Algorithmica
DOI 10.1007/s00453-017-0347-8

Independent Set of Convex Polygons: From n^{ϵ} to $1 + \epsilon$ via Shrinking

Andreas Wiese[1]

Received: 30 June 2016 / Accepted: 9 July 2017

Abstract In the Independent Set of Convex Polygons problem we are given a set of weighted convex polygons in the plane and we want to compute a maximum weight subset of non-overlapping polygons. This is a very natural and well-studied problem with applications in many different areas. Unfortunately, there is a very large gap between the known upper and lower bounds for this problem. The best polynomial time algorithm we know has an approximation ratio of n^{ϵ} and the best known lower bound shows only strong NP-hardness. In this paper we close this gap, assuming that we are allowed to shrink the polygons a little bit, by a factor $1 - \delta$ for an arbitrarily small constant $\delta > 0$, while the compared optimal solution cannot do this (resource augmentation). In this setting, we improve the approximation ratio of n^{ϵ} to $(1 + \epsilon)$ which matches the above lower bound that still holds if we can shrink the polygons.

Keywords Approximation algorithms · Independent Set · Geometric Intersection Graphs · PTAS · Shrinking · Resource augmentation · Convex Polygons

1 Introduction

Maximum Weight Independent Set of Convex Polygons (MWISCP) is a natural but algorithmically very challenging problem. We are given a set of convex polygons $\mathcal{P}$ in the plane and our goal is to select a subset $\mathcal{P}' \subseteq \mathcal{P}$ such that the polygons in $\mathcal{P}'$ are pairwise non-overlapping. Each input polygon $P_i \in \mathcal{P}$ is described by its vertices and additionally by a weight w_i. The objective is to maximize the sum of the weights of the selected polygons (note that the weight is independent of the shape of the polygon).

✉ Andreas Wiese
awiese@mpi-inf.mpg.de

[1] Max Planck Institute for Computer Science, Saarbrücken, Germany

Published online: 19 July 2017

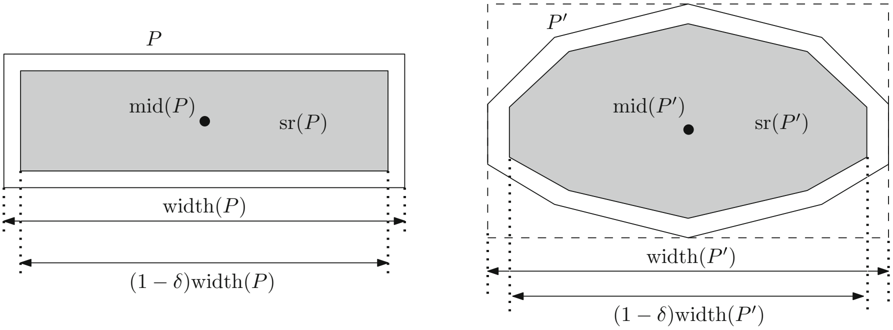

Fig. 1 The *black lines* denote the boundaries of the input polygons P and P'. The *gray areas* denote their shrunk counterparts $\mathrm{sr}(P)$ and $\mathrm{sr}(P')$. The values $\mathrm{width}(P)$ and $\mathrm{width}(P')$ denote the widths of the bounding boxes of P and P', respectively

The problem and its special cases arise in many settings such as map labeling [5,12, 23], cellular networks [11], unsplittable flow [6,8], chip manufacturing [18], or data mining [16,20,21].

On the one hand, the best known polynomial time approximation algorithm has an approximation ratio of n^{ϵ} [15]. On the other hand, the best complexity result shows only strong NP-hardness [14,19] which leaves an enormous gap. Even more, there is a QPTAS [3,17] which suggests that much better polynomial time approximation results are possible.

When dealing with a very difficult problem it is useful to first study simplified settings or relaxations of the original question in order to gain understanding. In this paper, we consider a relaxation of MWISCP in which we are allowed to shrink the input polygons slightly while the compared optimal solution cannot do this. We assume that there is a small constant $\delta > 0$ such that we can shrink each polygon by a factor $1-\delta$ and the new polygon lies in the center of the original one (see Fig. 1): when we consider the rectangular bounding boxes of the original and the shrunk polygon then the height and the width of the box of the shrunk polygon are by exactly a factor $1-\delta$ smaller than the respective values of the box of the original polygon. The reader may think of editing a polygon in a vector graphics program like Adobe InDesign or Inkscape and shrinking it by dragging two opposite corners of its bounding box slightly towards the center point. This yields the Maximum Weight Independent Set of δ-Shrinkable Convex Polygons problem (δ-MWISCP).

We believe that allowing to shrink the input polygons does not change the nature of the problem very much and, thus, insights for δ-MWISCP can be useful for the general case as well. Also, in many applications it is justified to shrink the input objects slightly without losing much benefit, e.g., in map labelling.

1.1 Our Contribution

We present a polynomial time $(1+\epsilon)$-approximation algorithm for δ-MWISCP. This generalizes a previous result for the special case of axis-parallel rectangles [1] to the much larger class of arbitrary convex polygons. Thus, we show that if we are allowed to shrink the input polygons by a little bit then we can improve the best known approximation ratio from n^ϵ to $1 + \epsilon$. This is the best possible approximation ratio since δ-MWISCP is NP-hard, even for unit squares [1]. Note that ϵ and δ are two independent parameters where δ controls only how much the polygons are allowed to be shrunk and ϵ controls only the approximation ratio.

Core of our reasoning is that there exists a $(1-\epsilon)$-approximative shrunk solution for which there is a special cutting sequence. This sequence recursively cuts the input plane into smaller and smaller pieces until each piece either coincides with a polygon from the solution (i.e., the polygon is "cut out") or it has empty intersection with all polygons from this solution. Importantly, each piece arising in this sequence and each recursive cut has only constant complexity, i.e., a constant number of vertices and edges. This allows us to design a dynamic program that recursively guesses the above cut sequence and then outputs the corresponding $(1 - \epsilon)$-approximative shrunk solution.

A key difficulty when approximating independent set in the geometric setting is that the input objects can have very different angles. Note that for Independent Set of (axis-parallel) Rectangles there is a polynomial time $O(\log n/\log\log n)$-approximation algorithm [10] but for straight line segments (with possibly very different angles) we know only a n^ϵ-approximation [15]. Also in our argumentation we need to control the angles of the polygons, or more precisely the angles of the polygon's edges with an underlying grid that guides the construction of our cutting sequence. We need that these angles are bounded away from $\pi/2$. To achieve this we give our grid a random rotation. We are not aware of any prior work in which a randomly rotated grid was used and in our setting case it turns out to be exactly the right tool to address one of our key difficulties.

1.2 Other Related Work

Many cases of geometric independent set have been studied, being distiguished by the types of the objects arising. For axis-parallel squares of arbitrary sizes there is a PTAS due to Erlebach et al. [13]. For axis-parallel rectangles, prior to the mentioned $O(\log n/\log\log n)$-approximation algorithm [10], several $O(\log n)$-approximation algorithms were known [5,7,20,22]. In the unweighted case there is even a $O(\log\log n)$-approximation by Chalermsook and Chuzhoy [9]. For curves in the plane Fox and Pach give a n^ϵ-approximation, assuming that any two curves intersect only $O(1)$ times [15]. This improves and generalizes an earlier $n^{1/2+o(1)}$-approximation due to Agarwal and Mustafa for straight line segments [4].

Going beyond polynomial time results, for independent set of arbitrary polygons there is a QPTAS [3,17], i.e., a $(1 + \epsilon)$-approximation in time $n^{(\log n)^{O_\epsilon(1)}}$, building on an earlier QPTAS for axis-parallel rectangles [2]. This implies that all the above problems are not APX-hard, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{\mathrm{poly}(\log n)})$.

2 Shrinking Model and Preliminaries

We assume that there is a value $N \in \mathbb{N}$ such that each of the n given input polygons $P_i \in \mathcal{P}$ is specified by vertices $v_{i,1}, v_{i,2}, \ldots \in \{0, \ldots, N\}^2$ and a weight $w_i \in \mathbb{N}$.

Note that for arbitrary rational input data we can always translate and scale the input polygons such that all of their vertices have non-negative integral coordinates. For any set of polygons $\mathcal{P}' \subseteq \mathcal{P}$ we define $w(\mathcal{P}') := \sum_{P_i \in \mathcal{P}'} w_i$. For each polygon $P \in \mathcal{P}$ we define its *midpoint* $\mathrm{mid}(P)$ to be the centroid of its (rectangular) bounding box, see Fig. 1. For any two points p, p' we define by $\ell(p, p')$ the line segment connecting p and p' and we define $\mathrm{dist}(p, p') := \left\|\ell(p, p')\right\|_2$. In our shrinking model for each polygon $P_i \in \mathcal{P}$ we define a new polygon $\mathrm{sr}(P_i)$ defined by vertices $v'_{i,1}, v'_{i,2}, \ldots$ such that $v'_{i,k} \in \ell(v_{i,k}, \mathrm{mid}(P))$ for each k and such that $\mathrm{dist}(\mathrm{mid}(P), v'{}_{i,k}) = (1 - \delta)\,\mathrm{dist}(\mathrm{mid}(P), v_{i,k})$. Observe that if P is convex then $\mathrm{sr}(P) \subseteq P$ and also $\mathrm{sr}(P)$ is convex.

In δ-MWISCP our task is to compute a set of polygons $\mathcal{P}' \subseteq \mathcal{P}$ such that for any two polygons $P, P' \in \mathcal{P}'$ we have that $\mathrm{sr}(P) \cap \mathrm{sr}(P') = \emptyset$. We compare the value of our (almost feasible) solution to the value of an optimal feasible solution $\mathrm{OPT}(\mathcal{P}) \subseteq \mathcal{P}$ which can *not* shrink the polygons, i.e., $\mathrm{OPT}(\mathcal{P})$ is a subset of $\mathcal{P}$ of maximum total weight $w(\mathrm{OPT}(\mathcal{P}))$ with the property that $P \cap P' = \emptyset$ for any two polygons $P, P' \in \mathrm{OPT}(\mathcal{P})$. Thus, an α-approximation algorithm for δ-MWISCP computes a solution $\mathcal{P}' \subseteq \mathcal{P}$ such that $w(\mathcal{P}') \geq \alpha^{-1} \cdot w(\mathrm{OPT}(\mathcal{P}))$.

Note that for a non-convex polygon P we cannot guarantee that $\mathrm{sr}(P) \subseteq P$. Thus, for arbitrary polygons we no longer obtain a relaxation to the original problem. In particular, the optimal solution for the shrunk polygons might be worse than the optimal solution for the original polygons. Therefore, in this paper we allow only convex polygons.

For technical reasons we assume w.l.o.g. that the width of the bounding box of each input polygon is larger than its height. This can be ensured by stretching the input plane horizontally. Note that also in our shrinking model this yields an equivalent instance and that this increases each coordinate by at most a factor of $O(N)$.

3 Preprocessing and Shrinking

In this section we describe preprocessing steps in which we remove some of the input polygons and shrink the remaining ones. While doing this, we lose at most a factor $1 + \epsilon$ in our approximation ratio. Also, we ensure that the shrunk polygons are "well-behaved" so that our main algorithm (described in the next section) has an easier task. First, we ensure that each polygon has only few, i.e., constantly many vertices and edges.

Lemma 1 *There exists a constant $K = O_\delta(1)$ such that by shrinking each polygon by a factor $1 - \delta$ we can assume that it has at most K vertices.*

Proof Let $P \in \mathcal{P}$ and assume w.l.o.g. that $\mathrm{mid}(P)$ is the origin, see Fig. 2. For each of the four quadrants, we shrink the intersection of P with the quadrant separately. Consider $P \cap [0, \infty) \times [0, \infty)$. Let p_0 be the point on the boundary of $P \cap [0, \infty) \times [0, \infty)$ with maximum y-coordinate. We assume w.l.o.g. that $p_0 = (0, 1)$. Similary, let p'_0 be the point on the boundary of $P \cap [0, \infty) \times [0, \infty)$ with maximum x-coordinate and we assume w.l.o.g. that $p'_0 = (1, 0)$. We define a set of $O_\delta(1)$ points $p_1, p_2, \ldots$ on the boundary of P in $[0, \infty) \times [0, \infty)$, see Fig. 2 for a sketch. For each $k \in \mathbb{N}$ such

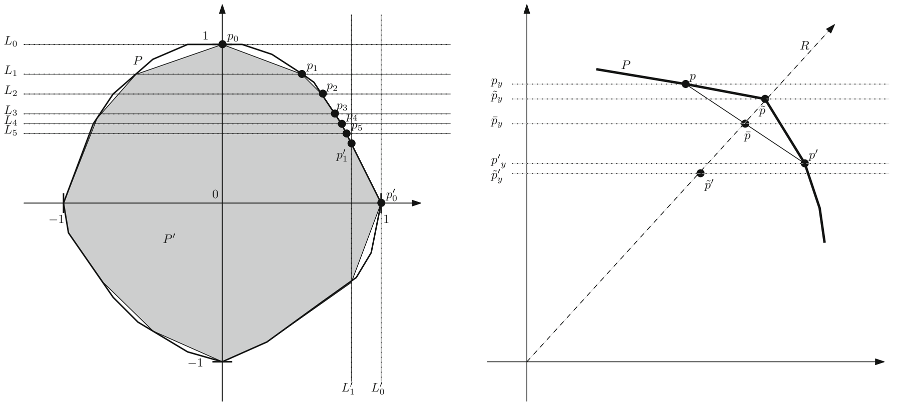

Fig. 2 *Left* shrinking the polygons so that they have only $K = O_\delta(1)$ vertices each. The *thick lines* show the boundary of P, the *gray polygon* is the smaller polygon P' with only K vertices. *Right* the placement of the points $p, p', \bar{p}, \tilde{p}, \tilde{p}'$ in the proof of Lemma 1. The *thick line* shows the boundary of P.

that $(1-\delta)^k \geq (1-\delta)\cdot\frac{1}{2}$ we define the horizontal line $L_k := \mathbb{R} \times \{(1-\delta)^k\}$. Note that there are $O_\delta(1)$ such lines. For each such line L_k we define the point p_k to be the point on the boundary of P in $[0,\infty)\times[0,\infty)$ that intersects L_k.

Similarly, we define a set of $O_\delta(1)$ points $p'_1, p'_2, \ldots$ on the boundary of P in $[0,\infty)\times[0,\infty)$. Let $x_{\max}$ denote the maximum x-coordinate of a point $p_0, p_1, \ldots$. Note that by convexity of P we have that $x_{\max} \geq 1/2$. For each $k \in \mathbb{N}$ such that $(1-\delta)^k \geq x_{\max}$ we define a vertical line $L'_k := \{(1-\delta)^k\} \times \mathbb{R}$ and for each such line L'_k we define the point p'_k to be the point on the boundary of P in $[0,\infty)\times[0,\infty)$ that intersects L'_k. This way, we define a collection of points $p_0, p'_0, p_1, p'_1, \ldots$. Note that by construction their clock-wise order is $p_0, p_1, \ldots, p_f, p'_{f'}, p'_{f'-1}, \ldots, p'_0$ for two integers f, f'. Doing this construction for each of the four quadrants, we define a new polygon P' as the convex hull of all these $O_\delta(1)$ points.

We claim that $\mathrm{sr}(P) \subseteq P' \subseteq P$. It is clear that $P' \subseteq P$. We prove now that $\mathrm{sr}(P) \subseteq P'$. It suffices to show that for any two consecutive vertices p, p' of P' inside one quadrant we have that $\ell(p,p') \subseteq P\backslash\mathrm{sr}(P)$. Consider two such vertices $p = (p_x, p_y)$ and $p' = (p'_x, p'_y)$ of P' in $P' \cap [0,\infty)\times[0,\infty)$ and assume w.l.o.g. that they are in clock-wise order. Note that by construction we have that $p'_y \geq (1-\delta)p_y$ or $p_x \geq (1-\delta)p'_x$. Assume w.l.o.g. that $p'_y \geq (1-\delta)p_y$.

In order to show that $\ell(p,p') \subseteq P\backslash\mathrm{sr}(P)$ it suffices to show that $\ell(p,p') \cap \mathrm{sr}(P) = \emptyset$ since $p, p' \in P$ and P is convex. Let $\bar{p} \in \ell(p,p')$ be a point. Let R be the ray that starts in $\mathrm{mid}(P)$ and goes through $\bar{p} = (\bar{p}_x, \bar{p}_y)$. Let $\tilde{p} = (\tilde{p}_x, \tilde{p}_y)$ be the (unique) point on the boundary of P that lies on R. Similarly, let $\tilde{p}' = (\tilde{p}'_x, \tilde{p}'_y)$ be the (unique) point on the boundary of $\mathrm{sr}(P)$ that lies on R. It suffices to show that $\bar{p}_y \geq \tilde{p}'_y$. We have that $p_y \geq \tilde{p}_y \geq \bar{p}_y \geq p'_y \geq (1-\delta)p_y$. Also, it holds that $\tilde{p}'_y = (1-\delta)\tilde{p}_y$. Therefore we have that $\bar{p}_y \geq (1-\delta)\tilde{p}_y = \tilde{p}'_y$. □

We group the polygons by their diameters. For each polygon P denote by $\mathrm{diam}(P)$ its diameter. We do our grouping to achieve two goals: we want that within each group the diameters of the polygons differ by at most a factor $O_{\delta,\epsilon}(1)$ and between two groups the diameters differ by at least a factor $\frac{1}{K^2\delta}$.

Lemma 2 *By losing a factor of* $1+\epsilon$ *in the value of the optimal solution, we can assume that there is a partition of the polygons* $\mathcal{P}$ *into* $O_{\delta,\epsilon}(\log N)$ *groups* $\mathcal{P}_i$ *and values* $\mu_i', \mu_i \in \mathbb{N}$ *for each group* $\mathcal{P}_i$ *such that*

- $\mu_i' \leq \mathrm{diam}(P) < \mu_i$ *for each* $P \in \mathcal{P}_i$ *and*
- $\delta\epsilon^2/K^2 \cdot \mu_i' = \mu_{i+1}$ *and* $\mu_i/\mu_i' = (1/(\delta K^2))^{1/\epsilon}$ *for each* i.

Proof We first group the polygons in $\mathcal{P}$ into groups $\bar{\mathcal{P}}_1, \ldots, \bar{\mathcal{P}}_m$ for $m = O(\log N)$ based on their diameters, where for each $j \in \{1, \ldots, m\}$ we have that

$$\bar{\mathcal{P}}_j = \left\{ P_i : \mathrm{diam}(P_i) \in \left[\left(\frac{K^2}{\delta\epsilon^2}\right)^{j-1}, \left(\frac{K^2}{\delta\epsilon^2}\right)^{j} \right) \right\}.$$

Then, we group every $1/\epsilon$ consecutive groups $\bar{\mathcal{P}}_j$ together to obtain *supergroups*. We define supergroups with respect to different values of "shifts" as follows. For each shift $s \in \{0, \ldots, 1/\epsilon - 1\}$ and for each $i \geq 1$ we define the supergroup $\mathcal{T}_{s,i} := \bigcup_{j=s+(i-1)/\epsilon+1}^{s+i/\epsilon-1} \bar{\mathcal{P}}_j$. Notice that for each fixed s, if we took the union of the supergroups $\mathcal{T}_{s,i}$, we would get $\mathcal{T}_s = \bigcup_i \mathcal{T}_{s,i} = \bigcup_{j:j \neq s \pmod{1/\epsilon}} \bar{\mathcal{P}}_j$.

We claim that $\sum_{s=1}^{1/\epsilon} \mathrm{OPT}(\mathcal{T}_s) \geq (1-\epsilon)\mathrm{OPT}/\epsilon$. Let $\mathcal{P}^*$ be an optimal solution. We argue that $\sum_{s=1}^{1/\epsilon} w(\mathcal{T}_s \cap \mathcal{P}^*) \geq (1-\epsilon)w(\mathcal{P}^*)/\epsilon$. Notice that each polygon $P_i \in \mathcal{P}^*$ appears in $(1/\epsilon) - 1$ terms on the left-hand-side. More precisely, only if $P_i \in \mathcal{P}_j$ where $j = s \pmod{1/\epsilon}$ the contribution from polygon P_i does not appear. It follows that $\sum_{s=1}^{1/\epsilon} \mathrm{OPT}(\mathcal{T}_s) \geq (1-\epsilon)\mathrm{OPT}/\epsilon$. Thus, there must be a shift $s \in \{0, \ldots, 1/\epsilon - 1\}$ such that $w(\mathcal{T}_s \cap \mathcal{P}^*) \geq (1-\epsilon)w(\mathcal{P}^*)$. For each supergroup $\mathcal{T}_{s,i}$ this yields values μ_i', μ_i such that $\mu_i' \leq \mathrm{diam}(P) < \mu_i$ for each $P \in \mathcal{T}_{s,i}$ and additionally for each i we have that $\delta\epsilon^2/K^2 \cdot \mu_i' = \mu_{i+1}$ and $\mu_i/\mu_i' = (1/(\delta K^2))^{1/\epsilon}$. Finally, we define $\mathcal{P}_i := \mathcal{T}_{s,i}$ for each i. □

3.1 Hierarchical Grids

We define a family of hierarchical vertical grids $G_0, G_1, \ldots, G_m$ with $m = O(\log N)$. For each $i \in \{0, \ldots, m\}$ we define $G_i := \{\{x\} \times \mathbb{R} | \exists k \in \mathbb{N} \,\text{s.t.}\, x = k \cdot g_i\}$ with $g_i := \frac{\delta\epsilon}{4K^2} \cdot \mu_i'$. Observe that the grids are hierarchical, i.e., each grid line of G_i is also grid line of $G_{i'}$ for each $i' > i$. We give these grids a random rotation. We rotate all of them by the same angle α. This angle α is drawn uniformly at random from the range $[\pi/4, \pi/2]$. Let ℓ be a line of the grids. We are interested in the angle between ℓ and the edges of the polygons. We say that ℓ and a line segment ℓ' have a *good angle* if the angle between ℓ and the line containing ℓ' have an angle of at least ϵ/K^2 and at most $\frac{\pi}{2} - \epsilon/K^2$, otherwise we say that they have a *bad* angle.

Lemma 3 *Let* $P \in \mathcal{P}$. *With probability at least* $1 - O(\epsilon)$ *all line segments connecting two vertices of* P *and all line segments connecting* $\mathrm{mid}(P)$ *with a vertex of* P *have a good angle with all grid lines.*

Proof The polygon P has at most K vertices. Thus, there are at most $K(K-1)/2$ line segments connecting two of its vertices. Also, there are at most K line segments connecting $\mathrm{mid}(P)$ with a vertex of P. This gives $O(K^2)$ segments in total. The probability that one of these segments has a bad angle with ℓ is at most $O(\epsilon/K^2)$. By the union bound, with probability at least $1-O(\epsilon)$ each such segment has a good angle with ℓ. □

We delete all polygons P that have two vertices v, v' such that $\ell(v, v')$ or $\ell(v, \mathrm{mid}(P))$ has a bad angle with the grid lines. Next, we give the grids a random shift upwards without changing their angles. Let ℓ denote an arbitrary vertical line. We observe that ℓ intersects all lines in G_0. Denote by L the distance of any two consecutive points on ℓ which intersect a point of G_0. We draw a value $L' \in [0, L)$ uniformly at random and shift all grid lines up by L' units. For simplicity, denote by $G_0, G_1, \ldots, G_m$ the grid lines after this shift.

Lemma 4 *Every $P \in \mathcal{P}_{i+1}$ intersects a grid line of G_i with probability at most 2ϵ.*

Proof Since $P \in \mathcal{P}_{i+1}$ we have that $\mathrm{diam}(P) \le \mu_{i+1}$. The spacing between two grid lines of G_i is $g_i = \frac{\delta\epsilon}{2K^2}\cdot\mu'_i$ and $\mu'_i = \mu_{i+1}\cdot\frac{K^2}{\delta\epsilon^2}$. Thus, $\mathrm{diam}(P) \le \mu'_i\cdot\delta\epsilon^2/K^2 = 2\epsilon\cdot g_i$ which implies that P intersects a grid line of G_i with probability at most 2ϵ. □

For each $i \in \mathbb{N}$ we delete all polygons $P \in \mathcal{P}_{i+1}$ that intersect a grid line of G_i. Due to Lemma 4 this costs at most a factor of $1-2\epsilon$ in the objective. Since the grids are hierarchical, if a polygon $P \in \mathcal{P}_{i+1}$ does not intersect a grid line of G_i then it does not intersect a grid line of $G_{i'}$ for any $i' \ge i$.

3.2 Shrinking

Next, we want to shrink the polygons. Intuitively, for each polygon $P \in \mathcal{P}_i$ let $v^{\uparrow}(P)$ and $v^{\downarrow}(P)$ denote the top-most and bottom-most vertices “relative to the grid lines”. Formally, we define $v^{\uparrow}(P)$ and $v^{\downarrow}(P)$ to be the two vertices of P for which there exists a line ℓ parallel to the rotated grid lines that intersects $v^{\uparrow}(P)$ (intersects $v^{\downarrow}(P)$) and no point in the interior of P, see Fig. 3.

We shrink P to a polygon P' such that $v^{\uparrow}(P')$ and $v^{\downarrow}(P')$ lie on grid lines of G_i. The next lemma shows that this is indeed possible. Heart of this reasoning is that there are at least $1/\delta$ grid lines of G_i between $v^{\uparrow}(P)$ and $v^{\downarrow}(P)$. We do this operation with all input polygons.

Lemma 5 *Let $P \in \mathcal{P}_i$. In polynomial time we can compute a polygon P' with at most $K+2$ edges such that $\mathrm{sr}(P) \subseteq P' \subseteq P$ and $v^{\uparrow}(P')$ and $v^{\downarrow}(P')$ lie on grid lines of G_i. Furthermore, all edges of P' crossing a grid line of G_i in a non-zero angle have a good angle with this grid line. Furthermore,* $\mathrm{diam}(P') \le \mathrm{diam}(P)$.

Proof For ease of notation, we define a distance measure given by the grid lines. For any two points p, p' we define the quantity $\mathrm{dist}_{\mathrm{grid}}(p, p')$ as the distance between the two lines ℓ_1, ℓ_2 that are parallel to the grid lines and that go through p and p', respectively, see Fig. 3.

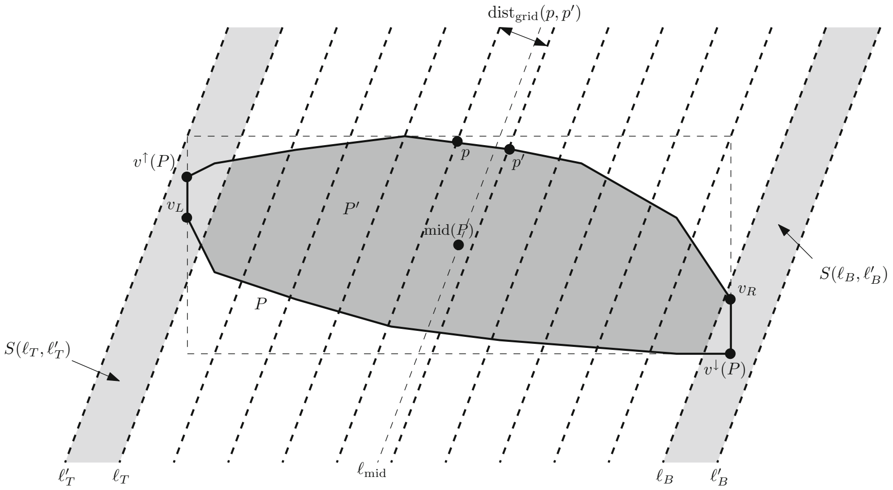

Fig. 3 Shrinking the polygon so that the points $v^{\uparrow}(P)$, $v^{\downarrow}(P)$ lie on grid lines (proof of Lemma 5). The *bold lines* denote the boundary of the polygon P. The *light gray* areas denote the regions $S(\ell_T, \ell'_T)$ and $S(\ell_B, \ell'_B)$. The *dark gray* area show the new polygon P'. The *bold dashed lines* are the lines of G_i

Let ℓ_{mid} denote a line parallel to the grid lines that goes through $\mathrm{mid}(P)$. Recall that we assumed that the width of each bounding box is larger than its height. Due to our range for the angles of the grid, ℓ_{mid} intersects the top and bottom edge of the bounding box of P but not its left or right edge. Consider a vertex v_L of P that lies on the left edge of the bounding box of P. We know that $\ell(\mathrm{mid}(P), v_L)$ has a good angle with the grid. Also, $\mathrm{dist}(\mathrm{mid}(P), v_L) \geq \frac{1}{4}\mathrm{diam}(P)$. Therefore, $\mathrm{dist}_{\mathrm{grid}}(\mathrm{mid}(P), v_L) \geq \frac{\epsilon}{4K^2}\mathrm{diam}(P) \geq \frac{\epsilon}{K^2} \cdot \frac{1}{4}\mu'_i$. The line ℓ_{mid} splits the plane into two sides. Assume w.l.o.g. that $v^{\uparrow}(P)$ lies on the same side as v_L. This implies that also $\mathrm{dist}_{\mathrm{grid}}(\mathrm{mid}(P), v^{\uparrow}(P)) \geq \frac{\epsilon}{K^2} \cdot \frac{1}{4}\mu'_i$.

Similarly, consider a vertex v_R of P that lies on the right edge of the bounding box of P. We have that $v^{\downarrow}(P)$ lies on the same side of ℓ_{mid} as v_R. Also, we can show like above that $\mathrm{dist}_{\mathrm{grid}}(\mathrm{mid}(P), v^{\downarrow}(P)) \geq \frac{\epsilon}{K^2} \cdot \frac{1}{4}\mu'_i$.

Now let ℓ_T be the top-most line of G_i that intersects P. Let ℓ'_T be the next higher grid line of G_i above ℓ_T and let $S(\ell_T, \ell'_T)$ denote the stripe enclosed by them. We remove all points in $P \cap S(\ell_T, \ell'_T)$ from P. We do the same operation with the bottom-most line ℓ_B of G_i intersecting P. Denote by P' the resulting polygon. Since there are $1/\delta$ grid lines of G_i crossing $\ell(\mathrm{mid}(P), v^{\uparrow}(P))$ and $\ell(\mathrm{mid}(P), v^{\downarrow}(P))$ we have that $\mathrm{sr}(P) \subseteq P'$. The other claimed properties of P' follow by construction. □

3.3 Horizontal Grids

From now on we do not shrink the polygons any further. Let us assume w.l.o.g. that the grid lines $G_0, G_1, \ldots, G_m$ are exactly vertical and that there is an integer N' such that the input polygons are contained in the area $[0, N'] \times [0, N']$ for some integer $N' = O(N)$. We add a hierarchy of horizontal grids $\bar{G}_0, \bar{G}_1, \ldots, \bar{G}_m$ to the vertical grids $G_0, G_1, \dots, G_m$. For each $i \in \{0, \dots, m\}$ we define $\bar{G}_i := \{\mathbb{R} \times \{y\} | \exists k \in \mathbb{N} \text{ s.t. } y = k \cdot g_i\}$ with (as defined before) $g_i = \frac{\delta\epsilon}{2K^2} \cdot \mu'_i$. Thus, for each i the grid $\bar{G}_i$ has exactly the same spacing as G_i. We give the horizontal grids $\bar{G}_i$ a random shift up. Formally, we draw a value $L' \in [0, g_0)$ uniformly at random and shift each grid line $\bar{G}_i$ up by L' units. Then, for each $i \in \{0, \dots, m\}$ we delete all remaining polygons from $\cup_{j>i+1}\mathcal{P}_i$ that intersect a grid line in $\bar{G}_i$. The following lemma can be proven similarly as Lemma 4. Like in that lemma, the key argument is that for each $P \in \mathcal{P}_{i+1}$ it holds that $\text{diam}(P) \le 2\epsilon \cdot g_i$ and observe here that due to the shrinking procedure the diameter of each polygon did not increase (see the last property of Lemma 5).

Lemma 6 *Each $P \in \mathcal{P}_{i+1}$ intersects a grid line of $\bar{G}_i$ with probability at most* 2ϵ.

Denote by $\mathcal{P}'$ the resulting set of shrunk polygons. For each integer i we define $\mathcal{P}'_i$ to be the sets of polygons we obtain when shrinking each polygon in $\mathcal{P}_i$. Note that we lost only a factor of $(1 + O(\epsilon))$ in our approximation ratio (see Lemmas 2, 4 and 6).

4 Dynamic Program

Our algorithm is a geometric divide-and-conquer algorithm similar to the algorithm used in [1,3]. It recursively divides the area containing the input polygons into smaller and smaller pieces. When it makes a recursive call for a piece $A \subseteq [0, N']^2$ then it computes a (near-optimal) solution to the subproblem given by all input polygons that are contained in A. To do this, it tries all possibilities to partition A into at most $k = O_{\delta,\epsilon}(1)$ subpieces such that the boundary of each of them consists of at most k line segments out of a suitable set $\mathcal{L}$ defined below. Then, it makes a recursive call on each of these subpieces and obtains a (near-optimal) solution for each of those. By putting them together, it obtains a candidate solution for the original piece A. Additionally, it checks what profit it can obtain by selecting only one polygon that is contained in A. Eventually, it returns the best solution out of all candidate solutions stemming from all partitions of A and all single polygons contained in A. We will show that if the parameter k is sufficiently large then our algorithm will output a set that is at least as profitable as the optimal solution for $\mathcal{P}'$.

We embed the whole procedure into a dynamic program (DP). Let $k = O_{\delta,\epsilon}(1)$ be a parameter to be defined later. Our DP table has one cell for each (not necessarily convex) area $A \subseteq [0, N']^2$ whose boundary consists of at most k line segments such that

- each line segment is a subset of an edge of a polygon in $\mathcal{P}'$ or a subset of a grid line in $\mathcal{G} := \cup_{i=0}^{m} G_i \cup \bar{G}_i$ and
- the endpoint of each line segment is
  - the vertex of a polygon in $\mathcal{P}'$, or
  - the intersection of an edge of a polygon in $\mathcal{P}'$ with a grid line in $\mathcal{G}$, or
  - the intersection of two grid lines in $\mathcal{G}$.

We call such an area A a *piece*. Denote by $\mathcal{L}$ the set of all line segments that arise on the boundaries of the pieces defined above, see Fig. 4 for an example. Denote by GEO-DP this dynamic program. As the following lemma shows, it has pseudo-polynomial running time. We will explain later how to improve this to polynomial time.

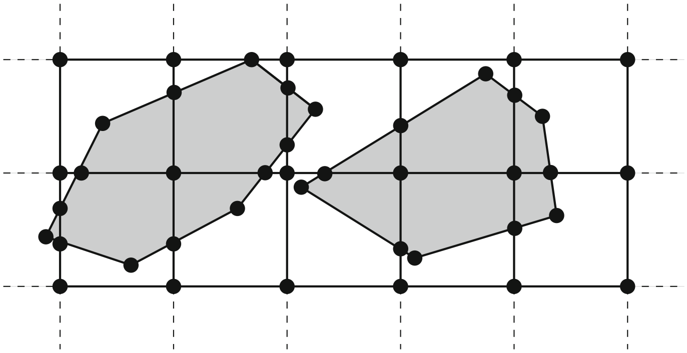

Fig. 4 An instance of δ-MWISCP. The *black circles* denote all possible endpoints of the lines in $\mathcal{L}$: vertices of the polygons, intersections of polygon edges with grid lines, and intersection of two grid lines. Each line in $\mathcal{L}$ is a subset of an input polygon or a subset of a grid line. Hence, each line in $\mathcal{L}$ is contained in one of the *bold lines* in the Figure

Lemma 7 *The number of DP-cells is bounded by* $(n+N)^{O_{\delta,\epsilon}(k)}$. *If* $k = O_{\epsilon,\delta}(1)$ *then the overall running time of GEO-DP is also bounded by* $(n+N)^{O_{\delta,\epsilon}(1)}$.

Proof First, we upper bound the total number of possible endpoints of line segments in $\mathcal{L}$. Each polygon in $\mathcal{P}'$ has at most $K+2$ vertices and edges. The total number of grid lines is bounded by $O_{\delta,\epsilon}(N)$. Thus, the total number of endpoints of line segments in $\mathcal{L}$ is bounded by $O_{\delta,\epsilon}(n \cdot K \cdot N + N^2)$ and therefore $|\mathcal{L}| \leq O_{\delta,\epsilon}(n^2 \cdot K^2 \cdot N^2 + N^4)$. Hence, the number of DP-cells is at most $(n \cdot K \cdot N)^{O_{\delta,\epsilon}(k)} = (n+N)^{O_{\delta,\epsilon}(k)}$ since $K = O_\delta(1)$. When we compute an entry of a DP-cell for a piece A we need to enumerate all partitions of A into at most k smaller pieces. This can be done in time $(n+N)^{O_{\delta,\epsilon}(k^2)}$ which is bounded by $(n+N)^{O_{\delta,\epsilon}(1)}$ if $k = O_{\epsilon,\delta}(1)$. □

There is a piece containing all input polygons. Thus, the subproblem corresponding to this piece is identical to the problem we want to solve. Our final output is the solution computed for this piece. We want to show that its solution is at least as profitable as the optimal solution for $\mathcal{P}'$. In order to show this we assume w.l.o.g. that the input to GEO-DP consists only of this optimal solution for $\mathcal{P}'$. We denote it $\mathcal{P}''$. We define $\mathcal{P}''_i := \mathcal{P}'_i \cap \mathcal{P}''$ for each i.

In the remainder of this section, we prove that GEO-DP computes a solution whose profit is at least $w(\mathcal{P}'') \geq (1 - O(\epsilon)) \cdot \mathrm{OPT}(\mathcal{P})$. To this end, we describe a recursive sequence of cuts that subdivides the input area $[0, N'] \times [0, N']$ into smaller and smaller pieces such that (i) for each arising piece there is a DP-cell, (ii) no polygon in $\mathcal{P}''$ is intersected by any of these cuts, and (iii) each of the pieces obtained at the end has non-empty intersection with at most one polygon from $\mathcal{P}''$. In Sect. 4.1 we formalize such a sequence and prove that if it exists then GEO-DP computes a solution of weight at least $w(\mathcal{P}'')$. In Sect. 4.2 we prove that it indeed exists for a suitable choice of $k = O_{\delta,\epsilon}(1)$. Finally, in Sect. 4.3 we explain how this yields our main algorithm.

4.1 Cutting Sequence

We describe a sequence of cuts that recursively subdivides the whole area $[0, N'] \times [0, N']$ and "cuts out" all polygons in $\mathcal{P}''$. It ensures that for each piece arising in the sequence there exists a DP-cell according to the above definition and that each cut for such a piece partitions it into at most k smaller pieces.

Formally, we describe this sequence of cuts by a tree T where each node v is associated with a piece A_v in the plane. We say that a tree T is a $(k, \mathcal{P}'')$*-region decomposition* if the following holds:

- For each node v in T and each polygon $P \in \mathcal{P}''$ we have that if P does not coincide with A_v, i.e., $P \neq A_v$, then either P is contained in A_v or P is disjoint from A_v.
- For tree nodes u and v such that v is a parent of u we have $A_u \subseteq A_v$. Each node $v \in T$ has at most $k' \leq k$ children $u_1, \ldots, u_{k'}$ in T and $\bigcup_{i=1}^{k'} A_{u_i} = A_v$.
- For each leaf node v of T the piece A_v contains at most one polygon in $\mathcal{P}''$ and it has empty intersection with all other polygons in $\mathcal{P}''$.
- For each node v in T the area A_v is connected and its boundary can be described by at most k line segments from $\mathcal{L}$.

Note that a similar type of decomposition was also used in [1].

Lemma 8 *If there exists a* $(k, \mathcal{P}'')$*-region decomposition then GEO-DP outputs a solution of weight at least* $w(\mathcal{P}'')$ *when it is parametrized by* k*.*

Proof We assume that for the set of (pairwise non-overlapping) polygons $\mathcal{P}''$ there exists a $(k, \mathcal{P}'')$-region decomposition. Let T be the tree that represents the region decomposition for $\mathcal{P}''$. We now prove the following statement by induction on the structure of T from its leaves to the root:

Consider a node $u \in T$*. When GEO-DP processes the instance given by the polygons* $\mathcal{P}'$ *that are contained in* A_u *it outputs a set of polygons* $\mathcal{P}''_u$ *whose weight* $w(\mathcal{P}''_u)$ *is at least the total weight of the polygons in* $\mathcal{P}''$ *that are contained in* P_u*.*

In particular, for the root node r with $A_r = [0, N'] \times [0, N']$ this implies that GEO-DP computes a set of polygons $\mathcal{P}''_r$ with weight $w(\mathcal{P}''_r) = w(\mathcal{P}'')$ as desired. The base case is obvious: for each leaf node v its piece A_v coincides with a polygon $P_i \in \mathcal{P}''$ and thus P_i is contained in A_v. Hence, GEO-DP returns a solution whose weight is at least $w(P_i)$. For the inductive step consider a node v for which the induction hypothesis holds for all children of v. Let $\mathcal{P}''_v$ denote all polygons from $\mathcal{P}''$ that are contained in P_v. Denote the children of v by $v_1, \ldots, v_{k'}$ for some $k' \leq k$. We have that $A_v = \bigcup_{j=1}^{k'} A_{v_j}$ and that the pieces $A_{v_1}, \ldots, A_{v_{k'}}$ are pairwise disjoint. For each $j \in \{1, \ldots, k'\}$ let $\mathcal{P}''_{v_j}$ denote the polygons from $\mathcal{P}''$ that are contained in A_{v_j}. The sets $\mathcal{P}''_{v_j}$ form a partition since each polygon in $\mathcal{P}''_v$ is contained in some piece A_{v_j}. In particular, this implies that $w(\mathcal{P}''_v) = \sum_{j=1}^{k'} w(\mathcal{P}''_{v_j})$. Moreover, GEO-DP considers the cut which partitions A_v into $A_{v_1}, \ldots, A_{v_{k'}}$ and returns, by the induction hypothesis, a solution $\bar{\mathcal{P}}_v$ consisting of one solution $\bar{\mathcal{P}}_{v_j}$ for each piece A_{v_j} such that $w(\bar{\mathcal{P}}_v) = \sum_{j=1}^{k'} w(\bar{\mathcal{P}}_{v_j}) \geq \sum_{j=1}^{k'} w(\mathcal{P}''_{v_j}) = w(\mathcal{P}''_v)$. This completes the proof. □

The proof of Lemma 8 is identical to the proof of a similar lemma in [1].

4.2 Existence of Region Decomposition

In this section we prove that a $(k, \mathcal{P}'')$-region decomposition always exists. Formally, we prove the following lemma.

Lemma 9 *There exists a universal constant $k = O_{\delta,\epsilon}(1)$ such that a $(k, \mathcal{P}'')$-region decomposition exists.*

To construct the $(k, \mathcal{P}'')$-region decomposition we need to construct its tree T and a piece A_v for each vertex v of T. For the root node v_r we define its region A_{v_r} to be the cell of an artificial grid $G_{-1} \cup \bar{G}_{-1}$ that has exactly one cell that contains $[0, N'] \times [0, N']$. Inductively, we describe the procedure to construct T. Assume that we are given a leaf $v \in T$ whose piece A_v is described as follows: there is a cell C of the grid $G_{i-1} \cup \bar{G}_{i-1}$ for some $i \in \mathbb{N}_0$ and up to two polygons $P_1, P_2 \in \cup_{j=0}^{i-1} \mathcal{P}''_j$ such that each of them intersects both the left and the right grid line of C. Also, we assume that A_v does not intersect any polygon in $\cup_{j=0}^{i-1} \mathcal{P}''_j$. We denote these conditions as the *invariant*. Observe that v_r fulfills it. If A_v contains at most one polygon from $\mathcal{P}''$ and has non-empty intersection with all other polygons in $\mathcal{P}''$ then we do not do anything further with v. Otherwise, we describe how to divide A_v step by step into smaller and smaller pieces, until each arising piece $A_{v'}$ satisfies the invariant for some cell C' of $G_i \cup \bar{G}_i$ and up to two polygons P'_1, P'_2. To describe this partitioning of A_v we create a subtree rooted at v, i.e., for each vertex v' of the subtree there is a region $A_{v'}$ which corresponds to one piece of this partitioning. Observe that A_v is the connected component of $C \backslash \{P_1, P_2\}$ that is adjacent to P_1 and P_2. Note that there is a DP-cell for A_v. Assume w.l.o.g. that P_1 crosses the left and right grid lines of C below P_2. One or both polygons P_1 and P_2 might be undefined and in this case the bottom and/or top boundaries of C take the role of P_1 and/or P_2.

We define a cut through A_v, see Fig. 5 for a sketch. Let ℓ_1 be a grid line of G_i such that ℓ_1 intersects the interior of A_v. Let p be the bottom-most point of $\ell_1 \cap A_v$. Note that at this point ℓ_1 intersects the boundary of P_1. Our cut starts in p and moves up along ℓ_1. If we do not hit any polygon of $\mathcal{P}''$ contained in A_v on the way up then we are done with our cut. Otherwise, suppose that we hit an edge e of a polygon P. Let p' denote the point on e that is hit by ℓ_1. Due to Lemma 3 we know that e has a good angle with ℓ_1. Also, $P \in \mathcal{P}''_i$ since otherwise it would have been deleted before as it intersects a grid line of G_i. Our cut moves along e in the direction that goes up. We continue along the boundary of P in the same direction until we arrive at the leftmost or rightmost point of P. Let p'' denote this point. Due to our shrinking, p'' lies on a grid line ℓ' of G_i. Since all edges of P have a good angle with the grid, we can prove the following lemma.

Lemma 10 *When moving from p' to p'' along the edge of P, we move up by at least $\sin(\epsilon^2/K) \cdot g_i = \Omega_{\delta,\epsilon}(1) \cdot g_i$.*

Proof Since all edges of P have a good angle with the grid lines, this angle is at least ϵ^2/K. Therefore, when we move from p' to p'' we move up by at least $\mathrm{dist}(p', p'') \cdot \sin(\epsilon^2/K) \geq \sin(\epsilon^2/K) \cdot g_i$ units. □

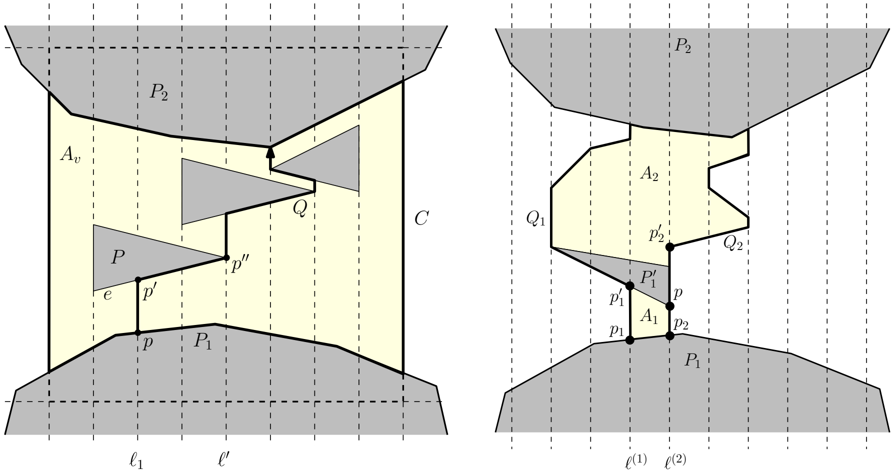

Fig. 5 *Left* the first cut Q that separates the piece A_v (given by the area in the cell C between the polygons P_1 and P_2) into two smaller pieces. The point p is defined as the bottom-most point of $\ell_1 \cap A_v$ and ℓ_1 intersects the boundary of P_1 on p. *Right* a piece (consisting of $A_1 \cup P \cup A_2$) for which there is no grid line that intersects P_1 on the boundary of the piece (first case in the proof of Lemma 11)

Note that the contructed path from p via p' to p'' consists of at most $K+1 \leq O_{\delta,\epsilon}(1)$ line segments. We continue iteratively where now p'' takes the role of p. We stop when we hit the upper boundary of A_v (defined by P_2 or the top boundary of C). Denote by Q the constructed path. The height of C is bounded by $g_{i-1} = O_{\delta,\epsilon}(g_i)$. In every iteration we move up by at least $\Omega_{\delta,\epsilon}(1) \cdot g_i$ units. Thus, there are at most $O_{\delta,\epsilon}(1)$ iterations and Q can be described with $O_{\delta,\epsilon}(1) \cdot (K+1)$ line segments. Observe that we cut only along edges of polygons and along grid lines of G_i. Thus, we did not intersect any polygon from $\mathcal{P}''$.

Our path Q splits A_v into two smaller pieces. Each of the two sides of $A_v \backslash Q$ defines a piece and for each of them we append a child node u_i to v such that A_{u_i} equals this piece. Importantly, each such piece is described by only C, P_1, P_2 and Q and thus its boundary has only $O_{\delta,\epsilon}(1)$ line segments. We continue with each newly created piece A_{u_i}. Assume that there is a grid line ℓ_2 of G_i such that ℓ_2 intersects the boundary of P_1 at a point $\bar{p}$ that lies on the boundary of A_{u_i}. Then $\bar{p}$ takes the role of p above and we find a path Q' that split A_{u_i} into two pieces. We append these pieces in T as child nodes of u_i. Each such piece is then described by C, P_1, P_2, Q and Q' and thus its boundary has at most $O_{\delta,\epsilon}(1)$ edges.

Consider a child node $\bar{u}_j$ of u_i. Similarly as before, assume that there is a grid line ℓ_3 of G_i such that ℓ_3 intersects the boundary of P_1 at a point that lies on the boundary of $A_{\bar{u}_j}$. We compute a path Q'' though $A_{\bar{u}_j}$ as above. Each connected component of $A_{\bar{u}_j} \backslash Q''$ can be described by C, P_1, P_2 and at most *two* of the paths Q, Q', Q''. Similarly, when we continue further like above in the recursion each resulting piece can be described by C, P_1, P_2 and two paths Q_1, Q_2 through A_v where each of the latter can be described by only $O_{\delta,\epsilon}(1)$ line segments.

We can apply the above reasoning as long as there is a grid line ℓ_k of G_i such that the boundary of P_1 intersects ℓ_k at a point that lies on the boundary of the considered piece. Suppose now that this is not possible, i.e, we have a piece $A_{\tilde{v}} \subseteq A_v$ such that at the boundary of $A_{\tilde{v}}$ there is no grid line of G_i that intersects P_1. As the next lemma shows, this piece can then be partitioned into two smaller pieces A_1, A_2 and one polygon P_1', see Fig. 5.

Lemma 11 *Assume that in the above construction we obtain a piece $A_{\tilde{v}}$ such that on the boundary of $A_{\tilde{v}}$ there is no point where a grid line of G_i intersects P_1. Then either $A_{\tilde{v}}$ is contained in a grid column of G_i or $A_{\tilde{v}}$ can be partitioned into two pieces A_1, A_2 and one polygon P_1' such that*

- *the boundary of A_1 consists of two paths $Q_1^{(1)}, Q_1^{(2)}$ with at most $O_{\delta,\epsilon}(1)$ edges each that both connect P_1 and P_1', and*
- *the boundary of A_2 consists of two paths $Q_2^{(1)}, Q_2^{(2)}$ with at most $O_{\delta,\epsilon}(1)$ edges each that both connect P_1' and P_2.*

Proof Assume that $A_{\tilde{v}}$ is not contained in a grid column of G_i (otherwise there is nothing to show). By construction $A_{\tilde{v}}$ is described by two paths Q_1, Q_2 that both connect P_1 and P_2. There is no point on the boundary of $A_{\tilde{v}}$ in which a grid line of G_i intersects P_1. Thus, for the points p_1 and p_2 on which the paths Q_1 and Q_2 start, there must be two consecutive grid lines $\ell^{(1)}, \ell^{(2)}$ of G_i such that $\ell^{(1)}$ intersects P_1 on p_1 and $\ell^{(2)}$ intersects P_1 on p_2, see Fig. 5. Assume that $\ell^{(1)}$ is on the left of $\ell^{(2)}$. Since $A_{\tilde{v}}$ is not contained in a grid column of G_i one of the paths Q_1, Q_2 is not completely vertical. Assume that both paths are not completely vertical (the other case can be proven with similar arguments). Let p_1', p_2' be the points on which Q_1 and Q_2 deviate from being only vertical. Assume w.l.o.g. that the y-coordinate of p_1' is not larger than the y-coordinate of p_2'.

Assume that on p_1' the path Q_1 turns left. Then on p_1' the path Q_1 hits a polygon $P \in \mathcal{P}_i''$. Thus, the boundary of P must intersect $\ell^{(2)}$ at a point p. Then the y-coordinate of this point p must be lower than the y-coordinate of p_2' (since Q_1 goes monotonely upwards). Then we set $P_1' := P$ and A_1 consists of the quadrilateral described by p_1, p_1', p, p_2, and $A_2 = A_{\tilde{v}} \setminus \{A_1, P\}$. The paths $Q_1^{(1)}, Q_1^{(2)}, Q_2^{(1)}, Q_2^{(2)}$ consist of the parts of Q_1 and Q_2 surrounding A_1 and A_2, respectively.

Assume now that on p_1' the path Q_1 turns right after hitting a polygon P. Then Q_1 must cross $\ell^{(2)}$ at a point p'. If the y-coordinate of p' is smaller than the y-coordinate of p_2' then we define $P_1' := P$ and we define the pieces A_1, A_2 and the paths $Q_1^{(1)}, Q_1^{(2)}, Q_2^{(1)}, Q_2^{(2)}$ similarly as in the previous case. Finally, suppose that the y-coordinate of p' is not smaller than the y-coodinate of p_2'. Then on p_2' the path Q_2 hits a polygon P' and it must turn right (since otherwise $P \cap P' \neq \emptyset$). Then the polygon P' must cross $\ell^{(1)}$ underneath p_1' and we define $P_1' := P'$ and the pieces A_1, A_2 and the paths $Q_1^{(1)}, Q_1^{(2)}, Q_2^{(1)}, Q_2^{(2)}$ accordingly. □

When we continue, for the piece A_1 the polygon P_1' takes the role of P_2, and for the piece A_2 the polygon P_1' takes the role of P_1. We continue until all pieces corresponding to leaves are contained in a column of G_i. Let $A_{v'}$ be such a piece. $A_{v'}$ might not yet fulfill the invariant since it might still span many grid cells of $G_i \cup \bar{G}_i$. We know that each polygon $P \in \mathcal{P}_i''$ intersects at least two grid columns of G_i. Thus, there can be no polygon $P \in \mathcal{P}''_i$ with $P \subseteq A_{v'}$. Thus, the boundary of $A_{v'}$ consists of two grid columns and (parts of) the boundary edges of at most two polygons of $\cup_{j=0}^{i}\mathcal{P}''_j$, defining the upper and lower boundary of $A_{v'}$.

As long as $A_{v'}$ is not contained in a grid cell of $G_i \cup \bar{G}_i$ there must be a grid row r of $\bar{G}_i$ such that r has non-empty intersection with the interior of $A_{v'}$. Note that r does not intersect any of the remaining polygons in $\cup_{j \geq i+1}\mathcal{P}''_j$ since we have removed such polygons before already. We split $A_{v'}$ along r into two pieces $A_{v''}$, $A_{v'''}$ and add the corresponding vertices v'' and v''' to T as children of v'. We continue this process until each piece is contained in a grid cell of G_i. Thus, each of our resulting pieces fulfill the invariant and we can repeat the procedure above. We apply the above procedure to each leaf node of T until for each leaf node v it holds that the piece A_v contains at most one polygon in $\mathcal{P}''$ and it has empty intersection with all other polygons in $\mathcal{P}''$. This completes the proof of Lemma 9.

4.3 Main Algorithm

With the above preparation we are ready to describe our main algorithm.

Theorem 1 *For any constants $\epsilon, \delta > 0$ there is a polynomial time $(1+\epsilon)$-approximation algorithm for the maximum independent set of δ-shrinkable convex polygons problem.*

Proof We parametrize GEO-DP by the constant $k = O_{\delta,\epsilon}(1)$ due to Lemma 9. Then together with Lemma 8 this implies that GEO-DP outputs a solution of weight at least $w(\mathcal{P}'')$. Due to our reasoning in Sect. 3 we have that $w(\mathcal{P}'') \geq (1 - O(\epsilon)) \cdot \mathrm{OPT}(\mathcal{P})$, see Lemmas 2, 4 and 6.

It remains to address the fact that in the above form our algorithm has a running time that might be exponential in the input size (since N might be exponential). We argue similarly as in [1]. First observe that there are only $O(\log N)$ recursion levels, which is polynomial in the length of the input encoding. In each level of the grids, it suffices to introduce only grid cells C for which there exists a polygon $P \in \mathcal{P}'$ with $P \subseteq C$. There can be only n such grid cells for each grid $\mathcal{G}_i \cup \bar{\mathcal{G}}_i$ and thus in total there are only $O(n \cdot \log N)$ such cells. Hence, the total number of needed grid lines is also bounded by $O(n \cdot \log N)$.

The preprocessing steps due to Lemmas 1, 2, and 5 can be implemented in time $(n \log N)^{O(1)}$, using that each polygon has at most n vertices and the largest arising number in the input is N. Within the same time bound we can construct the grids $G_i, \bar{G}_i$. Each cell in the DP-table can be decribed by $k = O_{\delta,\epsilon}(1)$ line segments connecting two points where each of them is a vertex of a polygon in $\mathcal{P}'$, the intersection of an edge of a polygon in $\mathcal{P}'$, or the intersection of two grid lines. For these points there are $nK + (n \log N)^{O(1)}$ possibilities. Hence, there are at most $(Kn + n \log N)^{O_{\delta,\epsilon}(1)}$ DP cells in total. To compute the solution for a DP-cell we try all possibilities to split its corresponding piece into at most $k = O_{\delta,\epsilon}(1)$ smaller pieces. Therefore, we obtain an overall running time of $(n + \log N)^{O_{\delta,\epsilon}(1)}$. □

Acknowledgements Open access funding provided by Max Planck Society. The author would like to thank Parinya Chalermsook for helpful discussions on the topic of the paper and the anonymous referees for many helpful comments.

References

1. Adamaszek, A., Chalermsook, P., Wiese, A.: How to tame rectangles: solving independent set and coloring of rectangles via shrinking. In: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015), volume 40 of Leibniz International Proceedings in Informatics (LIPIcs), Dagstuhl, Germany, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, pp. 43–60 (2015)
2. Adamaszek, A., Wiese, A.: Approximation schemes for maximum weight independent set of rectangles. In: Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2013), pp. 400–409. IEEE (2013)
3. Adamaszek, A., Wiese, A.: A QPTAS for maximum weight independent set of polygons with polylogarithmically many vertices. In: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2014), pp. 645–656. SIAM (2014)
4. Agarwal, P.K., Mustafa, N.H.: Independent set of intersection graphs of convex objects in 2D. Comput. Geom. **34**(2), 83–95 (2006)
5. Agarwal, P.K., van Kreveld, M., Suri, S.: Label placement by maximum independent set in rectangles. Comput. Geom. **11**, 209–218 (1998)
6. Anagnostopoulos, A., Grandoni, F., Leonardi, S., Wiese, A.: Constant integrality gap LP formulations of unsplittable flow on a path. In: Integer Programming and Combinatorial Optimization (IPCO 2013), Lecture Notes in Computer Science, vol. 7801, pp. 25–36. Springer (2013)
7. Berman, P., DasGupta, B., Muthukrishnan, S., Ramaswami, S.: Efficient approximation algorithms for tiling and packing problems with rectangles. J. Algorithms **41**(2), 443–470 (2001)
8. Bonsma, P., Schulz, J., Wiese, A.: A constant factor approximation algorithm for unsplittable flow on paths. In: Proceedings of the 52th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2011), pp. 47–56 (2011)
9. Chalermsook, P., Chuzhoy, J.: Maximum independent set of rectangles. In: Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA ’09), pp. 892–901. SIAM (2009)
10. Chan, T.M., Har-Peled, S.: Approximation algorithms for maximum independent set of pseudo-disks. Discrete Comput. Geom. **48**(2), 373–392 (2012)
11. Clark, B.N., Colbourn, C.J., Johnson, D.S.: Unit disk graphs. Discrete Math. **86**(1), 165–177 (1990)
12. de Floriani, L., Magillo, P., Puppo, E.: Applications of computational geometry to geographic information systems. In: Sack, J.-R., Urrutia, J. (eds.) Handbook of Computational Geometry, pp. 333–388. North Holland, New York (2000)
13. Erlebach, T., Jansen, K., Seidel, E.: Polynomial-time approximation schemes for geometric intersection graphs. SIAM J. Comput. **34**(6), 1302–1323 (2005)
14. Fowler, R.J., Paterson, M.S., Tanimoto, S.L.: Optimal packing and covering in the plane are NP-complete. Inf. Process. Lett. **12**(3), 133–137 (1981)
15. Fox, J., Pach, J.: Computing the independence number of intersection graphs. In: Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2011), pp. 1161–1165. SIAM (2011)
16. Fukuda, T., Morimoto, Y., Morishita, S., Tokuyama, T.: Data mining with optimized two-dimensional association rules. ACM Trans. Database Syst. (TODS) **26**(2), 179–213 (2001)
17. Har-Peled, S.: Quasi-polynomial time approximation scheme for sparse subsets of polygons. In: Proceedings of the Thirtieth Annual Symposium on Computational Geometry (SoCG 2014), pp. 120–129. ACM (2014)
18. Hochbaum, D.S., Maass, W.: Approximation schemes for covering and packing problems in image processing and VLSI. J. ACM **32**, 130–136 (1985)

19. Imai, H., Asano, T.: Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane. J. Algorithms **4**(4), 310–323 (1983)
20. Khanna, S., Muthukrishnan, S., Paterson, M.: On approximating rectangle tiling and packing. In: Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 1998), pp. 384–393. SIAM (1998)
21. Lent, B., Swami, A., Widom, J.: Clustering association rules. In: Proceedings of the 13th International Conference on Data Engineering, pp. 220–231. IEEE (1997)
22. Nielsen, F.: Fast stabbing of boxes in high dimensions. Theor. Comput. Sci. **246**, 53–72 (2000)
23. Verweij, B., Aardal, K.: An optimisation algorithm for maximum independent set with applications in map labelling. In: Nešetřil, J. (ed.) Algorithms-ESA'99, pp. 426–437. Springer, Berlin (1999)