A QPTAS for Maximum Weight Independent Set of Polygons with Polylogarithmically Many Vertices

Anna Adamaszek* Andreas Wiese*

Abstract

The Maximum Weight Independent Set of Polygons (MWISP) problem is a fundamental problem in computational geometry. Given a set of weighted polygons in the two-dimensional plane, the goal is to find a set of pairwise non-overlapping polygons with maximum total weight. Due to its wide range of applications and connections to other problems, the MWISP problem and its special cases have been extensively studied both in the approximation algorithms and the computational geometry community. Despite a lot of research, its general case is not well-understood yet. Currently the best known polynomial time algorithm achieves an approximation ratio of $n^\epsilon$ [Fox and Pach, SODA 2011], and it is not even clear whether the problem is APX-hard. We present a $(1+\epsilon)$-approximation algorithm, assuming that each polygon in the input has at most a polylogarithmic number of vertices. Our algorithm has quasi-polynomial running time, i.e., it runs in time $2^{poly(log n,1/\epsilon)}$. In particular, our result implies that for this setting the problem is not APX-hard, unless $NP \subseteq DTIME(2^{poly(log n)})$.

We use a recently introduced framework for approximating maximum weight independent set in geometric intersection graphs. The framework has been used to construct a QPTAS in the much simpler case of axis-parallel rectangles. We extend it in two ways, to adapt it to our much more general setting. First, we show that its technical core can be reduced to the case when all input polygons are triangles. Secondly, we replace its key technical ingredient which is a method to partition the plane using only few edges such that the objects stemming from the optimal solution are evenly distributed among the resulting faces and each object is intersected only a few times. Our new procedure for this task is not even more complex than the original one and, importantly, it can handle the arising difficulties due to the arbitrary angles of the input polygons. Note that already this obstacle makes the known analysis for the above framework fail. Also, in general it is not well understood how to handle this difficulty by efficient approximation algorithms.

*Max-Planck-Institut für Informatik, Saarbrücken, Germany, {anna, awiese}@mpi-inf.mpg.de
1 Introduction

One of the classical problems in combinatorial optimization is the Maximum Independent Set problem. Given a graph, the goal is to select a subset of the vertices such that no two selected vertices are connected by an edge. The objective is to maximize the cardinality or the total weight of the selected vertices. While this is a fundamental problem, in its full generality it is essentially computationally intractable for efficient algorithms: the problem is \( \text{NP-hard} \) to approximate within a factor of \( O(n^{1-\epsilon}) \) for any \( \epsilon > 0 \) \[25\].

However, when restricting to special graph classes, it is possible to obtain polynomial time algorithms with much better approximation guarantees. An important class are intersection graphs of two-dimensional objects in the plane, i.e., where each vertex in the graph corresponds to a shape in the plane and two vertices are connected by an edge if and only if their corresponding shapes overlap. Without loss of generality, we can assume that the given shapes are (not necessarily convex) polygons. In this paper, we study the resulting Maximum Weight Independent Set of Polygons (MWISP) problem. It arises in many settings like map labeling \[3, 22, 24\], chip manufacturing \[16\], cellular networks \[11\], unsplittable flow \[4, 7\], or data mining \[14, 17, 19\].

Computationally, the MWISP problem is not well-understood. When interpreting the problem equivalently as finding a set of non-intersecting curves in the plane, the best known polynomial time algorithm achieves an approximation ratio of \( n^{\epsilon} \) \[13\]. On the other hand, the best known hardness result for the geometric setting is strong \( \text{NP-hardness} \). This holds already if all objects are straight line segments with at most two different angles \[18\]. Thus, there is a very large gap between the known approximation and hardness results for the MWISP problem. In this paper we completely close this gap under the assumption that the polygons do not have too many vertices (at most \( (\log n)^{O(1)} \) many) and when allowing quasi-polynomial running time.

1.1 Our Contribution

We present a \((1 + \epsilon)\)-approximation algorithm for the maximum weight independent set of polygons problem, running in quasipolynomial time \( 2^{\text{poly}(\log n, 1/\epsilon)} \), where we assume that each polygon has at most a polylogarithmic number of vertices. As mentioned above, in comparison the best known polynomial time algorithm for this problem achieves an approximation ratio of \( n^{\epsilon} \). To the best of our knowledge, no better approximation algorithms are known that run in quasi-polynomial time.

We use a recently introduced framework for approximating maximum weight independent set in geometric intersection graphs \[1\]. The framework has been introduced to construct a QPTAS in the much simpler case where all input objects are axis-parallel rectangles. However, for arbitrary polygons the problem is much more difficult and the techniques from \[1\] alone are by far not sufficient. There are two major obstacles we have to overcome: handling arbitrary angles and dealing with more complex polygons, i.e., polygons with a super-constant number of vertices. In fact, it is not well understood how to handle these obstacles by efficient algorithms. For instance, if all polygons are axis-parallel rectangles, many algorithms are known that achieve an \( O(\log n) \)-approximation ratio or better \[3, 6, 9, 10, 17, 21\]. However, already for straight line segments with arbitrary angles, that are a special case of non-axis-parallel rectangles, the best known polynomial time algorithm is the \( n^{\epsilon} \)-approximation mentioned above \[13\]. Generalizations of axis-parallel rectangles to polygons with more than just four axis-parallel edges are not much better understood either. In particular, the LP approach (which has been extensively exploited in the case of axis-parallel rectangles) does not work in a more general setting. Already for non-axis-parallel rectangles and axis-parallel L-shapes (i.e., polygons with six vertices each) the integrality gap of the LP can be as large as \( \Omega(n) \). We show how to overcome the two major obstacles discussed above and thus in particular contribute towards better algorithmic understanding of them.

Our algorithm is a dynamic program parametrized by an integer \( k \), where for our result we choose
$k = \text{poly}(\log n, 1/\epsilon)$. Its DP-table has an entry for each polygon in the plane (not necessarily part of the input) with at most $k$ edges. In order to bound the size of the DP-table, we require the corners of these polygons to be from some suitable polynomial size set. Each DP-cell represents the subproblem given by all input polygons that are contained in the polygon associated with the DP-cell. When computing the value for the cell, the dynamic program tries all combinations to partition its corresponding polygon into up to $k$ smaller polygons with at most $k$ edges each, and selects the most profitable combination according to the precomputed values for the DP-cells of the smaller polygons.

For bounding the approximation ratio of our DP, we first show that we can reduce the general setting to the special case where all polygons are triangles. In fact, it is even true that if for some parameter $k$ our algorithm is a $(1-\epsilon)$-approximation for triangles, it yields a $(1-K \cdot \epsilon)$-approximation for arbitrary $K$-gons. Observe that the latter implication cannot be applied to arbitrary algorithms that work well on triangles. Secondly, we show that for any feasible (and in particular the optimal) solution for a problem instance where all polygons are triangles there is a balanced cut with few edges that intersects only triangles with small total weight. Applying such a cut recursively for $O(\log n/\epsilon)$ levels eventually subdivides the input plane into pieces such that each of them contains only a single triangle of the optimal solution. Note that the DP has to guess these cuts, so it is very important that they have small complexity.

For showing that a balanced cut always exists, our technical core is to show that there is a partition of the plane into faces, using only a bounded number of edges, such that for each face of the partition the total weight of triangles from the optimal solution touching the face is relatively small and each triangle is intersected only a bounded number of times by edges of the faces. The techniques from [1] for axis-parallel rectangles entirely fail here. One key ingredient of the partition for rectangles is a procedure that stretches the input area non-uniformly along the two dimensions, ensuring that the rectangles contained in any thin vertical or horizontal stripe of the input have small total weight. The partition is then constructed so that each face is contained in a bounded number of such stripes. For triangles with arbitrary angles, a stretching procedure preserving the combinatorial structure of the input would turn them again into polygons with many bends. Even worse, when trying to adapt the construction from [1], the arbitrary angles of the triangles cause that one cannot guarantee that the faces are contained in a bounded number of thin horizontal or vertical stripes, or even in thin stripes with a small number of suitably chosen directions.

Therefore, we provide a completely new construction of the partition which is much more general than the specialized version for axis-parallel rectangles, and that is not even more complicated than the latter. Altogether, this yields a QPTAS for the MWISP problem for arbitrary $K$-gons with $K = \text{poly}(\log n)$. We note that our algorithm—the geometric dynamic program—might well be the right algorithmic technique for better polynomial time approximation algorithms for MWISP, when parametrized by $k = O(1)$.

### 1.2 Related Work

There is a lot of research on the maximum independent set problem for geometric shapes in the plane. For the case when all shapes are unit disks (i.e., the resulting intersection graph is a unit disk graph), polynomial time approximation schemes have been presented by Hunt III et al. [15], and by Hochbaum and Maass [16]. Subsequently, Nieberg, Hurink, and Kern [20] presented a PTAS that does not even need the geometric representation of the graph. Observe here that it is NP-hard to decide whether a given graph is a unit disk graph [8]. Using the geometric embedding, Erlebach, Jansen, and Seidel presented a PTAS for disks with arbitrary diameters, which generalizes to arbitrary fat objects [12].

When going beyond the fat objects, the problem becomes much less understood. Even for axis-parallel rectangles the best known polynomial time algorithms are $O(\log n/\log \log n)$-approximation for weighted case due to Chan and Har-Peled [10], and a $O(\log \log n)$-approximation for unweighted case by Chalermsook and Chuzhoy [9]. Prior to the latter results, many $O(\log n)$-approximation algorithms have been found [3, 6, 17, 21]. Very recently, a QPTAS has been presented [1].
For the general case, the best known result is a $n^\ell$-approximation algorithm for collections of curves due to Fox and Pach [13], which assumes that any two curves intersect only $\ell = O(1)$ times, i.e., that they are $\ell$-intersecting. Prior to this, Agarwal and Mustafa presented an algorithm for the special case of straight line segments that computes an independent set of size $(OPT/\log(2n/OPT))^{1/2}$ [2], which yields a worst case approximation ratio of $n^{1/2+o(1)}$. Note that it is NP-hard to decide whether a given graph can be represented as an intersection graph of a collection of curves (i.e., is a string graph) [23].

1.3 Problem Definition and Notation

We are given a set of $n$ polygons $P$ and an integer $K$. Each polygon $P_i \in P$ is a simple polygon specified by a sequence of vertices $p_{i,1}, \ldots, p_{i,\ell}$, for some $\ell \leq K$, each vertex $p_{i,j} = (x_{i,j}, y_{i,j})$ having integer coordinates $x_{i,j}, y_{i,j} \in \mathbb{N}$. Each polygon is considered as an open set. Also, each polygon $P_i$ has a weight $w(P_i) > 0$ associated with it. The goal is to find a maximum weight independent set of polygons, i.e., a set $P' \subseteq P$ such that for all pairs of polygons $P, P' \in P'$ with $P \neq P'$ it holds that $P \cap P' = \emptyset$. We will show that for $K = \text{poly}(\log n)$ this problem admits a QPTAS. Without loss of generality we assume that no three vertices of the input polygons lie on a straight line (i.e., the input points are in general position, see Appendix A for details).

For any two points $p, p'$ we define $L[p, p']$ to be the straight line segment connecting $p$ and $p'$. For two point sets $A, A'$ (which could be lines, polygons, etc.) we say that $A$ touches $A'$ if $A \cap A' \neq \emptyset$. We say that $A$ crosses $A'$ if $A \setminus A$ has at least two connected components. For a given problem instance (that will always be clear from the context) we assume w.l.o.g. that there is an integer $N$—not necessarily polynomially bounded—such that for all points $p_{i,j}$ it holds that $p_{i,j} \in I := [0, N] \times [0, N]$. For a set of polygons $P' \subseteq P$ we define $w(P') := \sum_{P \in P'} w(P)$. Due to space constraints, the proof of the next lemma, as well as some of the other proofs, are in the appendix.

**Lemma 1.** By losing at most a factor of $1 + O(\epsilon)$ we can assume that $1 \leq w(P) \leq n/\epsilon$ for each polygon $P \in P$.

2 Dynamic Program

We describe our dynamic program, parametrized by an integer parameter $k$, for approximating the MWISP problem. The dynamic program is a generalization of the DP presented in [1] for approximating the maximum weight independent set of axis-parallel rectangles (rather than general polygons).

Intuitively, the DP has an entry in its DP-table for each polygon with at most $k$ edges, possibly with holes, within the input square. To make the size of the table polynomially bounded, the corners of the polygons are chosen from a suitable, polynomially bounded set of points. The entry for each polygon $Q$ represents a good approximation on the optimal solution for the subproblem given by the input polygons from $P$ that are contained in $Q$. In order to compute an entry for a polygon $Q$, the dynamic program tries all possibilities to subdivide $Q$ into at most $k$ smaller polygons with at most $k$ edges each. In a leaf subproblem $Q$, at most one polygon from $P$ contained in $Q$ is chosen.

To define the DP formally, we first define the set of possible corners of the polygons in the DP-table, which we call the **DP-points**. For each polygon $P \in P$ we fix a triangulation of $P$ and we denote by $T(P)$ the resulting set of triangles. Denote by $E_T$ the set of all boundary edges of the triangles $T(P)$ for all polygons $P \in P$. (Notice that if all input polygons are triangles, $E_T$ is the set of their boundary edges.)

The set of **basic DP-points** contains the four corners of the input square $I$ and each intersection of a vertical line $\{(x, y) \mid y \in \mathbb{R}\}$ such that $x = x_{i,j}$ for some polygon $P_i \in P$ with a line $L \in E_T$ or with the horizontal boundary edge of the input square, see Figure 1. The set of **additional DP-points** contains all intersection points of pairs of line segments whose endpoints are basic DP-points, which are contained in $I$. When we
use the notion of DP-points, we refer to the set of all, basic and additional DP-points. Note that all corners of polygons from \( \mathcal{P} \) are basic DP-points and the number of all DP-points is upper bounded by \( O((nK)^4) \). Denote by \( \mathcal{Q} \) the set of all polygons with at most \( k \) edges and possibly with holes, whose corners are DP-points. In the DP-table, we have an entry for each polygon \( \mathcal{Q} \in \mathcal{Q} \). To distinguish polygons from the input \( \mathcal{P} \) and polygons of the DP-table, we will always denote the latter by \( \mathcal{Q}, \mathcal{Q}', \text{etc.} \).

**Proposition 2.** The number of DP-cells is at most \((nK)^O(k)\).

For each polygon \( \mathcal{Q} \in \mathcal{Q} \) the dynamic program computes a set \( \text{sol}(\mathcal{Q}) \) of pairwise non-touching polygons from the set \( \mathcal{P}_\mathcal{Q} := \{ P \in \mathcal{P} : P \subseteq \mathcal{Q} \} \). The set \( \text{sol}(\mathcal{Q}) \) is computed as follows. If \(|\mathcal{P}_\mathcal{Q}| \leq 1\), we set \( \text{sol}(P) := \mathcal{P}_\mathcal{Q} \) and terminate. Otherwise, we enumerate all possible partitions of \( \mathcal{Q} \) into at most \( k \) polygons from \( \mathcal{Q} \), see Figure 1. By Proposition 2 we have \(|\mathcal{Q}| \leq (nK)^O(k)\), so the number of potential partitions we need to consider is upper bounded by \((nK)^O(k^2)\). Also, for any set of at most \( k \) polygons in \( \mathcal{Q} \) we can check efficiently whether they form a partition of some larger polygon in \( \mathcal{Q} \) (e.g., we can use as a skeleton the subdivision of the plane obtained by drawing all line segments between any pair of DP-points). Let \( \{Q_1, \ldots, Q_{k'}\} \), with \( k' \leq k \), denote the partition that maximizes \( \sum_{i=1}^{k'} w(\text{sol}(Q_i)) \), and let \( \text{sol}'(Q) := \bigcup_{i=1}^{k'} \text{sol}(Q_i) \). We set \( \text{sol}(Q) := \text{sol}'(Q) \) if \( w(\text{sol}'(Q)) > \max_{P \in \mathcal{P}_\mathcal{Q}} w(P) \), and otherwise \( \text{sol}(Q) := \{P_{\max}\} \) for a polygon \( P_{\max} \in \mathcal{P}_\mathcal{Q} \) with maximum profit in \( \mathcal{P}_\mathcal{Q} \). At the end, the algorithm outputs the value in the DP-cell which corresponds to the polygon containing the entire input region. We call this algorithm \( \text{GEO-DP} \), like its specialization for axis-parallel rectangles given in [1].

**Proposition 3.** When parametrized by \( k \), the running time of \( \text{GEO-DP} \) is upper bounded by \((nK)^O(k^2)\).

### 2.1 Approximation Ratio

The core of our reasoning is showing that for any set of pairwise non-touching polygons there is a balanced cheap cut.

**Definition 4.** Let \( \ell \in \mathbb{N} \) and \( \alpha \in \mathbb{R} \) with \( 0 < \alpha < 1 \). Let \( \mathcal{P} \) be a set of pairwise non-touching polygons. A polygon \( \Gamma \) is a balanced \( \alpha \)-cheap \( \ell \)-cut if:

- \( \Gamma \) has at most \( \ell \) edges,
- for the set of all polygons \( \mathcal{P}' \subseteq \mathcal{P} \) intersecting the boundary of \( \Gamma \) we have \( w(\mathcal{P}') \leq \alpha \cdot w(\mathcal{P}) \),
- for the set of all polygons \( \mathcal{P}_{\text{in}} \subseteq \mathcal{P} \) contained in \( \Gamma \) it holds that \( w(\mathcal{P}_{\text{in}}) \leq 2/3 \cdot w(\mathcal{P}) \), and

Figure 1: Left: Basic DP-points are denoted by fat dots. The dashed line indicates the triangulation of the left polygon. Right: The dynamic program GEO-DP tries all possibilities to subdivide the polygon \( \mathcal{Q} \) into at most \( k \) smaller polygons with at most \( k \) edges each. One partition is shown in the figure.
• for the set of all polygons \( \mathcal{P}_{\text{out}} \subseteq \overline{\mathcal{P}} \) contained in the complement of \( \Gamma \), i.e., in \( I \setminus \Gamma \), it holds that \( w(\mathcal{P}_{\text{out}}) \leq 2/3 \cdot w(\overline{\mathcal{P}}) \).

The next lemma shows that if for any set of polygons there is a good enough balanced cheap cut, then GEO-DP has a good approximation ratio.

**Lemma 5.** Let \( \epsilon > 0 \) and choose \( \alpha := \frac{\epsilon}{\log(n/\epsilon)} \). Let \( \ell \geq 4 \) be a value such that for any set \( \overline{\mathcal{P}} \) of at most \( n \) pairwise non-touching polygons with at most \( K \) vertices each there exists a balanced \( \alpha \)-cheap \( \ell \)-cut whose vertices are basic DP-points, or there is a polygon \( P \in \overline{\mathcal{P}} \) such that \( w(P) \geq 1/3 \cdot w(\overline{\mathcal{P}}) \). Then GEO-DP has an approximation ratio of at most \( 1 + O(\epsilon) \) when parametrized by \( k = O(\ell^2 \cdot \log^2(n/\epsilon)) \).

**Proof sketch.** Let \( \mathcal{P}_{\text{opt}} \subseteq \mathcal{P} \) denote the optimal solution for a problem instance \( \mathcal{P} \). We start with the polygon \( Q_0 := \{(0, N) \times [0, N]\} \). If there is a polygon \( P \in \mathcal{P}_{\text{opt}} \) with \( w(P) \geq 1/3 \cdot w(\mathcal{P}_{\text{opt}}) \) then GEO-DP guesses the partition \( Q_1 := Q_0 \setminus P \) and \( Q_2 := P \) and recurses on \( Q_1 \). Note that \( w(Q_1) \leq 2/3 \cdot w(\mathcal{P}_{\text{opt}}) \). Otherwise, GEO-DP guesses a balanced \( \alpha \)-cheap \( \ell \)-cut \( \Gamma \) for \( \mathcal{P}_{\text{opt}} \) and recurses on \( Q_1' := \Gamma \) and \( Q_2' := Q_0 \setminus \Gamma \). Note that \( w(Q_1') \leq 2/3 \cdot w(\mathcal{P}_{\text{opt}}) \) and \( w(Q_2') \leq 2/3 \cdot w(\mathcal{P}_{\text{opt}}) \), and the total weight of intersected polygons from \( \mathcal{P}_{\text{opt}} \) is upper bounded by \( \alpha \cdot w(\mathcal{P}_{\text{opt}}) = \frac{\alpha}{\log(n/\epsilon)} \cdot w(\mathcal{P}_{\text{opt}}) \). We continue recursively for \( O(\log(n/\epsilon)) \) levels until we obtain subproblems containing at most one polygon from \( \mathcal{P}_{\text{opt}} \) each (recall that \( w(\mathcal{P}_{\text{opt}}) \leq n^2/\epsilon \), see Lemma 1). As in each level we intersect, i.e., lose, polygons of total weight at most \( \alpha \cdot w(\mathcal{P}_{\text{opt}}) \), the total approximation ratio is \( 1/(1 - \alpha)^{O(\log(n/\epsilon))} \leq 1 + O(\epsilon) \).

Each polygon \( Q_i \) inducing a subproblem in each level can be expressed as an intersection of at most \( O(\log(n/\epsilon)) \) polygons with at most \( \ell \) edges each, all corners of those being basic DP-points. Hence, the resulting polygon has at most \( k \) edges and all its corners are basic or additional DP-points. Also, when applying a cut \( \Gamma \) on such a subproblem \( Q_i \), the number of resulting connected components of \( \Gamma \) and \( Q_i \setminus \Gamma \) is upper bounded by \( k \). Hence, when executed with parameter \( k \), eventually GEO-DP will guess the sequence of cuts described above and thus compute a \((1 + O(\epsilon))\)-approximative solution.

**3 Finding a Balanced Cheap Cut**

In this section we show that for any set of pairwise non-touching polygons with at most \( K = \text{poly}(\log n) \) vertices each we can always find a balanced cheap cut with good parameters. Together with Lemma 5, this will prove that the algorithm GEO-DP parametrized by a suitable parameter \( k \) is a QPTAS, assuming that the input consists of polygons with at most \( K \) vertices each.

Our reasoning has three steps. First, we show that we can reduce the problem of finding the desired cut to the special case when all polygons are triangles. In the second step we construct a partition of the plane that uses only a bounded number of edges and which has some useful properties. Finally, we apply a planar graph separator theorem from [5] to the partition, which then yields the cut.

**3.1 Reduction to Triangles**

We show that up to a factor of \( K \) in the cost of the cut and its complexity, it suffices to find a good cut for the case that all polygons are triangles.

**Lemma 6.** Assume that for any \( \delta > 0 \) and for any set of pairwise non-touching triangles \( \mathcal{T} \) such that \( \max_{T \in \mathcal{T}} w(T) < w(\mathcal{T})/3 \) there is a balanced \( \delta \)-cheap \( f(\delta) \)-cut with vertices at basic DP-points for some function \( f \). Then for any \( \delta > 0 \) and for any set of pairwise non-touching polygons \( \mathcal{P} \) with at most \( K \) edges each such that \( \max_{P \in \mathcal{P}} w(P) < w(\mathcal{P})/3 \) there is a balanced \( K \cdot \delta \)-cheap \( K \cdot f(\delta) \)-cut with vertices at basic DP-points.
Proof. Consider a set of pairwise non-touching polygons \( \mathcal{P} \) with at most \( K \) edges each and let \( \delta > 0 \). We triangulate each polygon \( P \in \mathcal{P} \) into at most \( K - 2 \) triangles \( \mathcal{T}(P) \), using the same triangulation as used for choosing the basic DP-points, and distribute the weight of \( P \) equally among the triangles in \( \mathcal{T}(P) \). Let \( T := \bigcup_{P \in \mathcal{P}} \mathcal{T}(P) \) denote the resulting set of pairwise non-touching triangles. We have that \( \max_{T \in T} w(T) \leq \max_{P \in \mathcal{P}} w(P) < w(P)/3 = w(T)/3 \). By assumption, there is a balanced \( \delta \)-cheap \( f(\delta) \)-cut \( \Gamma \) for \( T \) with vertices at basic DP-points. Notice that the sets of basic DP-points for \( \mathcal{P} \) and \( T \) are the same. We will transform \( \Gamma \) to a \( K \cdot \delta \)-cheap \( K \cdot f(\delta) \)-cut \( \Gamma' \) for \( \mathcal{P} \).

An edge \( e \) of \( \Gamma \) can cross a polygon \( P \in \mathcal{P} \) in two ways. The first case is that \( e \) touches at least one triangle \( T \in \mathcal{T}(P) \). Since \( \Gamma \) is \( \delta \)-cheap, the total weight of triangles touched by edges of \( \Gamma \) is upper bounded by \( \delta \cdot w(T) \). Therefore, the total weight of polygons \( P \in \mathcal{P} \) such that at least one triangle of \( \mathcal{T}(P) \) is touched by an edge of \( \Gamma \) is upper bounded by \( \delta (K - 2) \cdot w(T) \leq \delta K \cdot w(P) \). The other case is that \( e \) crosses \( P \), but it does not cross any triangle from \( \mathcal{T}(P) \) (see Figure 8 in the appendix). Since we assumed that no three vertices of the input polygons lie on a straight line (and edges of \( \Gamma \) are straight lines), each edge \( e \) of \( \Gamma \) can cross at most one polygon \( P(e) \in \mathcal{P} \) in this way. We fix this by replacing \( e \) by at most \( K \) edges such that \( P(e) \) is circumscribed and still for each side of the cut we have the property that the weight of contained polygons is at most \( 2/3 \cdot w(P) \). Note that here we use that \( w(P(e)) < w(P)/3 \). All newly introduced vertices of the cut \( \Gamma' \) are vertices of input polygons \( \mathcal{P} \), and so they are basic DP-points. We perform this operation for each edge of \( \Gamma \). Denote by \( \Gamma' \) the resulting cut for \( \mathcal{P} \). By construction, an edge \( e' \) of \( \Gamma' \) can cross a polygon \( P \) only if there is an edge \( e \) of \( \Gamma \) that crosses a triangle \( T \in \mathcal{T}(P) \). Hence, \( \Gamma' \) is a balanced \( K \cdot \delta \)-cheap \( K \cdot f(\delta) \)-cut, and all vertices of \( \Gamma' \) are basic DP-points. \( \square \)

In fact, with similar arguments as in the above lemma one can show that if GEO-DP parametrized by \( k \) yields a \( (1 - \epsilon)^{-1} \)-approximation algorithm for triangles, then GEO-DP parametrized by some parameter in \( \text{poly}(k, K) \) is a \( (1 - K \cdot \epsilon)^{-1} \)-approximation algorithm for polygons with at most \( K \) edges. Similarly as above, the idea is to triangulate each polygon in the input, consider an execution of GEO-DP on this triangulation, and circumvent polygons \( P \) that are touched without any of their triangles \( \mathcal{T}(P) \) being touched. Hence, an alternative way of showing that GEO-DP is a QPTAS for MWISP if \( K \leq (\log n)^{O(1)} \) is to prove that for triangles it is a \( (1 + \epsilon) \)-approximation with running time \( n^{\text{poly}(\log n, 1/\epsilon)} \), and then apply the above reasoning. Note that the actual dependence of \( \epsilon \) on the running time is crucial for this line of reasoning.

Following the (other) approach suggested by Lemma 6 in the remainder of this section we prove that for any set of triangles \( \mathcal{T} \) and for any \( \delta > 0 \) there is a balanced \( O(\delta) \)-cheap \( \left( \frac{1}{\delta} \right)^{O(1)} \)-cut.

### 3.2 Partitioning the Plane

Suppose we are given a set \( \mathcal{T} \) of pairwise non-touching triangles. Denote by \( \mathcal{E} \) the set of all boundary edges of the triangles in \( \mathcal{T} \). For constructing the cut, we associate a weight \( \bar{w}(p_{i,j}) \) with each vertex \( p_{i,j} \) of a triangle from \( \mathcal{T} \). For each triangle \( T_i \), we define \( \bar{w}(p_{i,j}) := w(T_i)/3 \) for each point \( p_{i,j} \), i.e., we equally distribute the weight of \( T_i \) among its vertices. For any area \( C \subseteq I \) we define \( \bar{w}(C) := \sum_{p_{i,j} \in C} \bar{w}(p_{i,j}) \).

We construct the partition in three steps:

1. We subdivide the plane into \( O(1/\delta^2) \) vertical stripes such that each stripe contains points with total weight smaller than \( w(T) \cdot \delta^2 \). The stripes are disjoint open sets.

2. We subdivide each vertical stripe, along the lines from \( \mathcal{E} \) crossing the stripe from left to right, into \( O(1/\delta^4) \) cells. Each cell has either no lines in \( \mathcal{E} \) crossing it from left to right, or all triangles touching the cell have total weight at most \( w(T) \cdot \delta^4 \). The cells, similarly as stripes, are open sets.

3. We transform the collection of cells into a subdivision of the plane such that each face of the subdivision surrounds exactly one triangle or touches triangles of total weight at most \( O(\delta^2 \cdot w(T)) \), and each triangle in \( \mathcal{T} \) is intersected only \( O(1) \) times.
We construct the vertical stripes as maximal stripes of the input square which contain points of total weight smaller than $\delta$. Later, we will apply the planar graph separator theorem from [5] to the subdivision to obtain a cut with the desired properties.

3.2.1 Partition into Cells

We construct the vertical stripes as maximal stripes of the input square which contain points of total weight smaller than $\delta^2 \cdot w(T)$.

We proceed iteratively as follows. First, we define $x_0 := 0$ and we set $x_1$ such that $\bar{w}((x_0, x_1) \times [0, N]) < \delta^2 \cdot w(T)$ and $\bar{w}((x_0, x_1) \times [0, N]) \geq \delta^2 \cdot w(T)$. Such $x_1$ is uniquely defined. We create a stripe $S_1 := (x_0, x_1) \times [0, N]$. Iteratively, we define $x_i$ such that $\bar{w}((x_{i-1}, x_i) \times [0, N]) < \delta^2 \cdot w(T)$ and $\bar{w}((x_{i-1}, x_i) \times [0, N]) \geq \delta^2 \cdot w(T)$ and we set $S_i := (x_{i-1}, x_i) \times [0, N]$. We stop when we have defined a value $x_{i^*} - 1$ such that $\bar{w}((x_{i^*}, N) \times [0, N]) < \delta^2 \cdot w(T)$. In that case we set $x_{i^*} = N$, we define $S_{i^*} := (x_{i^*}, x_{i^*}) \times [0, N]$ and we terminate. We obtain the following property.

**Proposition 7.** There is a subdivision of the input square $I$ into a set of vertical stripes $S_1, ..., S_{i^*}$ such that $i^* \leq 1/\delta^2$ and for each stripe $S_i$ we have $\bar{w}(S_i) < \delta^2 \cdot w(T)$.

As a next step, we subdivide each stripe $S_i$, along the lines of $E$ crossing $S_i$ from left to right, into a collection of cells. We define the subdivision in such a way that each cell has either no lines in $E$ crossing it from left to right, or all triangles touching the cell have total weight at most $w(T) \cdot \delta^4$.

Consider a stripe $S_i = (x_{i-1}, x_i) \times [0, N]$. Each cell for the stripe $S_i$ will be a polygon with four edges, whose four consecutive corners are of the form $(x_{i-1}, y_1^L), (x_{i-1}, y_2^L), (x_i, y_2^R), (x_i, y_1^R)$ for some values $y_1^L, y_2^L, y_1^R, y_2^R$ such that $L[(x_{i-1}, y_1^L) \times (x_i, y_1^R)]$ and $L[(x_{i-1}, y_2^L) \times (x_i, y_2^R)]$ are subsegments of lines in $E$, or of the boundary of $I$ (see Figure 2). We define the set $E_i$ to be the set of all lines in $E$ which cross $S_i$ from left to right. The subdivision of $S_i$ into cells is obtained by selecting a set $\hat{E}_i \subseteq E_i$ whose elements then yield the top and bottom boundaries of the cells. This selection is done in a straightforward manner.

**Lemma 8.** For any stripe $S_i$ there is a set $\hat{E}_i \subseteq E_i$ with $|\hat{E}_i| \leq O(\frac{1}{\delta^4})$ such that for any connected component $C$ of $S_i \setminus \hat{E}_i$ we have that $C \cap \hat{E}_i = \emptyset$ or the total weight of triangles touching $C$ is at most $\delta^4 \cdot w(T)$.

We define $L_0 := \cup_i \hat{E}_i$. These lines subdivide the stripes into cells. An area $C \subseteq I$ is a cell if there is a stripe $S_i$ such that $C$ is a connected component of $S_i \setminus \hat{E}_i$. We call a cell $C$ dense if $C \cap \hat{E}_i = \emptyset$ and light
Eventually, we have obtained a collection of lines \( L \) that connect a line \( L \in L_0 \) with the top boundary of \( I \). The set of lines \( L_0 \cup L_{\text{ext}} \) subdividing the input area. The the lines from \( L_0 \) are drawn in black, the lines from \( L_{\text{ext}} \) are gray. The gray lines in the interior of the two large triangles will be removed while transforming \( L_0 \cup L_{\text{ext}} \) into \( L \).

otherwise, see Figure 2 As the endpoints of all lines in \( L_0 \) are vertices of the input polygons, they are basic DP-points.

**Proposition 9.** The number of lines in \( L_0 \) is upper bounded by \( O(1/\delta^0) \). The endpoints of lines in \( L_0 \) are basic DP-points.

### 3.2.2 Creating a Subdivision of the Plane

Starting with the lines in \( L_0 \) we create a partition of the plane with some useful properties that will allow us later to find a balanced cheap cut. To this end, we construct a set of lines \( L_{\text{ext}} \) that complete \( L_0 \) to a subdivision of \( I \). The endpoints of all lines in \( L_{\text{ext}} \) will be at basic DP-points. Initially, we define \( L_{\text{ext}} \) to be the four boundary lines of \( I \).

Recall that we have two types of cells: dense and light cells. First, we consider the dense cells, and for each dense cell \( C \) we perform the following operation. Suppose that the consecutive corners of \( C \) are the points \((x_{i-1}, y_1^L), (x_{i-1}, y_2^R), (x_i, y_2^R), (x_i, y_1^R)\), see Figure 2. We add to \( L_{\text{ext}} \) the two vertical lines \( L[(x_{i-1}, y_1^L), (x_{i-1}, y_2^R)] \) and \( L[(x_i, y_1^R), (x_i, y_2^R)] \).

In the second step, we take each line \( L \in L_0 \) and add a sequence of lines to \( L_{\text{ext}} \) in order to connect both endpoints of \( L \) with another line in \( L_0 \cup L_{\text{ext}} \). While doing this we carefully ensure that the number of lines in \( L_{\text{ext}} \) is upper bounded by \( O(1/\delta^0) \).

Let \( L_0 \in L_0 \). The idea is to define a set of lines which connect \( L_0 \) to the right with either another line in \( L_0 \) or with the top or right boundary of \( I \). Let \( x_i \) be the rightmost value in \( \{x_1, \ldots, x_i\} \) such that \( L_0 \) touches the vertical line \( \{x_i\} \times [0, N] \), and let \( p \) be the point where \( L_0 \) touches this line. If \( p \) is contained in the top or right boundary of \( I \) or in a line \( L \in L_0 \) that crosses \( S_{i+1} \), we stop. Now suppose that the line \( L' := L[p, (x_i, N)] \) does not touch an edge in \( \mathcal{E} \) that crosses \( S_{i+1} \). In that case we define \( L'_0 := L' \) and we stop. Otherwise, suppose that \( L' \) touches an edge \( L'' \in \mathcal{E} \) crossing \( S_{i+1} \) at a point \( p' \) and \( p' \) is the bottom-most point with that property (possibly \( p' = p \)). We define \( L'_0 := L[p, p'] \). If \( L'' \in L_0 \), or if \( p' \) is at the top boundary of \( I \), we stop. Otherwise, we define \( L_1 := L'' \) and we continue iteratively with \( L_1 \). Eventually, we have obtained a collection of lines \( L_0, L'_0, L_1, L'_1, \ldots, L_{\ell}, L'_\ell \) where \( L'_i \) touches the outside boundary of \( I \) or a line in \( L_0 \). Note that all lines \( L'_i \) are vertical. We add the set \( \{L'_0, L_1, L'_1, \ldots, L_{\ell}, L'_\ell\} \) to
\[ L_{\text{ext}} \]. See Figure 3 for a sketch.

Similarly, we connect \( L_0 \) to the left with another line in \( L_0 \cup L_{\text{ext}} \) or with the bottom or left boundary of \( I \). We do this procedure with every line \( L \in L_0 \). In case that \( L_{\text{ext}} \) contains two lines \( L, L' \) with \( |L \cap L'| > 1 \) (i.e., they are parallel and overlap), we replace \( L \) and \( L' \) by \( L \cup L' \).

**Proposition 10.** The number of lines in \( L_0 \cup L_{\text{ext}} \) is upper bounded by \( O(1/\delta^8) \). The endpoints of lines in \( L_0 \cup L_{\text{ext}} \) are basic DP-points.

The lines in \( L_0 \cup L_{\text{ext}} \) give a subdivision of the input square \( I \) (see Figure 3). By construction, the only lines that touch triangles from \( \mathcal{T} \) are vertical lines in \( L_{\text{ext}} \), and if a line \( L \in L_{\text{ext}} \) touches a triangle \( T \in \mathcal{T} \), then \( L \) crosses \( T \) (recall here that the triangles are open sets). We want to ensure that each triangle is crossed only \( O(1) \) times. However, right now it can happen that a triangle is crossed a superconstant number of times. We will show that if a triangle \( T \) is crossed by more than four lines in \( L_{\text{ext}} \), then there must be a dense cell \( C \) such that \( C \subseteq T \). In such a case we say that \( T \) *owns* the cell \( C \). We denote by \( \mathcal{T}_{\text{own}} \) the set of triangles from \( \mathcal{T} \) which own some cell.

**Lemma 11.** Let \( T \in \mathcal{T} \) be a triangle which is crossed by more than four lines in \( L_{\text{ext}} \). Then \( T \in \mathcal{T}_{\text{own}} \).

**Proof.** Suppose that there is a triangle \( T \in \mathcal{T} \) that is crossed by five lines in \( L_{\text{ext}} \). As \( T \) has three corners, there must be a vertical line \( L \in L_{\text{ext}} \) crossing \( T \) with an \( x \)-coordinate \( x_i \) such that neither \( S_i \) nor \( S_{i+1} \) contain a corner of \( T \). By construction, \( L \) cannot have been added when connecting the lines in \( L_0 \) with the boundary of \( I \). Hence, \( L \) has been added due to a dense cell \( C \) with \( C \subseteq S_i \) or \( C \subseteq S_{i+1} \). Assume the former is true. As \( S_i \) does not contain a corner of \( T \), two edges of \( T \) must cross \( S_i \). As \( T \cap S_i \subseteq C \) and \( C \) is a dense cell, this implies that \( T \cap S_i = C \). Thus, \( T \) owns \( C \) and therefore \( T \in \mathcal{T}_{\text{own}} \).

Since the total number of cells, similarly as the total number of lines in \( L_0 \), is upper bounded by \( O((1/\delta)^6) \) (see Proposition 9), this is also an upper bound on the number of triangles that are crossed by more than four lines. We change the subdivision given by the lines \( L_0 \cup L_{\text{ext}} \) by “cutting out” every triangle \( T \in \mathcal{T}_{\text{own}} \). For constructing the new set of lines \( L \), let \( L' \) be the set of all boundary edges of the triangles in \( \mathcal{T}_{\text{own}} \), together with the line segments given by \( L \setminus \bigcup_{T \in \mathcal{T}_{\text{own}}} T \) for each line \( L \in L_0 \cup L_{\text{ext}} \). We add to \( L \) the lines resulting from subdividing \( L' \) so that any two lines can touch only at their endpoints. Formally, we add to \( L \) each line \( L[p, p'] \) such that \( L[p, p'] \) is contained in a line in \( L' \), each of the points \( p \) and \( p' \) is either an endpoint of a line in \( L' \), or a point where two lines in \( L' \) touch, and no line in \( L' \) touches \( L[p, p'] \) at any point other than \( p \) and \( p' \).

**Proposition 12.** The number of lines in \( L \) is upper bounded by \( (1/\delta)^O(1) \). The endpoints of lines in \( L \) are basic DP-points.

The lines in \( L \) induce a connected graph which gives us a subdivision of the plane. We call each connected component of \( I \setminus L \) a face. Let \( \mathcal{F} \) denote the set of all faces. We prove now some crucial properties of \( \mathcal{F} \) that will allow us later to find a balanced cheap cut as a part of the subdivision given by \( L \). One important property of the subdivision is that for each face \( F \in \mathcal{F} \) we can bound the weight of the triangles touching it, unless \( F \) coincides with some triangle \( T \in \mathcal{T} \).

**Lemma 13.** Consider a face \( F \in \mathcal{F} \). Either \( F = T \) for some triangle \( T \) or the total weight of triangles touching \( F \) is upper bounded by \( 3\delta^2 \cdot w(T) \).

**Proof sketch.** If \( F \) does not coincide with any triangle \( T \), then there are two cases. The first case is that \( F \) touches a dense cell \( C \). In that case, due to the vertical lines added for \( C \), the face \( F \) is in fact contained in \( C \). As no triangles cross \( C \), the weight of triangles touching \( F \) is bounded by \( 3 \cdot w(C) \leq 3\delta^2 \cdot w(T) \). If \( F \) does not touch any dense cell, then one can show that it touches at most one (light) cell per stripe \( S_i \). As there are at most \( 1/\delta^2 \) stripes and the total weight of triangles touching any light cell is at most \( \delta^4 \cdot w(T) \), the weight of triangles touching \( F \) is at most \( \delta^2 \cdot w(T) \).
The second important property of our subdivision is that each triangle is touched by at most a constant number of lines in $\mathcal{L}$.

**Lemma 14.** For each triangle $T \in \mathcal{T}$ there are at most four lines $L \in \mathcal{L}$ touching $T$.

*Proof.* From the construction of $\mathcal{L}$ we know that if a triangle is touched by a line $L \in \mathcal{L}$, then $T$ is crossed by $L$. Moreover, the only lines from $\mathcal{L}$ which can cross triangles from $\mathcal{T}$ are the lines which were originally in $\mathcal{L}_{\text{ext}}$. If a triangle has been crossed by more than four lines of $\mathcal{L}_{\text{ext}}$, then it owned a cell and has been cut out when constructing $\mathcal{L}$ (see Lemma 11). Hence, for each triangle $T \in \mathcal{T}$ there are at most four lines $L \in \mathcal{L}$ touching $T$. \hfill \qed

### 3.3 Obtaining the Cut

We use the subdivision given by the lines in $\mathcal{L}$ in order to create a balanced cheap cut. We construct a planar graph as follows. For each point $p$ that is an endpoint of a line in $\mathcal{L}$ we create a vertex $v_p$. By Proposition 12 such a point $p$ must be a basic DP-point. For each line $L[p, p'] \in \mathcal{L}$ we create an edge $e_{p,p'} := \{v_p, v_{p'}\}$. We define the cost $c_{p,p'}$ of each edge $e_{p,p'}$ to be the total weight of triangles in $\mathcal{T}$ touching $L[p, p']$.

For defining the weights of the faces, if a triangle $T \in \mathcal{T}$ has non-empty intersection with $m$ faces of the graph, then each of these faces obtains a $1/m$-fraction of $w(T)$. Denote by $G = (V, E)$ the resulting graph. The proposition below follows from Proposition 12 and Lemmas 13 and 14, using that $\delta < 1/3$.

**Proposition 15.** The total cost of the edges $E$ is bounded by $O(w(\mathcal{T}))$ and the total weight of the faces of $G$ is $w(\mathcal{T})$. The total number of edges is bounded by $(\frac{1}{\delta})^{O(1)}$. The cost of each face is bounded by $\frac{1}{\delta} \cdot w(\mathcal{T})$. The boundary of each face touches triangles of weight at most $3\delta^2 \cdot w(\mathcal{T})$.

For extracting a balanced cheap cut from $G$ we use the following theorem due to Arora et al. [5]. We note that this theorem has been used in a similar way in [1]. A $V$-cycle $C$ is a Jordan curve in the embedding of a given planar graph $G$ which might go along the edges of $G$ and also might cross faces of $G$. The parts of $C$ crossing an entire face of $G$ are called face edges.

**Theorem 16 ([5]).** Let $G$ denote a planar, embedded graph with weights on the vertices and faces and with costs on the edges. Let $W$ denote the total weight, and $M$ the total cost of the graph. Then, for any parameter $\tilde{k}$, we can find in polynomial time a separating $V$-cycle $C$ such that the interior and exterior of $C$ each has weight at most $2W/3$, $C$ uses at most $\tilde{k}$ face edges, and $C$ uses ordinary edges of total cost $O(M/\tilde{k})$.

We apply Theorem 16 with $\tilde{k} = O(1/\delta)$ to our graph $G$, which yields the proof of the following lemma. In combination with Lemmas 5 and 6 it then yields our main theorem.

**Lemma 17.** Let $\delta > 0$ and let $\mathcal{T}$ be a set of pairwise non-touching triangles in the plane such that $w(T) < w(\mathcal{T})/3$ for each $T \in \mathcal{T}$. Then there exists a balanced $O(\delta)$-cheap $(\frac{1}{\delta})^{O(1)}$-cut with corners at basic DP-points.

**Theorem 18.** Let $K \in \mathbb{N}$. There exists a $(1 + \epsilon)$-approximation algorithm with a running time of at most $(nK)^{(\frac{1}{2}K \log n)^{O(1)}}$ for the Maximum Weight Independent Set of Polygons problem, assuming that each polygon has at most $K$ vertices. Hence, for the case that $K \leq (\log n)^{O(1)}$ there is a QPTAS for the Maximum Weight Independent Set of Polygons problem.
References

[1] A. Adamaszek and A. Wiese. Approximation schemes for maximum weight independent set of rectangles. In Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2013). IEEE, 2013. To appear. Technical report available online: arXiv:1307.1774.

[2] P. K. Agarwal and N. H. Mustafa. Independent set of intersection graphs of convex objects in 2D. Computational Geometry, 34(2):83–95, 2006.

[3] P. K. Agarwal, M. van Kreveld, and S. Suri. Label placement by maximum independent set in rectangles. Computational Geometry, 11:209–218, 1998.

[4] A. Anagnostopoulos, F. Grandoni, S. Leonardi, and A. Wiese. Constant integrality gap LP formulations of unsplittable flow on a path. In Integer Programming and Combinatorial Optimization, pages 25–36. Springer, 2013.

[5] S. Arora, M. Grigni, D. Karger, P. Klein, and A. Woloszyn. A polynomial-time approximation scheme for weighted planar graph TSP. In Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms, SODA ’98, pages 33–41, Philadelphia, PA, USA, 1998. Society for Industrial and Applied Mathematics.

[6] P. Berman, B. DasGupta, S. Muthukrishnan, and S. Ramaswami. Efficient approximation algorithms for tiling and packing problems with rectangles. Journal of Algorithms, 41(2):443 – 470, 2001.

[7] P. Bonsma, J. Schulz, and A. Wiese. 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), pages 47–56, 2011.

[8] H. Breu and D. G. Kirkpatrick. Unit disk graph recognition is NP-hard. Computational Geometry. Theory and Applications, 9(1-2):3–24, Jan. 1998.

[9] P. Chalermsook and J. Chuzhoy. Maximum independent set of rectangles. In Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA ’09), pages 892–901. SIAM, 2009.

[10] T. M. Chan and S. Har-Peled. Approximation algorithms for maximum independent set of pseudodisks. Discrete & Computational Geometry, 48(2):373–392, 2012.

[11] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. Discrete mathematics, 86(1):165–177, 1990.

[12] T. Erlebach, K. Jansen, and E. Seidel. Polynomial-time approximation schemes for geometric intersection graphs. SIAM Journal on Computing, 34(6):1302–1323, 2005.

[13] J. Fox and J. Pach. Computing the independence number of intersection graphs. In Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA ’11, pages 1161–1165. SIAM, 2011.

[14] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Data mining with optimized two-dimensional association rules. ACM Transactions on Database Systems (TODS), 26(2):179–213, 2001.

[15] H. B. Hunt III, M. V. Marathe, V. Radhakrishnan, S. S. Ravi, D. J. Rosenkrantz, and R. E. Stearns. NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. J. Algorithms, 26(2):238–274, 1998.
[16] D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM*, 32(1):130–136, Jan. 1985.

[17] S. Khanna, S. Muthukrishnan, and M. Paterson. On approximating rectangle tiling and packing. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA ’98)*, pages 384–393. SIAM, 1998.

[18] J. Kratochvíl and J. Nešetřil. Independent set and clique problems in intersection-defined classes of graphs. *Commentationes Mathematicae Universitatis Carolinae*, 31(1):85–93, 1990.

[19] B. Lent, A. Swami, and J. Widom. Clustering association rules. In *Proceedings of the 13th International Conference on Data Engineering*, pages 220–231. IEEE, 1997.

[20] T. Nieberg, J. L. Hurink, and W. Kern. A robust PTAS for maximum weight independent sets in unit disk graphs. In *30th International Workshop of Graph-Theoretic Concepts in Computer Science (WG 2004)*, volume 3353 of *LNCS*, pages 214–221, Berlin, 2004. Springer-Verlag.

[21] F. Nielsen. Fast stabbing of boxes in high dimensions. *Theor. Comp. Sc.*, 246:53 – 72, 2000.

[22] J.-R. Sack and J. Urrutia. *Handbook of computational geometry*. North Holland, 1999.

[23] M. Schaefer, E. Sedgwick, and D. Štefankovič. Recognizing string graphs in np. *Journal of Computer and System Sciences*, 67(2):365–380, 2003.

[24] B. Verweij and K. Aardal. An optimisation algorithm for maximum independent set with applications in map labelling. In *Algorithms-ESA’99*, pages 426–437. Springer, 1999.

[25] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3:103–128, 2007.
Figure 4: Shifting vertices of the polygons so that no three vertices are lying on one line. Some vertices are moved by a small distance towards the interior of their polygons. As a consequence of this operation, no two vertices of the polygons are coincident. The combinatorial structure of the input remains the same.

A Position of Input Points

We show that w.l.o.g. we can assume that no three points of the input are lying on a straight line (and, as a consequence, no two points of the input are coincident).

Lemma 19. Let \( K, N \in \mathbb{N} \) and let \( P \) be a set of \( n \) polygons on the plane with at most \( K \) vertices each such that each vertex has integer coordinates in \( \{0, ..., N\} \) and no two polygons are identical. In polynomial time we can compute a set of polygons \( P' \) with at most \( K \) vertices each with integer coordinates in \( \{0, ..., \text{poly}(K, N, n)\} \) and a bijection \( f : P \to P' \) such that for any \( P, P' \in P \) the polygons \( P \) and \( P' \) touch if and only if \( f(P) \) and \( f(P') \) touch.

Proof. First we show that for any point \( p \) with integral coordinates in \( \{0, ..., N\} \) and any line \( L \) going through two points with integral coordinates in \( \{0, ..., N\} \) it holds that either \( p \) lies on \( L \) or the distance between \( p \) and \( L \) is at least \( \epsilon := \frac{1}{2N} \).

Let \( L \) be a line going through two points \((x_1, y_1), (x_2, y_2)\) with \( x_1, x_2, y_1, y_2 \in \{0, ..., N\} \) and let \( p \) be a point with integer coordinates such that \( p \) does not lie on \( L \). If \( L \) is either vertical or horizontal, then the distance between \( L \) and \( p \) is at least one. Otherwise, consider an arbitrary point \( p' \in L \) such that at least one coordinate of \( p' \) is integral. Then the other coordinate of \( p' \) is either of the form \( x_1 \pm \epsilon \cdot \frac{|y_2-y_1|}{|x_2-x_1|} \) or \( y_1 \pm \epsilon \cdot \frac{|x_2-x_1|}{|y_2-y_1|} \) for \( h, h' \in \mathbb{N} \). As \( 1 \leq |y_2-y_1| \leq N \) and \( 1 \leq |x_2-x_1| \leq N \), this implies that the distance between \( p \) and \( L \) in the Manhattan metric is at least \( 1/N \). In particular, for the point \( \tilde{p} \in L \) that minimizes the distance between \( p \) and \( L \) in the Manhattan metric, the distance between \( \tilde{p} \) and \( p \) in the Manhattan metric is at least \( 1/N \) and thus their distance in the Euclidean metric is at least \( \frac{1}{2N} \).

As we will show, if we move every vertex \( p \) of every polygon \( P \in P \) by a distance of at most \( \epsilon' := \epsilon/5 \) towards the interior of the respective polygon \( P \), we do not change the combinatorial structure of \( P \) (see Figure 4). We can assume that all input vertices have even integer coordinates (otherwise, we could scale up all coordinates by a factor of two). Observe that this implies that the mid-point of every edge has integer coordinates.

As the polygons in \( P \) have at most \( K \cdot n \) vertices in total, there are at most \((K \cdot n)^2\) possible lines going through two of these vertices. For each of those lines \( L \) we want to ensure that no third vertex lies on \( L \). We achieve this as follows. We consider the vertices of the polygons one by one. Let \( p \) be a vertex of a polygon \( P \), and let \( e, e' \) be the edges of \( P \) adjacent to \( p \). We will move \( p \) towards the interior of \( P \). Let \( \vec{v} \) denote the vector representing \( e \), i.e., \( p + \vec{v} \) yields the coordinates of the other endpoint of \( e \). Similarly, denote by \( \vec{v}' \) the vector representing \( e' \). We move \( p \) to the point \( p + (\ell \cdot \vec{v} + \ell' \cdot \vec{v}')/N \cdot \frac{1}{(Kn)^2} \) for some integers \( \ell, \ell' \in \{0, ..., (Kn)^2 - 1\} \) if the angle of the polygon at \( p \) is smaller than \( \pi \), and to the point...
p - (ℓ · \vec{v} + ℓ' · \vec{v}') \frac{1}{4N(Kn)^2}$ if the angle at $p$ is greater than $\pi$ (see Figure 5). In total, $p$ moves by a distance of at most $\ell/5$ since $\|\vec{v}\| \leq 4N$ and $\|\vec{v}'\| \leq 4N$. We claim that we can find values for $\ell$ and $\ell'$ such that after moving $p$ the resulting point does not lie on a line $L$ that goes through two other vertices $p_1, p_2$ of polygons in $\mathcal{P}$.

There are $(Kn)^4$ possible positions $p'$ where a point $p$ can be moved (note that no two different pairs \{\ell_1, \ell_1', \ell_2, \ell_2'\} yield the same point since the vectors corresponding to the edges $e$ and $e'$ are linearly independent). Each pair $p_1, p_2$ of two vertices of polygons in $\mathcal{P}$ with $p_1 \neq p \neq p_2$ might forbid us to use up to $(Kn)^2$ of these candidates and there are at most $(Kn - 1) \cdot (Kn - 2) < (Kn)^2$ such pairs. Thus, by the pigeonhole principle, we can find a pair $\ell, \ell'$ such that $p$ does not lie on any line going through two vertices $p_1, p_2$ with $p_1 \neq p \neq p_2$. Performing this operation for every vertex $p$ of every polygon $P$ defines the map $f$ (see the statement of the lemma).

We claim that after this modification two polygons $f(P), f(P')$ touch if and only if $P$ and $P'$ touch. We moved all vertices of the polygons towards the interior of the respective polygon. Thus, $P \cap P' = \emptyset$ implies that $f(P) \cap f(P') = \emptyset$.

Now suppose that $P \cap P' \neq \emptyset$. We want to show that then $f(P) \cap f(P') \neq \emptyset$. We distinguish two cases. The first case is that there is an edge $e$ of $P$ and an edge $e'$ of $P'$ such that $e$ and $e'$ cross each other. Note that then the point $p := e \cap e'$ is not an endpoint of either of the two. By the above reasoning, the distance between $p$ and any of the four endpoints of $e$ and $e'$ is at least $c$ (as the distance between any vertex and any line going through two vertices of a polygon is either zero or at least $c$). Since we moved each point by a distance of at most $\ell' = c/5$, after moving the points $e$ and $e'$ still cross each other and thus $f(P) \cap f(P') \neq \emptyset$. Assume now that there are no such two edges. Then every vertex of the polygon $P'' := P \cap P'$ coincides with a vertex of $P$ or a vertex of $P'$, see Figure 6.

As $P \neq P'$, there must be two points $p, p'$ that are vertices of $P$ or $P'$ such that $L[p, p'] \in \partial P''$, the line segment $L[p, p']$ does not contain any vertex from $P$ or $P'$ apart from $p$ and $p'$, and $L[p, p']$ is not contained in any edge of $P$ or it is not contained in any edge of $P'$. Suppose w.l.o.g. that $L[p, p']$ is not contained in any edge of $P$. Then the midpoint $\tilde{p}$ of $L[p, p']$ has integral coordinates and is contained in $P$. This implies that $\tilde{p}$ has a distance of at least $c$ to $\partial P$ and note that $\tilde{p} \in \partial P'$. Therefore, after moving the polygons, there is a point on $\partial f(P')$ which is contained in $f(P)$. Thus, $f(P) \cap f(P') \neq \emptyset$.

Initially, all vertices have integer coordinates. Then, each vertex $p$ is moved to some point $p \pm (\ell \cdot \vec{v} + \ell' \cdot \vec{v}') \frac{1}{4N(Kn)^2}$ for integers $\ell, \ell' \in \{0, ..., (Kn)^2 - 1\}$. The resulting coordinates are of the form $\tilde{\ell} \cdot \frac{1}{h(K, N, n)}$ for an integer $\tilde{\ell} \in \{0, ..., N \cdot h(K, N, n)\}$ for a polynomial $h$ that is independent of the problem instance. By scaling all coordinates up linearly by a factor of $h(K, N, n)$, we obtain a set of polygons $\mathcal{P}'$ with integer coordinates in $\{0, ..., N \cdot h(K, N, n)\}$ that fulfill the claim of the lemma.

The above operation can be performed efficiently. When moving each vertex it suffices to identify the
Figure 6: If two polygons $P$ (solid boundary) and $P'$ (dashed boundary) touch, and there are no two edges $e$ and $e'$ of $P$ and $P'$, respectively, which cross each other, there must be a point $\tilde{p}$ on the boundary of one of the polygons (in this case $P'$), which lies in the interior of the other polygon ($P$) and which has integer coordinates. Hence, it has a distance of at least $c = \frac{1}{2N}$ to the boundary of the second polygon ($P$).

Correct values for $\ell$ and $\ell'$, and there is only a polynomial number of possibilities.

\[\Box\]

B Proof of Lemma 5

In order to prove Lemma 5 we show that there is a family of polygons with corners at DP-points which yield a recursion of GEO-DP with the claimed approximation factor. In fact, the following two lemmas were almost identically given in [1].

Note that, in the following, the polygons $\Gamma$ and $Q$ do not have to be simple, i.e., they can have holes.

Lemma 20. Let $\Gamma_1, \ldots, \Gamma_m$ be a collection of polygons, where each $\Gamma_i$ has at most $m\bar{\ell}$ edges. Then the intersection $\bigcap_{i=1, \ldots, m} \Gamma_i$ consists of at most $(m\bar{\ell})^2$ connected components, and each connected component is a polygon with at most $(m\bar{\ell})^2$ edges.

Proof. As the polygons $\Gamma_i$ have at most $m\bar{\ell}$ edges in total, and any two edges can cross at most once, we have at most $(m\bar{\ell})^2$ pairs of crossing edges. Each connected component of the intersection $\bigcap_{i=1, \ldots, m} \Gamma_i$ has at least three corners, and each corner corresponds to a different pair of crossing edges. Hence, there are at most $(m\bar{\ell})^2$ connected components. The number of edges of one connected component equals the number of vertices of $\Gamma$, which is also upper bounded by the number of crossing pairs $(m\bar{\ell})^2$. \[\Box\]

Lemma 21. Let $\alpha, \ell$ be as defined in Lemma 5 and let $\bar{P}$ be a set of at most $n$ pairwise non-touching polygons. Let $j^* = \left\lceil \log_{3/2} n^2 / \epsilon \right\rceil$. Then for each $j \in \{0, 1, \ldots, j^*\}$ there is a family of polygons $Q_j$, such that:

a) each polygon $Q \in Q_j$ has at most $(j+1)^2 \cdot (\ell + 4)^2$ edges and all its vertices are DP-points,

b) $Q_0 = \{[0, N] \times [0, N]\},$

c) the polygons in $Q_j$ are disjoint, and each polygon $Q \in Q_{j-1}$ is a disjoint union of at most $(j+1)^2(\ell + 4)^2$ polygons from $Q_{j-1},$

d) each polygon $Q \in Q_{j^*}$ contains at most one polygon from $\bar{P},$

e) for each set $Q_j$ we get $\sum_{Q \in Q_j} w(Q) \geq (1 - \alpha)^j \cdot w(\bar{P})$, where $w(Q) = \sum_{P \in \bar{P}, P \subseteq Q} w(P).$
Proof. We set \(Q_0 = \{[0, N] \times [0, N]\}\), i.e., \(Q_0\) consists of one square which contains all the polygons from \(\mathcal{P}\). We then construct the sets \(Q_1, \ldots, Q_j\), one by one, as follows. To construct \(Q_j\), we consider each polygon \(Q \in Q_{j-1}\), and add to \(Q_j\) the following set of polygons, which together give a disjoint union of \(Q\). If \(Q\) contains at most one polygon \(P \in \mathcal{P}\), we add \(Q\) to the set \(Q_j\). Otherwise, if there is a polygon \(P_0 \in \mathcal{P}, P_0 \subseteq Q\) with \(w(P_0) \geq 1/3 \cdot w(Q)\), we add to \(Q\) the following polygons: \(Q \cap P_0 = P_0\), and the connected components of \(Q \cap \bar{P}_0\), i.e. the connected components of \(Q \setminus P_0\). Finally, consider the case that no polygon \(P_0 \in \mathcal{P}\) with \(P_0 \subseteq Q\) has weight \(w(P_0) \geq 1/3 \cdot w(Q)\). Then there exists a balanced \(\alpha\)-cheap \(\ell\)-cut for the set of polygons from \(\mathcal{P}\) which are contained in \(Q\). Let \(\bar{\Gamma}\) be the polygon defining this cut, and let \(\bar{\Gamma}\) be its complement intersected with the input square. Due to the assumptions in Lemma 5 all corners of \(\Gamma\) are basic DP-points. We add to \(Q_j\) all connected components of \(Q \cap \Gamma\) and \(Q \cap \bar{\Gamma}\). Notice that \(\Gamma\) has at most \(\ell\) edges, and so \(\bar{\Gamma}\) has at most \(\ell + 4\) edges.

We now have to check that all required properties are satisfied.

a) The only polygon in \(Q_0\) has 4 edges. Each polygon \(Q \in Q_j\) for \(j \geq 1\), is a connected component of an intersection of at most \(j + 1\) polygons with at most \(\ell + 4\) edges each whose corners are basic DP-points. Hence, the vertices of \(Q\) are all DP-points and, due to Lemma 20 \(Q\) has at most \((j + 1)^2(\ell + 4)^2\) edges.

b) Defined at the beginning of the proof.

c) From the construction of the sets \(Q_j\) it can be easily observed that the polygons in \(Q_j\) are disjoint, and each polygon \(Q \in Q_{j-1}\) is a union of polygons from \(Q_j\). We now have to upper bound the number of polygons from \(Q_j\) which can be contained in one polygon \(Q \in Q_{j-1}\). Polygon \(Q\) is a connected component of an intersection of at most \(j\) polygons, each with at most \(\ell + 4\) edges. By construction each polygon \(Q' \in Q_j\) is contained in some polygon \(Q \in Q_{j-1}\). Each polygon \(Q' \in Q_j\) contained in \(Q\) is a connected component of an intersection \(Q\) with a polygon \(\Gamma\) with at most \(\ell\) edges, or with a polygon \(\bar{\Gamma}\) with at most \(\ell + 4\) edges. Therefore \(Q'\) is a connected component of an intersection of at most \(j + 1\) polygons, each with at most \(\ell + 4\) edges. From Lemma 20 the number of such components is upper bounded by \((j + 1)^2(\ell + 4)^2\).

d) For a polygon \(Q\) let \(w(Q) := \sum_{P \in \mathcal{P}, P \subseteq Q} w(P)\) denote the weight of all polygons from \(\mathcal{P}\) contained in \(Q\). We will show by induction that if \(Q \in Q_j\) contains more than one polygon from \(\mathcal{P}\), then \(w(Q) \leq \frac{2j^2}{3}w(\mathcal{P}) \leq \frac{2j^2}{3}n^2/\epsilon\). For \(Q \in Q_j\) that value is at most 1 since \(\frac{2j^2}{3}n^2/\epsilon \leq 1\). As each polygon from \(\mathcal{P}\) has weight at least 1, \(Q\) cannot contain more than one polygon.

For the (unique) polygon \(Q \in Q_0\) we have \(w(Q) = w(\mathcal{P}) \leq n^2/\epsilon\) (see Lemma 1). We assume by induction that the property holds for \(Q_{j-1}\), and we will show that it holds also for \(Q_j\). Let \(Q \in Q_j\) be contained in a polygon \(Q_0 \in Q_{j-1}\), where \(w(Q_0) \leq \frac{2j-1}{3}w(\mathcal{P}) \leq \frac{2j-1}{3}n^2/\epsilon\). If \(Q\) contains more than one polygon from \(\mathcal{P}\), then either \(Q \subseteq Q_0 \setminus P^*\) for a polygon \(P^*\) with \(w(P^*) \geq \frac{4}{3} \cdot w(Q_0)\), or \(Q\) is obtained from \(Q_0\) by a balanced cut. In both cases we get \(w(Q) \leq \frac{2}{3}w(Q_0) \leq \frac{2j}{3}w(\mathcal{P})\).

e) The property holds for \(Q_0\), as \(\sum_{Q \in Q_0} w(Q) = w(\mathcal{P})\). We give a proof by induction. Assume that the property holds for \(Q_{j-1}\), i.e., \(\sum_{Q \in Q_{j-1}} w(Q) \geq (1 - \alpha)^{j-1} \cdot w(\mathcal{P})\). The polygons which are intersected by \(Q_{j-1}\), but not by \(Q_{j-1}\), must be intersected by the newly introduced cuts \(\Gamma\), which intersect polygons \(Q \in Q_{j-1}\). As each cut \(\Gamma\) is a \(\alpha\)-cheap \(\ell\)-cut for the set of polygons contained in the corresponding polygon \(Q\), we get \(\sum_{Q' \in Q_{j-1}, Q' \subseteq Q} w(Q') \geq (1 - \alpha)w(Q)\) for each \(Q \in Q_{j-1}\), and so \(\sum_{Q \in Q_{j-1}} w(Q) \geq (1 - \alpha)^2 \cdot w(\mathcal{P})\). □

With this preparation we are able to prove Lemma 5.

Proof of Lemma 5 We run GEO-DP, parametrized by \(k := \left(\left\lceil \log_{3/2}(n^2/\epsilon) \right\rceil + 1 \right)^2 \cdot (\ell + 4)^2\), on \(\mathcal{P}\). Denote by \(Q_j\), with \(j \in \{0, 1, \ldots, j^*\}\) for \(j^* = \left\lceil \log_{3/2}(n^2/\epsilon) \right\rceil\), the families of polygons with DP-points as coordinates as given by Lemma 21.

From Lemma 21c any polygon \(Q \in Q_j\) has at most \(k\) edges, all its vertices are DP-points, and so GEO-DP has a DP-cell for \(Q\). If \(Q \in Q_{j^*}\), from Lemma 21d we know that \(Q\) contains at most one polygon
from \( \bar{P} \), and so \( w(sol(Q)) = w(Q) \), where for each polygon \( Q \) we denote by \( w(Q) \) the total weight of all polygons from \( P \) which are contained in \( Q \). From Lemma 21(c), each polygon \( Q \in Q_j \) is a union of at most \( k \) polygons \( Q_1, \ldots, Q_m \in Q_{j+1} \). Therefore GEO-DP tries the subdivision of \( Q \) into these components and we get that \( w(sol(Q)) \geq \sum_{i=1}^{m} w(sol(Q_i)) \), which for the input polygon \( Q_0 \in Q_0 \) (see Lemma 21(b)) gives

\[
\sum_{Q \in Q_j^*} w(sol(Q)) \geq \sum_{Q \in Q_j^*} w(sol(Q)) \geq (1 - \alpha)^j \cdot w(\bar{P}),
\]

where the last inequality comes from Lemma 21(c). The approximation ratio of GEO-DP is then \((1 - \alpha)^{-j} = (1 - \alpha)^{O(\log(n/\epsilon))} = 1 + O(\epsilon)\).

\[\square\]

\( C \) Other Omitted Proofs and Figures

**Proof of Lemma 4** Suppose we are given a set of polygons \( P \) with arbitrary weights. First, we scale the weights of all polygons such that \( \max_{P \in P} w(P) = n/\epsilon \). Since then \( OPT \geq n/\epsilon \), all polygons \( P' \in P \) with \( w(P') < 1 \) can contribute a total weight of at most \( n \cdot 1 = \epsilon \cdot \frac{n}{\epsilon} \leq \epsilon \cdot OPT \). We remove them from the instance which reduces the value of the optimal solution by at most \( \epsilon \cdot OPT \). Thus, the value of the optimal solution decreases at most by a factor of \((1 - \epsilon)^{-1} = 1 + O(\epsilon)\).

**Proof of Lemma 3** As the lines in \( E_i \) do not cross one another, we can order them from top to bottom. We construct the set of lines \( \tilde{E}_i \) one by one. If \( \tilde{E}_i = \emptyset \) or if the total weight of triangles touching \( S_i \) is at most \( \delta^4 \cdot w(T) \), we set \( \tilde{E}_i = \emptyset \) and we are done.

Let \( e_1 \in E_i \) be the bottom-most line from \( E_i \) such that the total weight of triangles touching \( S_i \) above \( e_1 \) is at most \( \delta^4 \cdot w(T) \). If such a line does not exist, we set \( e_1 \in E_i \) to be the top-most line crossing \( S_i \). Iteratively, we define \( e_j \in E_i \) be the bottom-most line from \( E_i \) which is below \( e_{j-1} \) such that the total weight of triangles touching \( S_i \) between \( e_{j-1} \) and \( e_j \) is at most \( \delta^4 \cdot w(T) \). Again, if such a line does not exist, we set \( e_j \in E_i \) to be the top-most line crossing \( S_i \) below \( e_{j-1} \). We stop when we have added to \( \tilde{E}_i \) a line \( e_{j^*} \in E_i \) such that the total weight of triangles touching \( S_i \) below \( e_{j^*} \) is at most \( \delta^4 \cdot w(T) \), or when \( e_{j^*} \) is the bottom-most line in \( E_i \).

Let \( C \) be a connected component of \( S_i \cup \tilde{E}_i \) such that \( C \cap \bigcup E_i \neq \emptyset \). Then from the construction above the total weight of triangles touching \( C \) is at most \( \delta^4 \cdot w(T) \).

Let \( e_j, e_{j+1}, e_{j+2} \) be three consecutive lines from \( \tilde{E}_i \). From the construction above we know that the total weight of triangles touching \( S_i \) between \( e_j \) and \( e_{j+2} \) is greater than \( \delta^4 \cdot w(T) \). As for each triangle \( T \in T \) the intersection \( T \cap S_i \) is contained between two consecutive lines from \( \tilde{E}_i \) (or possibly above the first or below the last line from \( \tilde{E}_i \)), we get that \( |E_i| \leq \frac{2}{\delta^4} \).

**Proof of Proposition 7** For each stripe \( S_i \) we have \( |E_i| \leq O(1/\delta^4) \). Thus, the number of cells in a stripe \( S_i \) is at most \( O(1/\delta^4) \). Since we have \( O(1/\delta^2) \) stripes the total number of cells is upper bound by \( O(1/\delta^6) \). For each dense cell we added two lines to \( L_{\text{ext}} \), so \( O(1/\delta^6) \) in total. When connecting each of the \( O(1/\delta^5) \) lines in \( L_0 \) (cf., Proposition 9) to the boundary of \( I \) we added at most \( O(1/\delta^2) \) lines per each line in \( L_0 \). Thus, \( |L_0 \cup L_{\text{ext}}| = O(1/\delta^6) \).

Each endpoint of a line in \( L_{\text{ext}} \) created in the first step, i.e., while adding vertical boundary lines for dense cells, is a corner of a cell. Each endpoint of a line in \( L_{\text{ext}} \) created in the second step, i.e., while connecting lines from \( L_0 \) with other lines, is either a vertex of an input triangle (this applies to the lines \( L_1, L_2, \ldots \)), or an intersection of an edge of a triangle with a vertical line \( \{x_1\} \times [0, N] \) where the latter contains some vertex of an input triangle (this applies to the lines \( L_0', L_1', \ldots \)). By the definition of the DP-points each such point is a basic DP-point.
Figure 7: Two cells $C_1^i, C_2^i$ of the same stripe $S_i$ cannot lie in the same face of the partition.

**Proof of Lemma 13.** Suppose that $F$ touches some dense cell $C \subseteq S_i$. Due to the definition of the lines in $L$ then either $F = T$ for some triangle $T$ (when $T$ was cut out since it owned some cell, not necessarily $C$) or $F \subseteq C$. If $F = T$, we are done. Now assume that $F \subseteq C$. From the construction of the lines $L_0 \cup L_{\text{ext}}$ and the definition of a dense cell, $F$ is contained in the stripe $S_i$, $F$ is not contained in a single triangle $T \in \mathcal{T}$, and $F$ is not crossed by any line in $\mathcal{E}_i$. The two latter properties give us that each triangle touching $F$ has at least one endpoint inside $F \subseteq S_i$. As $\bar{w}(S_i) < \delta^2 w(\mathcal{T})$, the total weight of triangles touching $F$ is upper bounded by $3\delta^2 \cdot w(T)$.

Now assume that $F$ does not touch any dense cells and there is no triangle $T$ such that $F = T$. We will show that for every stripe $S_i$ the intersection $F \cap S_i$, if non-empty, is contained in one cell $C_i$. Let $C_1^i$ and $C_2^i$ be two cells of $S_i$. Let $L \in L_0$ be a line crossing $S_i$ and separating $C_1^i$ from $C_2^i$ (see Figure 7). From the construction of the lines in $L_0 \cup L_{\text{ext}}$, there is a path of lines in $L_0 \cup L_{\text{ext}}$ (and therefore also a path of lines in $L$) containing $L$ and connecting the left or bottom boundary of $I$ with the top or right boundary of $I$. The cells $C_1^i$ and $C_2^i$ are separated by this path. Therefore any face $F$ touching $C_1^i$ does not touch $C_2^i$.

As the number of stripes is not greater than $1/\delta^2$ and from Lemma 8 for each light cell $C_i$ the total weight of triangles touching $C_i$ is upper bounded by $\delta^4 \cdot w(T)$, the total weight of triangles touching $F$ is upper bounded by $\delta^2 \cdot w(T)$.

**Proof of Proposition 15.** Due to Lemma 14 each triangle $T \in \mathcal{T}$ is touched by at most four lines in $L$. Thus, each triangle $T \in \mathcal{T}$ is touched by at most four edges of $G$ and hence the total cost of the edges $E$ is bounded by $O(w(\mathcal{T}))$. The other claims follow directly from Lemmas 13 and 14 and Proposition 12, using that $\delta < 1/3$.

**Proof of Lemma 17.** We apply Theorem 16 to the graph $G = (V, E)$ obtained from the subdivision $L$, with parameter $k := \frac{1}{\delta \cdot w(T)} \sum_{e_{p,p'} \in E} c_{p,p'}$. From Proposition 15 we know that $k = O(1/\delta)$, and that the obtained $V$-cycle $C$ uses at most $O(1/\delta)$ face edges and ordinary edges with total cost at most $O(\delta \cdot w(\mathcal{T}))$. We transform $C$ into a $V$-cycle $C'$ using only ordinary edges of $G$, by replacing each face edge connecting two vertices $v_p$ and $v_p'$ with a path $R(v_p, v_p')$ lying on the boundary of the crossed face. As due to Proposition 15 each face has weight at most $w(\mathcal{T})/3$, while performing the above operation we ensure that the modified
cut $C'$ remains balanced (i.e., each side of the cut contains triangles of total weight at most $2/3 \cdot w(T)$). We do that by choosing the path $R(v_p, v'_p)$ in one of the two possible ways, depending on whether we want to connect the face with the interior, or with the exterior of $C'$. Notice that after this operation $C'$ might not be a simple cycle. However, we can modify $C'$ so that each edge appears only $O(1)$ times in $C'$. As from Proposition 15 the number of edges in $G$ is bounded by $(\frac{1}{\delta})^{O(1)}$, $C'$ uses at most $(\frac{1}{\delta})^{O(1)}$ edges. All vertices of $C'$ are vertices of $G$, and so they are basic DP-points.

It remains to show is that $C'$ is $O(\delta)$-cheap. The ordinary edges of $C$ have cost $O(\delta \cdot w(T))$. The remaining edges of $C'$ lie on the boundary of $O(1/\delta)$ faces of $G$, and from Proposition 15 they intersect triangles of total weight $O(\delta \cdot w(T))$. Thus, the cycle $C'$ is $O(\delta)$-cheap.

Proof of Theorem 18 From Lemma 17 when setting $\delta := \Theta(\frac{\epsilon}{K \log(n/\epsilon)})$, we obtain that for any set $T$ of pairwise non-touching triangles in the plane such that $w(T) < w(T)/3$ for each $T \in T$ there exists a balanced $\frac{\epsilon}{K \log(n/\epsilon)}$-cheap $(\frac{\epsilon}{K \log(n/\epsilon)})^{O(1)}$-cut with corners at basic DP-points. Applying Lemma 6 gives that for any set of pairwise non-touching polygons $P$ with at most $K$ edges each such that $\max_{P \in P} w(P) < w(P)/3$ there is a balanced $\frac{\epsilon}{\log(n/\epsilon)}$-cheap $K \cdot (\frac{\epsilon}{\log(n/\epsilon)})^{O(1)}$-cut with vertices at basic DP-points. From Lemma 5 GEO-DP parametrized by $k = (\frac{K \log(n/\epsilon)}{\epsilon})^{O(1)}$ has an approximation ratio of $1+O(\epsilon)$. According to Proposition 3, the running time of GEO-DP is upper bounded by $(nK)^{O(k^2)} = (nK)^{(\frac{1}{2} K \log(n/\epsilon))^{O(1)}} \leq (nK)^{(\frac{1}{2} K \log(n))^{O(1)}}$. If $K \leq (\log n)^{O(1)}$ this yields a running time of $n^{(\frac{1}{2} \log n)^{O(1)}}$ which is quasipolynomial.