and the bounding box $C^*$ lie on the grid $\mathcal{G}_{2d}$.

**Handling corner polygons.** If $Z \neq \emptyset$, then we construct the first few nodes of the recursive partition as follows. Take any corner polygon $P \in Z$. Recall that the root $r$ of the recursive partition corresponds to $(C^*, \emptyset)$. We add two children $u_1, u_2$ to $r$ and partition $C^*$ into the containers $C_{u_1} = P$ and $C_{u_2} = C^* \setminus P$. Set $\mathrm{Pr}(C_{u_1}), \mathrm{Pr}(C_{u_2}) = \emptyset$. By construction, $\mathrm{OPT}(C_{u_1}) = \{P\}$ (so $u_1$ is a leaf in the final tree and $\mathrm{OPT}(C_{u_2}) = \mathrm{OPT} \setminus \{P\}$. Notice that $C^* \setminus P$ is convex with at most five line segments since $C^*$ is convex. $C^* \setminus P$ has five line segments if $P$ is a triangle, and less if $P$ has more than three sides.) We recurse by treating $C_{u_2}$ as the new bounding box.

We end up with a tree on $|Z| + 1$ leaves, where for one leaf $u$, $C_u$ is a convex polygon such that $\mathrm{OPT}(C_u) = \mathrm{OPT} \setminus Z$ and with at most eight line segments (since $|Z| \leq 4$) and $\mathrm{Pr}(C_u) = \emptyset$. Each of the remaining $|Z|$ leaves coincides with a unique element in $Z$. Thus, it suffices to construct the recursive partition of $\mathrm{OPT} \setminus Z$ by treating $C_u$ as the bounding box with at most 8 line segments. Equivalently, we assume $Z = \emptyset$ and allow $C^*$ to have up to eight line segments for the rest of this paper.

### 5.1 The partitioning lemma – formal statement

For any $P \in \mathrm{OPT}$, let the *top of* $P$ be defined as the curve $\mathrm{top}(P) = e_2(P)e_3(P) \cdots e_d(P)$ and the *bottom of* $P$ as the curve $\mathrm{bot}(P) = e_{d+2}(P)e_{d+3}(P) \cdots e_{2d}(P)$. We define the bottom

---

[4] If we could guarantee a maximal extension in which all the polygons have at least 4 sides, then we would improve $\frac{3}{4d}$ to $\frac{1}{d}$. In particular, when $d = 2$ we are in the case of axis-parallel rectangles and we obtain a $2d = 4$-approximation algorithm. This is the same approximation factor achieved in [21, 22, 38] by charging each lost rectangle to one protected rectangle (the improved $2 + \varepsilon$ factor requires a more complex charging).

and top of the bounding box $C^*$ in the same way. The following definitions are illustrated in Figure 6.

▶ **Definition 11** (Top and bottom fences). *Let $P, P' \in \mathrm{OPT}$ be two polygons such that $P$ sees $P'$. A* top-fence *is (a segment of) the curve* $\mathrm{top}(P)\overline{h(e_1(P))t(e_{d+1}(P'))}\,\mathrm{top}(P')$ *such that the first and last line segment is not vertical. Symmetrically, a* bottom-fence *is (a segment of) the curve* $\mathrm{bot}(P)\overline{t(e_1(P))h(e_{d+1}(P'))}\,\mathrm{bot}(P')$ *such that the first and last line segment is not vertical.*

*If $P \in \mathrm{OPT}$ does not see any polygon, then a segment of its bottom (or top) is also called a bottom-fence (resp. top-fence).*

*For a vertical line segment (cutting line) $s$, we say that a fence* emerges from $s$ *if one extreme point of the fence lies on $s$.*

To prove the partitioning lemma, we further specialize the definition of a container (see Section 2)

▶ **Definition 12** (Structured container). *A container $C$ with $\partial C = s_1 f_1 s_2 f_2 \cdots s_\kappa f_\kappa$, $\kappa \leq 5$, is* structured *if the* cutting lines *$s_1, \ldots, s_\kappa$ are vertical and the curves $f_1, \ldots, f_\kappa$ are fences.*

*We say that a cutting line is a* left cutting line *if it is oriented downwards (or degenerate), and* right cutting line *if it is oriented upwards (or degenerate). In a structured container, the left cutting lines (and thus right cutting lines) are consecutive (e.g., $s_1, \ldots, s_{\kappa'}$ are left and $s_{\kappa'+1}, \ldots, s_\kappa$ are right cutting lines for some $\kappa' \in [\kappa - 1]$).*

▶ **Definition 13** (Protected by fences). *Let $C$ be a structured container and $s$ be a (possibly degenerate) cutting line on $C$. We say that a polygon $P \in \mathrm{OPT}(C)$ is* protected from the left in $C$ via $s$ *if $s$ is a left cutting line on $\partial C$ and*

- *there exists a top-fence $\gamma_h$ in $C$ emerging from $s$, ending in $h(e_1(P))$, and with* $\mathrm{top}(P) \subseteq \gamma_h$, *and*
- *there exists a bottom-fence $\gamma_t$ in $C$ emerging from $s$, ending in $t(e_1(P))$, and with* $\mathrm{bot}(P) \subseteq \gamma_t$.

*We say that $P$ is protected by fences $\gamma_h$* and *$\gamma_t$. Symmetrically, we say that a polygon $P \in \mathrm{OPT}(C)$ is* protected from the right in $C$ via $s$ *if $s$ is a right cutting line on $\partial C$ and*

- *there exists a top-fence $\sigma_h$ in $C$ emerging from $s$, ending in $t(e_{d+1}(P))$, and with* $\mathrm{top}(P) \subseteq \sigma_h$, *and*
- *there exists a bottom-fence $\sigma_t$ in $C$ emerging from $s$, ending in $h(e_{d+1}(P))$, and with* $\mathrm{bot}(P) \subseteq \sigma_t$.

*We say that $P$* is protected by fences *$\sigma_h$* and *$\sigma_t$. A polygon $P \in \mathrm{OPT}(C)$ is* protected by fences in $C$ *if it is either protected from the left in $C$ or protected from the right in $C$.*

We will show that each polygon in $\mathrm{Pr}(C)$ appearing in the construction of the recursive partition can be protected *by fences* in $C$, beginning by stating the partitioning lemma. The lemma holds only for *structured* containers, which matters for the construction of the recursive partition but it does not affect the algorithm, as it considers all possible containers.

▶ **Lemma 14** (Partitioning lemma). *Let $C$ be a structured container such that $|\mathrm{OPT}(C)| \geq 2$, and let $\mathcal{P}$ be a set of polygons in $C$ protected by fences. Then, there exists a curve $\Gamma$ such that*

**(P1)** *$\Gamma$ partitions $C$ into two structured containers $C_1, C_2 \in \mathcal{C}$ with non-empty interiors.*

**(P2)** *All the polygons in $\mathrm{OPT}(C)$ that are intersected by $\Gamma$ are intersected by one vertical cutting line $\ell \subseteq \Gamma$.*

**(P3)** *$\Gamma$ does not intersect any polygon protected by fences.*

**(P4)** *Any polygon protected by fences in $C$ is protected by fences in either $C_1$ or $C_2$.*

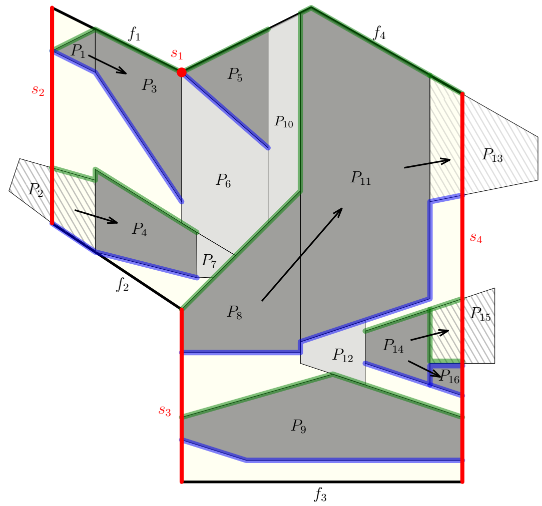



**Figure 6** Example of a structured container with $\kappa = 4$. The black arrows represent "seeing", top-fences are green, bottom-fences are blue. The polygons $P_1, P_3, P_4, P_5, P_8$ are protected (only) from the left, $P_{14}, P_{16}$ are protected (only) from the right, $P_9, P_{11}$ are protected both from the left and from the right. Notice that the fences that protect $P_{14}$ (from the right) are not unique since $P_{14}$ sees $P_{15}$ and $P_{16}$ which are cut and touch $s_4$, respectively. Note also that the bottom-fences touching $P_8, P_{11}$ and $P_{11}, P_{13}$ overlap.

## 5.2 Construction and analysis of the recursive partition

In this section we prove Proposition 5, i.e., we provide a recursive partition for $\mathcal{R} \subseteq \mathrm{OPT}$ with $|\mathcal{R}| \geq \frac{1}{2}|\mathrm{ACC}|$. (Recall that we already argued that we can assume $Z = \emptyset$.) We give an iterative construction of a recursive partition with the help of the partitioning lemma.

We initialize a tree $T$ with root node $r$, $C_r = C^*$, and $\mathrm{Pr}(C_r) = \emptyset$. Then, iteratively, for every childless node $u \in V(T)$ with $|\mathrm{OPT}(C_u)| \geq 2$, add two children $u_1, u_2$ to $u$ and choose $C_{u_1}, C_{u_2} \in \mathcal{C}$ as provided by (P1) in the partitioning lemma applied to $C_u$ and $\mathrm{Pr}(C_u)$. Define the set of protected polygons $\mathrm{Pr}(C_{u_1})$ and $\mathrm{Pr}(C_{u_1})$ as follows.

**(A1)** Set $\mathrm{Pr}(C_{u_1}) = \mathrm{Pr}(C_u) \cap \mathrm{OPT}(C_{u_1})$ and $\mathrm{Pr}(C_{u_1}) = \mathrm{Pr}(C_u) \cap \mathrm{OPT}(C_{u_1})$.

**(A2)** For each $P \in \mathrm{ACC}$ that is intersected by $\ell$, i.e., each $P \in \mathrm{ACC}$ that is lost, if $P$ sees a polygon $P' \in \mathrm{OPT}(C_u)$ (if $P$ sees more than one polygon in $\mathrm{OPT}(C_u)$, choose one of them arbitrarily), add $P'$ to $\mathrm{Pr}(C_{u_1})$ if $P'$ is in $C_{u_1}$ or to $\mathrm{Pr}(C_{u_2})$ if $P'$ is in $C_{u_2}$. Moreover, charge the loss of $P$ to $P'$.

**(A3)** For each $Q' \in \mathrm{OPT}(C_u)$ intersected by $\ell$ for which there is a polygon $Q \in \mathrm{OPT}(C_u)$ that sees $Q'$, add $Q$ to either $\mathrm{Pr}(C_{u_1})$ or $\mathrm{Pr}(C_{u_2})$ depending whether $Q$ is in $C_{u_1}$ or $C_{u_2}$.

We first show to that by this construction, a polygon is protected only if it is protected by fences.

▶ **Lemma 15.** *Let $P' \in \mathrm{Pr}(C_u)$ for a node $u$ of $T$. There exist fences that protect $P'$ in $C_u$.*

**Proof.** We first argue in the case that $P'$ is protected for the first time, i.e., added to $\mathrm{Pr}(C_u)$ via (A2) or (A3). Let $u'$ be the parent of $u$ in $T$.

First assume that $P'$ is protected via (A2). Let $P \in \mathrm{ACC} \cap \mathrm{OPT}(C_{u'})$ be the polygon that sees $P'$. By definition, $P$ is intersected by the cutting line $\ell_{u'}$ from (P1) during the bipartitioning of $C_{u'}$ Let $p_x$ and $p_y$ be the two intersection points of $\ell_{u'}$ and $\partial P$, where $p_x$ is above $p_y$, see Figure 7. Since $P$ sees $P'$, the curve $\gamma_x$ on $\mathrm{top}(P)$ and $\mathrm{top}(P')$ from $p_x$

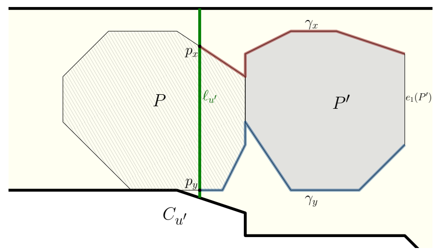



**Figure 7** Illustration for the proof of Lemma 15: $P'$ is protected by fences via (A2).

to $h(e_1(P'))$ is a top-fence and the curve $\gamma_y$ on $\text{bot}(P)$ and $\text{bot}(P')$ from $p_y$ to $t(e_1(P'))$ is a bottom-fence. $\gamma_x$ and $\gamma_y$ both emerge from $\ell_{u'}$ and thus protect $P'$ from the left in $C_u$. Hence, $P'$ is protected by fences in $C_u$.

The argument is symmetric if $P'$ is protected via (A3): there is a polygon $Q \in \text{OPT}(C_{u'})$ seen by $P'$ that is intersected by the the cutting line $\ell_{u'}$. Therefore, the curves on $\text{top}(P')$ and $\text{top}(Q)$ from $e_{d+1}(P')$ to $\ell_{u'}$ and of $\text{bot}(P')$ and $\text{bot}(Q)$ from $e_{d+1}(P')$ to $\ell_{u'}$ form a pair of fences that protect $P'$ from the right in $C_u$.

If $P'$ is protected via (A1), then it has been protected for the first time in an ancestor of $u$, so the claim follows inductively from by (P3) and (P4). ◀

With (P3) and (P4), Lemma 15 implies that protected polygons are not lost and stay protected, i.e., $\text{Pr}(C_u) \subseteq \text{Pr}(C_{u_1}) \cup \text{Pr}(C_{u_2})$ for every interior node $u$ in $T$. This in particular holds for every charged polygon. By the construction above, every charged polygon is protected and charged only once by Lemma 8. To make our charging scheme work, we need to make sure that *every* lost accountable polygon provides one charge, which follows by (P2) and the following lemma.

▶ **Lemma 16.** *Let $P \in \text{ACC}$ be a polygon that is intersected by the vertical line segment $\ell_u$ for an internal node $u \in T$. Then there exists a polygon $P' \in \text{OPT}(C_u)$ that is seen by $P$.*

**Proof.** Let $\mathcal{P}$ be the set of polygons seen by $P$. For the sake of contradiction, suppose that $\mathcal{P} \cap \text{OPT}(C_u) = \emptyset$. If some $P' \in \mathcal{P}$ partially lies in $C_u$, i.e., $P' \cap \text{int}(C_u) \neq \emptyset$, then $P'$ was intersected by the vertical line $\ell_{u'}$ in an ancestor $u'$ of $u$, so $P$ is protected via (A3). Otherwise, if all polygons in $\mathcal{P}$ lie outside of $C_u$, then $e_1(P)$ lies on a cutting line in $\partial C_u$. Therefore, $\text{top}(P)$ and $\text{bot}(P)$ form a top-fence and a bottom-fence, respectively, that protect $P$ by fences in $C_u$. ◀

**Proof of Proposition 5.** By Lemma 9, we have $|\text{ACC}| - |Z| \geq \frac{3}{4d}|\text{OPT}| - |Z|$. Recall that we have already assigned each polygon of $Z$ to a unique leaf of $T$. By the charging scheme described above and since a protected (and thus charged) polygon is never lost, we have a unique polygon contained in a leaf of $T$ for each lost accountable polygon during the partition. The proposition follows since at least half of the polygons in ACC are either lost, or at least half of the polygons in ACC are not lost. ◀

## 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** Anna Adamaszek and Andreas Wiese. A quasi-ptas for the two-dimensional geometric knapsack problem. In Piotr Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1491–1505. SIAM, 2015. doi:10.1137/1.9781611973730.98.

**3** Aris Anagnostopoulos, Fabrizio Grandoni, Stefano Leonardi, and Andreas Wiese. A mazing $2+\varepsilon$ approximation for unsplittable flow on a path. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 26–41. SIAM, 2014. doi:10.1137/1.9781611973402.3.

**4** Nikhil Bansal, Amit Chakrabarti, Amir Epstein, and Baruch Schieber. A quasi-ptas for unsplittable flow on line graphs. In Jon M. Kleinberg, editor, *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 721–729. ACM, 2006. doi:10.1145/1132516.1132617.

**5** Nikhil Bansal, Zachary Friggstad, Rohit Khandekar, and Mohammad R. Salavatipour. A logarithmic approximation for unsplittable flow on line graphs. *ACM Trans. Algorithms*, 10(1):1:1–1:15, 2014. doi:10.1145/2532645.

**6** Paul S. Bonsma, Jens Schulz, and Andreas Wiese. A constant-factor approximation algorithm for unsplittable flow on paths. *SIAM J. Comput.*, 43(2):767–799, 2014. doi:10.1137/120868360.

**7** Parinya Chalermsook and Julia Chuzhoy. Maximum independent set of rectangles. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 892–901. SIAM, 2009. URL: http://dl.acm.org/citation.cfm?id=1496770.1496867.

**8** Parinya Chalermsook and Bartosz Walczak. Coloring and maximum weight independent set of rectangles. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 860–868. SIAM, 2021. doi:10.1137/1.9781611976465.54.

**9** Timothy M Chan. Polynomial-time approximation schemes for packing and piercing fat objects. *Journal of Algorithms*, 46(2):178–189, 2003.

**10** Timothy M. Chan and Sariel Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. *Discret. Comput. Geom.*, 48(2):373–392, 2012. doi:10.1007/s00454-012-9417-5.

**11** Hsien-Chih Chang, Jeff Erickson, and Chao Xu. Detecting weakly simple polygons. In *Proceedings of the twenty-sixth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1655–1670. SIAM, 2014.

**12** Jana Cslovjecsek, Michał Pilipczuk, and Karol Węgrzycki. A polynomial-time $opt^{\varepsilon}$-approximation algorithm for maximum independent set of connected subgraphs in a planar graph. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 625–638. SIAM, 2024.

**13** Leila De Floriani, Paola Magillo, and Enrico Puppo. Applications of computational geometry to geographic information systems. *Handbook of computational geometry*, 7:333–388, 2000.

**14** Erik D Demaine and Joseph O'Rourke. *Geometric folding algorithms: linkages, origami, polyhedra*. Cambridge university press, 2007.

**15** Thomas Erlebach, Klaus Jansen, and Eike Seidel. Polynomial-time approximation schemes for geometric intersection graphs. *SIAM J. Comput.*, 34(6):1302–1323, 2005. doi:10.1137/S0097539702402676.

**16** Robert J Fowler, Michael S Paterson, and Steven L Tanimoto. Optimal packing and covering in the plane are np-complete. *Information processing letters*, 12(3):133–137, 1981.

**17** Jacob Fox and János Pach. Computing the independence number of intersection graphs. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1161–1165. SIAM, 2011. doi:10.1137/1.9781611973082.87.

**18** Takeshi Fukuda, Yasuhiko Morimoto, Shinichi Morishita, and Takeshi Tokuyama. Data mining with optimized two-dimensional association rules. *ACM Transactions on Database Systems (TODS)*, 26(2):179–213, 2001.

**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 *37th International Symposium on Computational Geometry (SoCG)*, volume 189, pages 39:1–39:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.SoCG.2021.39.

**21** Waldo Gálvez, Arindam Khan, Mathieu Mari, Tobias Mömke, Madhusudhan Reddy Pittu, and Andreas Wiese. A 3-approximation algorithm for maximum independent set of rectangles. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 894–905. SIAM, 2022. doi:10.1137/1.9781611977073.38.

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

**23** Fabrizio Grandoni, Edin Husić, Mathieu Mari, and Antoine Tinguely. Approximating the maximum independent set of convex polygons with a bounded number of directions. *arXiv preprint arXiv:2402.07666*, 2024.

**24** Fabrizio Grandoni, Stefan Kratsch, and Andreas Wiese. Parameterized approximation schemes for independent set of rectangles and geometric knapsack. In Michael A. Bender, Ola Svensson, and Grzegorz Herman, editors, *27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany*, volume 144 of *LIPIcs*, pages 53:1–53:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ESA.2019.53.

**25** Fabrizio Grandoni, Tobias Mömke, and Andreas Wiese. A PTAS for unsplittable flow on a path. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 289–302. ACM, 2022. doi:10.1145/3519935.3519959.

**26** Fabrizio Grandoni, Tobias Mömke, and Andreas Wiese. Unsplittable flow on a path: The game! In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 906–926. SIAM, 2022. doi:10.1137/1.9781611977073.39.

**27** Fabrizio Grandoni, Tobias Mömke, Andreas Wiese, and Hang Zhou. A $(5/3+\epsilon)$-approximation for unsplittable flow on a path: placing small tasks into boxes. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 607–619. ACM, 2018. doi:10.1145/3188745.3188894.

**28** Dorit S Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and vlsi. *Journal of the ACM (JACM)*, 32(1):130–136, 1985.

**29** Hiroshi Imai and Takao Asano. Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane. *Journal of algorithms*, 4(4):310–323, 1983.

**30** Klaus Jansen and Guochuan Zhang. On rectangle packing: maximizing benefits. In J. Ian Munro, editor, *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 204–213. SIAM, 2004. URL: http://dl.acm.org/citation.cfm?id=982792.982822.

**31** Sanjeev Khanna, S. Muthukrishnan, and Mike Paterson. On approximating rectangle tiling and packing. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 384–393. ACM/SIAM, 1998. URL: http://dl.acm.org/citation.cfm?id=314613.314768.

**32** James T Klosowski, Martin Held, Joseph SB Mitchell, Henry Sowizral, and Karel Zikan. Efficient collision detection using bounding volume hierarchies of k-dops. *IEEE transactions on Visualization and Computer Graphics*, 4(1):21–36, 1998.

**33** Yoshiyuki Kusakari, Hitoshi Suzuki, and Takao Nishizeki. A shortest pair of paths on the plane with obstacles and crossing areas. *International Journal of Computational Geometry & Applications*, 9(02):151–170, 1999.

**34** Brian Lent, Arun Swami, and Jennifer Widom. Clustering association rules. In *Proceedings 13th International Conference on Data Engineering*, pages 220–231. IEEE, 1997.

**35** Ewa Malesinska. *Graph theoretical models for frequency assignment problems*. Citeseer, 1997.

**36** Dániel Marx. Efficient approximation schemes for geometric problems? In *13th Annual European Symposium on Algorithms (ESA)*, volume 3669, pages 448–459. Springer, 2005. doi:10.1007/11561071_41.

**37** Joseph S. B. Mitchell. Approximating maximum independent set for rectangles in the plane. *CoRR*, abs/2101.00326, 2021. Version 1. URL: https://arxiv.org/abs/2101.00326v1.

**38** Joseph SB Mitchell. Approximating maximum independent set for rectangles in the plane. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 339–350. IEEE, 2022.

**39** Frank Nielsen. Fast stabbing of boxes in high dimensions. *Theor. Comput. Sci.*, 246(1-2):53–72, 2000. doi:10.1016/S0304-3975(98)00336-3.

**40** Bram Verweij and Karen Aardal. An optimisation algorithm for maximum independent set with applications in map labelling. In *Algorithms-ESA'99: 7th Annual European Symposium Prague, Czech Republic, July 16–18, 1999 Proceedings 7*, pages 426–437. Springer, 1999.

**41** Andreas Wiese. Independent set of convex polygons: From $n^\epsilon$ to $1+\epsilon$ via shrinking. *Algorithmica*, 80(3):918–934, 2018.