A QPTAS for the Base of the Number of Triangulations of a Planar Point Set

Marek Karpinski *[1], Andrzej Lingas †[2], and Dzmitry Sledneu[3]

1 **Department of Computer Science, University of Bonn** `marek@cs.uni-bonn.de`
2 **Department of Computer Science, Lund University** `Andrzej.Lingas@cs.lth.se`
3 **The Centre for Mathematical Sciences, Lund University** `Dzmitry@maths.lth.se`

Abstract

The number of triangulations of a planar n point set is known to be c^n, where the base c lies between 2.43 and 30. The fastest known algorithm for counting triangulations of a planar n point set runs in $O^*(2^n)$ time. The fastest known arbitrarily close approximation algorithm for the base of the number of triangulations of a planar n point set runs in time subexponential in n. We present the first quasi-polynomial approximation scheme for the base of the number of triangulations of a planar point set.

1 Introduction

A *triangulation* T of a set S of n points in the Euclidean plane is a maximal set of properly non-intersecting straight-line segments with both endpoints in S. These straight-line segments are called *edges* of T. Let $F(S)$ stand for the set of all triangulations of S.

The problem of computing the number of triangulations of S, i.e., $|F(S)|$, is easy when S is convex. Simply, by a straightforward recurrence, $|F(S)| = C_{n-2}$, where C_k is the k-th Catalan number, in this special case. However, in the general case, the problem of computing the number of triangulations of S is neither known to be $\#P$-hard nor known to admit a polynomial-time counting algorithm.

It is known that $|F(S)|$ lies between $\Omega(2.43^n)$ and $O(30^n)$. Since the so called flip graph whose nodes are triangulations of S is connected [7], all triangulations of S can be listed in exponential time by a standard traversal of this graph. Only recently, Alvarez and Seidel have presented an elegant algorithm for the number of triangulations of S running in $O^*(2^n)$ time [4] which is substantially below the aforementioned lower bound on $|F(S)|$ (the O^* notation suppresses polynomial in n factors).

Also recently, Alvarez, Bringmann, Ray, and Seidel [3] have presented an approximation algorithm for the number of triangulations of S based on a recursive application of the planar simple cycle separator [6]. Their algorithm runs in subexponential $2^{O(\sqrt{n}\log n)}$ time and over-counts the number of triangulations by at most a subexponential $2^{O(n^{\frac{3}{4}}\sqrt{\log n})}$ factor. It also yields a subexponential-time approximation scheme for the base of the number of triangulations of S, i.e., for $|F(S)|^{\frac{1}{n}}$. The authors of [3] observe also that just the inequalities $\Omega(2.43^n) \le |F(S)| \le O(30^n)$ yield the large exponential approximation factor $O(\sqrt{30 \times 2.43}^n)$ for $|F(S)|$ trivially computable in polynomial time.

1.1 Our contribution

We take a similar approximation approach to the problem of counting triangulations of S as Alvarez, Bringmann, Ray, and Seidel in [3]. However, importantly, instead of using

* Research partially supported by DFG grants and the Hausdorff Center grant.
† Research supported in part by VR grant 621-2011-6179.

recursively the planar simple cycle separator [6], we apply recursively the so called balanced α-cheap l cuts of maximum independent sets of triangles within a dynamic programming framework developed by Adamaszek and Wiese in [1, 2]. By using the aforementioned techniques, the authors of [1] designed the first quasi-polynomial time approximation scheme (QPTAS) for the maximum weight independent set of polygons belonging to the input set set of polygons with poly-logarithmically many edges.

Observe that a triangulation of S can be viewed as a maximum independent set of triangles drawn from the set of all triangles with corners in S that are free from other points in S (triangles, or in general polygons, are identified with their open inferiors). This simple observation enables us to use the aforementioned balanced α-cheap l cuts recursively in order to bound an approximation factor of our approximation algorithm. The parameter α specifies the maximum fraction of an independent set of triangles that can be destroyed by the l-cut, which is a polygon with at most l corners in a specially constructed set of points of polynomial size.

Similarly as the approximation algorithm from [3], our algorithm may over-count the true number of triangulations because the same triangulation can be be partitioned recursively in many different ways. In contrast with the approximation algorithm in [3], our algorithm may also under-count the number of triangulations of S, since our partitions generally destroy a fraction of triangles in a complete triangulation of S.

Our approximation algorithm for the number of triangulations of a set S of n points with integer coordinates in the plane runs in $n^{(\log n/\epsilon)^{O(1)}}$ time. For $\epsilon > 0$, it returns a number at most $2^{\epsilon n}$ times smaller and at most $2^{\epsilon n}$ times larger than the number of triangulations of S. Note that even for $\epsilon = (\log n)^{-O(1)}$, the running time is still quasi-polynomial.

As a corollary, we obtain a quasi-polynomial approximation scheme for the base of the number of triangulations of S, i.e., for $|F(S)|^{\frac{1}{n}}$. This implies that the problem of approximating $|F(S)|^{\frac{1}{n}}$ cannot be APX-hard (under standard complexity theoretical assumptions).

1.2 Organization of the paper

In Preliminaries, we introduce basic concepts of the dynamic programming framework from [2]. Section 3 presents our approximation counting algorithm for the number of triangulations of S. In Section 4, the time complexity of the algorithm is examined while in Sections 5, upper bounds on the under-counting and the over-counting of the algorithm are derived, respectively. In Section 6, our main results are formulated. We conclude with a short discussion on possible extensions of our results in Section 7.

2 Preliminaries

The Maximum Weight Independent Set of Polygons Problem (MWISP) is defined as follows [1]. We are given a set Q of n polygons in the Euclidean plane. Each polygon has at most k vertices, each of the vertices has integer coordinates. Next, each polygon P in Q is considered as an open set, i.e., it is identified with the set of points forming its interior. Also, each polygon $P \in Q$ has weight $w(P) > 0$ associated with it. The task is to find a maximum weight independent set of polygons in Q, i.e., a maximum weight set $Q' \subseteq Q$ such that for all pairs $P_i,\ P_j$ of polygons in Q', if $P_i \neq P_j$ then it holds $P_i \cap P_j = \emptyset$.

The *bounding box* of Q is the smallest rectilinear square containing all polygons in Q.

Note that in particular if Q consists of all triangles with corners in a finite planar point set S such that no other point in S lies inside them or on their perimeter, each having weight 1, then the set of all maximum independent sets of polygons in Q is just the set of all triangulations of S. We shall denote the latter set by $F(S)$.

Adamaszek and Wiese have shown that if $k = poly(\log n)$ then MWISP admits a QPTAS [1].

Fact 1[1]. *Let k be a positive integer. There exists a $(1+\epsilon)$-approximation algorithm with a running time of $(nk)^{(\frac{k}{\epsilon}\log n)^{O(1)}}$ for the Maximum Weight Independent Set of Polygons Problem provided that each polygon has at most k vertices.*

Recently, Har-Peled generalized Fact 1 to include arbitrary polygons [5].

▶ **Definition 1.** Let $l \in N$ and $\alpha \in R$ where $0 < \alpha < 1$. Let T be a set of pairwise non-touching triangles. A polygon Γ is a balanced α-cheap l-cut of T if

- Γ has at most l edges,
- the total weight of all triangles in T that intersect Γ does not exceed the α fraction of the total weight of triangles in T,
- the total weight of the triangles in T contained in Γ does not exceeds two thirds of the total weight of triangles in T,
- the total weight of the triangles in T outside Γ does not exceeds two thirds of the total weight of triangles in T.

For a set of triangles T in the plane, the set of *DP-points* consists of *basic DP-points* and *additional DP-points*. The set of basic DP-points contains the four corners of the bounding box of T and each intersection of a vertical line passing through a corner of a triangle in T with any edge of a triangle in T or a horizontal edge of the bounding box. The set of additional DP-points consists of all intersections of pairs of straight-line segments whose endpoints are basic DP-points. The authors of [1] observe that the number of all DP-points is $O((3n)^4)$.

Fact 2[1]. *Let $\delta > 0$ and let T be a set of pairwise non-touching triangles in the plane such that the weight of none triangle in T exceeds one third of the weight of T. Then there exists a balanced $O(\delta)$-cheap $(\frac{1}{\delta})^{O(1)}$-cut with corners at basic DP-points.*

3 Dynamic programming

The QPTAS of Adamaszek and Wiese for maximum weight independent set of polygons [1] is based on dynamic programming. For each polygon with at most k edges and vertices at the DP points induced by the input polygons (termed a DP cell), an approximate maximum weight independent subset of the input polygons contained in the DP cell is computed. The computation is done by considering all possible partitions of the DP cell into at most k smaller DP cells. For each such partition, the union of the approximate solutions for the component DP cells is computed. Then, a maximum weight union is picked as the approximate solution for the DP cell.

Our dynamic programming algorithm, termed Algorithm 1 and depicted in Fig. 1., is in part similar to that of Adamaszek and Wiese [1]. The set of input polygons consists of all triangles with three corners in the input planar point set S that do not contain any other point in S. For each DP cell, an approximate number of triangulations within the cell is computed instead of an approximate maximum number of non-touching triangles within the cell. Further modification of the dynamic programming of Adamaszek and Wiese are as follows.

Input: A set S of n points with integer coordinates in the Euclidean plane and natural number parameters k and Δ.
Output: An approximate number of triangulations of S.

$T \leftarrow$ the set of all triangles with corners in S that do not contain any other point in S;
$P \leftarrow$ a list of polygons (possibly with holes) with at most k corners in total at DP points induced by T, topologically sorted with respect to geometric containment;
for each polygon set $Q \in P$ containing at most Δ points in S **do**
  $tr(Q) \leftarrow$ exact number of maximal triangulations of points in $S \cap P$ within P;
end for
for each polygon set $Q \in P$ containing more than Δ points in S **do**
  $tr(Q) \leftarrow 0$;
  for each partition of Q into polygons $Q_1, ..., Q_l \in P$, where $l \leq k$, no Q_j contains more than two thirds of points in $S \cap Q$, and $tr(Q_1)$ through $tr(Q_l)$ are defined **do**
    $tr(Q) \leftarrow tr(Q) + \prod_{j=1}^{l} tr(Q_j)$;
  end for
end for
Output $tr(B)$, where B is the bounding box of T.

Figure 1 Algorithm 1 for approximately counting triangulations of a finite planar point set.

1. Solely those partitions of a DP cell into at most k component DP cells are considered where no component cell contains more than two thirds of the input points (i.e., the corners of the input triangles) in the partitioned cell.
2. While a partition of a DP cell into at most k cells is processed, instead of the union of the solutions to the subproblems for these cells, the product of the numerical solutions for the component DP cells is computed.
3. Instead of taking the maximum of the solutions induced by the partition of a DP cell into at most k DP-cells, the sum of the numerical solutions induced by these partitions is computed.
4. When the number of points contained in a DP cell does not exceed the threshold number Δ then the exact number of maximal triangulations within the cell is computed.

Algorithm 1 also in part resembles the approximation counting algorithm for the number of triangulations of a planar point set due to Alvarez, Bringmann, Ray, and Seidel [3]. The main difference is in the used implicit recursive partition tool. Algorithm 1 uses balanced α-cheap l-cuts within the dynamic programming framework from [2] instead of the simple cycle planar separator theorem [3, 6]. Thus, Algorithm 1 recursively partitions a DP cell defining a subproblem into at most k smaller DP-cells while the algorithm in [3] recursively splits a subproblem by a simple cycle that yields a balanced partition.

4 Time complexity

The cardinality of T does not exceed n^3. Then, by the analogy with the dynamic programming algorithm of Adamaszek and Wiese for nearly maximum independent set of triangles, the number of DP cells is $(3n^3)^{O(k)} = n^{O(k)}$ (see Proposition 2.1 in [1]). Consequently, the number of possible partitions of a DP cell into at most k DP cells is $O(\binom{n^{O(k)}}{k})$ which is $n^{O(k^2)}$.

It follows that if we neglect the cost of computing the exact number of triangulations contained with a DP cell including at most Δ input points, then Algorithm 1 runs in $n^{O(k^2)}$

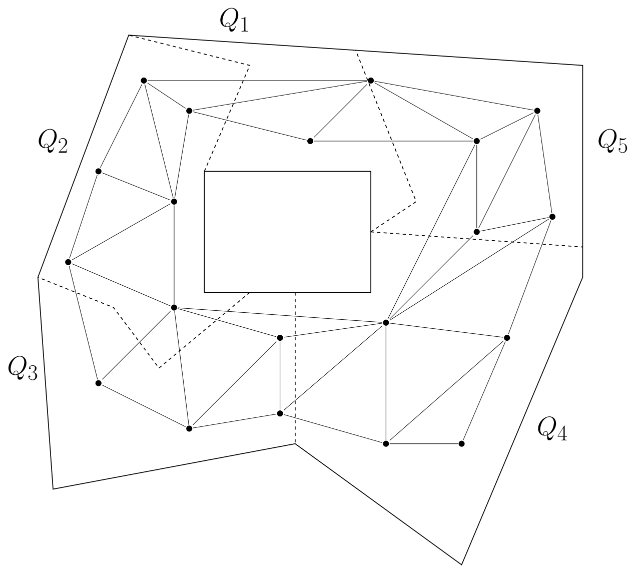

Figure 2 An example of a maximal triangulation within a DP cell and a partition of the DP cell into smaller DP cells Q,...,Q_5 crossing some triangles in the triangulation.

time.

We can compute the exact number of maximal triangulations contained within a DP cell with at most Δ input points by listing all complete triangulations of these points and counting only their maximal fragments lying within the cell. To list the complete triangulations of the input points within a DP cell we can apply any traversal algorithm, like BFS or DFS, to the so called *flip* graph of the triangulations which is known to be connected [7]. Hence, the listing takes time proportional to the number of the triangulations, which is at most 30^{Δ} by [8]. It follows that extracting the maximal fragments and counting them can be done also in $2^{O(\Delta)}n^{O(1)}$ time.

We conclude with the following lemma.

▶ **Lemma 2.** *Algorithm 1 runs in $n^{O(k^2)}2^{O(\Delta)}$ time.*

5 Approximation factor

5.1 Under-counting

The potential under-counting stems from the fact that when a DP cell is partitioned into at most k smaller DP cells then the possible combinations of triangulation edges crossing the partitioning edges are not counted. Furthermore, in the leaf DP cells, i.e., those including at most Δ points from S, we count only maximal triangulations while the restriction of a triangulation of S to a DP cell does not have to be maximal. See Fig. 2.

Intuitively, the general idea of the proof of our upper bound on under-counting is as follows. For each triangulation $W \in F(S)$, there is triangulation counted by Algorithm 1 that can be obtained by removal $O(\epsilon n)$ edges and augmenting with $O(\epsilon n)$ other edges. Such a triangulation is a union of maximal triangulations contained in leaf DP cells.

▶ **Lemma 3.** *Let S be a set of n points in the plane and let $\epsilon > 0$. For each $W \in F(S)$, there is a partial triangulation $W^* \subseteq W$ of S containing at least a $1 - O(\epsilon)$ fraction of the triangular faces of W and a partial triangulation $M(W^*)$ of S which is an extension of W^* by $O(\epsilon n)$*

edges such that the estimation returned by Algorithm 1 with k set to $\log^{O(1)}(n/\epsilon)/\epsilon^{O(1)}$ is not less than $|\bigcup_{W\in F(S)}\{M(W^*)\}|$.

Proof. Let $W \in F(S)$. By adapting the idea of the proof of the approximation ratio of the QPTAS in [1], consider the following tree U of DP cells obtained by recursive applications of balanced α-cheap l-cuts.

At the root of U, there is the bounding box. By Fact 2, there is a balanced α-cheap l-cut, where $l = \alpha^{-O(1)}$, that splits the box into at most k children DP cells such that only α fraction of the triangular faces of W is crossed by the cut. The construction of U proceeds recursively in children DP cells and stops in DP cells that contain at most Δ points.

Note that the that height of U is not greater than $\log_{3/2} n$.

For a node u of U, let W_u be the maximal partial triangulation of the points in the DP cell Q_u associated with u that is a restriction of W (i.e., its set of edges is a subset of that of W) and it is contained in Q_u. Next, let W_u^* denote the restriction of W_u to the union of W_t över the the leaves t of the subtree of U rooted at u.

By induction on the height $h(u)$ of u in U, we obtain that the partial triangulation $W_u^* \subseteq W_u$ contains $(1-\alpha)^{h(u)}$ fraction of triangular faces of W_u. Set α to $\frac{O(\epsilon)}{\log(n/\epsilon)}$. It follows in particular that for the root r of U, $W_r^* \subseteq W$ contains at least an $(1-\alpha)^{\log_{3/2} n/\epsilon} \geq 1-O(\epsilon)$ fraction of triangular faces in W. Set W^* to W_r^*.

For a leaf t of U, let $M(W_t)$ be an extension of W_t to a maximal triangulation within the leaf cell D_t. For a node u of U, let $M(W_u^*)$ be a partial triangulation within Q_u that is the union of $M(W_t)$ over the leaves t of the subtree of U rooted at u. We have also $M(W^*) = M(W_r^*)$ by $W^* = W_r^*$.

By the definition of $M(W^*)$, $M(W^*)$ is an extension of W^*. By the definition of W^*, any edge of $M(W^*)$ that is not an edge of W^* has both endpoints at corners of triangles in W that are missing in W^*. It follows that the number of edges in $M(W^*) \setminus W^*$ is at most $3 \times O(\epsilon n) = O(\epsilon n)$.

We shall show by induction on $h(u)$ that Algorithm 1 while computing an estimation for Q_u approximately counts the number of $M(W_u^*)$.

If $h(u) = 0$, i.e., u is a leaf in U then $W_u^* = W_u$ and consequently in particular $M(W_u^*) = M(W_u)$ is counted by Algorithm 1.

Suppose in turn that u is an internal node in U with l children $u_1, ..., u_l$. When the estimation for Q_u is computed by Algorithm 1, the sum of products of estimations yielded by different partitions of Q_u into at most k DP cells is computed. In particular, the partition into $Q_{u_1}, ..., Q_{u_l}$ is considered. By the induction hypothesis, the estimation for Q_{u_j} includes $M(W_{u_j}^*)$ for $j = 1, ..., l$. Hence, the product of these estimations counts also $M(W_u^*) = \bigcup_{j=1}^{l} M(W_{u_j}^*)$.

By $M(W^*) = M(W_r^*)$, to obtain the lemma it remains to show that the bound $\log^{O(1)}(n/\epsilon)/\epsilon^{O(1)}$ on k is sufficiently large. Following the proof of Lemma 2.1 in [1], observe that each DP cell Q_u at each level of U is an intersection of at most $O(\log(n/\epsilon))$ polygons, each with at most l edges and corners at basic DP points. Hence, by $\alpha = \frac{O(\epsilon)}{\log(n/\epsilon)}$ and $l = \alpha^{-O(1)}$, the resulting polygons have at most $O(l^2 \log^2(n/\epsilon)) = \log^{O(1)}(n/\epsilon)/\epsilon^{O(1)}$ edges and corners at basic and additional DP points. ◀

▶ **Theorem 4.** *The under-counting factor of Algorithm 1 with k set to $\log^{O(1)}(n/\epsilon)/\epsilon^{O(1)}$ is at most $2^{O(\epsilon n \log n)}$.*

Proof. Consider any triangulation $W \in F(S)$. By Lemma 3, the partial triangulation $W^* \subseteq W$ contains at least an $1 - O(\epsilon)$ fraction of the triangular faces of W. Hence, the edges completing W^* to any triangulation in $F(S)$ can be incident only to the corners of the triangular faces in W that do not occur in W^*. The number of the latter is at most $3 \times O(\epsilon n) = O(\epsilon n)$. It follows that the number of ways of completing W^* to a full triangulation in $F(S)$ is not greater than the number of full triangulations on $O(\epsilon n)$ vertices which is not greater than $30^{O(\epsilon n)} = 2^{O(\epsilon n)}$.

By Lemma 3, the estimation returned by Algorithm 1 with k set to $\log^{O(1)}(n/\epsilon)/\epsilon^{O(1)}$ is not less than $|\bigcup_{W \in F(S)}\{M(W^*)\}|$.

Now it remains to show that the maximum number of partial triangulations $(W')^*$, $W' \in F(S)$, for which $M((W')^*) = M(W^*)$ is at most $2^{O(\epsilon n \log n)}$. To see this observe that the edges extending $(W')^*$ to $M((W')^*)$ are incident to at most $O(\epsilon n)$ corners of the $O(\epsilon n)$ triangular faces of W' that are missing in $(W')^*$. Consequently, the maximum number of such partial triangulations $(W')^*$ is upper bounded by the number of subsets of at most $O(\epsilon n)$ edges of $M(W^*)$ (whose removal may form a partial triangulation $(W')^*$ satisfying $M((W')^*) = M(W^*)$). The latter number is $2^{O(\epsilon n \log n)}$.

We conclude that for $W \in F(S)$, the number of other triangulations $W' \in F(S)$ for which $M((W')^*) = M(W^*)$ is at most $2^{O(\epsilon n)}2^{O(\epsilon n \log n)} = 2^{O(\epsilon n \log n)}$. Now, the theorem follows from Lemma 3. ◀

5.2 Over-counting

The reason for over-counting in the estimation returned by our algorithm is that the same triangulation within a DP cell may be cut in the number of ways proportional to the number of considered partitions of the DP cell into at most k smaller DP-cells. The reason is similar to that for over-counting of the approximation triangulation counting algorithm of Alvarez, Bringmann, Ray, and Seidel [3] based on the planar simple cycle separator theorem. Therefore, our initial recurrences and calculations are similar to those derived in the analysis of the over-counting from [3].

▶ **Lemma 5.** *Let Q be an arbitrary DP cell processed by Algorithm 1 which contains more than Δ input points. Recall the calculation of the estimation for Q by summing the products of estimations for smaller DP cells $Q_1, ..., Q_l$ over $n^{O(k^2)}$ partitions of Q into $Q_1, ..., Q_l$, $l \le k$. Substitute the true value of the number of maximal triangulations within each such smaller cell Q_i for the estimated one in the calculation. Let r be the resulting value. The number of maximal triangulations within Q is at least $r/n^{O(k^2)}$.*

Proof. Note that r is the sum of the number of different combinations of maximal triangulations within smaller DP cells $Q_1, ..., Q_l$ over $n^{O(k^2)}$ partitions of Q into smaller cells $Q_1, ..., Q_l$, $l \le k$. Importantly, each such combination can be completed to some maximal triangulation within Q but no two different combinations coming from the same partition $Q_1, ..., Q_l$ can be extended to the same maximal triangulation within Q.

Let M be the set of maximal triangulations W within Q for which there is a partition into smaller DP cells $Q_1, ..., Q_l$, $l \le k$, such that for $i = 1, ..., l$, W constrained to Q_i is a maximal triangulation within Q_i. Note that for each $W \in M$, the number of the combinations that can be completed to W cannot exceed that of the considered partitions, i.e., $n^{O(k^2)}$, as each of the combinations has to come from a distinct partition $Q_1, ..., Q_l$.

Thus, there is a binary relationship between maximal triangulations within Q that belong to M and the aforementioned combinations. It is defined on all the maximal triangulations in M and on all the combinations, and a maximal triangulation in M is in relation with at most $n^{O(k^2)}$ combinations. This yields the lemma. ◀

By Lemma 5, we can express the over-counting factor $L(Q,\Delta)$ of Algorithm 1 for a DP cell Q by the following recurrence:

$$L(Q,\Delta) = \sum_{(Q_1,\dots,Q_l)} \prod_{j=1}^{l} L(Q_j,\Delta) \leq n^{O(k^2)} \prod_{j=1}^{l^*} L(Q_j^*,\Delta)$$

where the summation is over all partitions of Q into DP cells $Q_1,\dots,Q_l$, where $l \leq k$, and $Q_1^*,\dots,Q_{l^*}^*$ is a partition that maximizes the term $\prod_{j=1}^{l} L(Q_j,\Delta)$. When Q contains at most Δ input points, Algorithm 1 computes the exact number of maximal triangulations of these points within Q. Thus, we have $L(Q,\Delta)=1$ in this case.

Following [3], it will be more convenient to transform our recurrence by taking logarithm of both sides. For any DP cell P, let $L'(P,\Delta)=\log L(P,\Delta)$. We obtain now:

$$L'(Q,\Delta) \leq O(k^2 \log n) + \sum_{j=1}^{l^*} L'(Q_j^*,\Delta)$$

▶ **Lemma 6.** *Let B be a bounding box for a set S of n points in the plane. The following equality holds*

$$L'(B,\Delta) = O(k^2 n \log^2 n/\Delta)$$

Proof. Let U be the recurrence tree and let D be the set of direct ancestors of leaves in U. For each node $d \in D$, the corresponding DP cell includes at least $\Delta+1$ points in S. It follows that $|D| \leq n/\Delta$. Also, any node in D has depth $O(\log n)$ in U. Consequently, the contribution of the subproblems corresponding to nodes in D and their ancestors to the estimation for $L'(B,\Delta)$ can be upper bounded by $O(k^2 \log n \times (n/\Delta)\log n)$. Finally, recall that the subproblems corresponding to leaves of U do not contribute to the estimation. ◀

Lemma 5 immediately yields the following corollary.

▶ **Theorem 7.** *Let B be a bounding box for a set of n points in the plane. Set the parameter k in Algorithm 1 as in Theorem 4. If for $\epsilon > 0$ the parameter Δ in Algorithm 1 is set to $\frac{c}{\epsilon}k^2 \log^2 n$ for sufficiently large constant c then the over-counting factor is at most $2^{\epsilon n}$.*

6 Main result

By combining Lemma 2 with Theorems 4, 7 with ϵ set to $\epsilon/\log n$, we obtain our main result.

▶ **Theorem 8.** *There exists an approximation algorithm for the number of triangulations of a set S of n points with integer coordinates in the plane with a running time of at most $n^{(\log n/\epsilon)^{O(1)}}$ that returns a number at most $2^{\epsilon n}$ times smaller and at most $2^{\epsilon n}$ times larger than the number of triangulations of S.*

▶ **Corollary 9.** *There exists a $(1+\epsilon)$-approximation algorithm with a running time of at most $n^{(\log n/\epsilon)^{O(1)}}$ for the base of the number of triangulations of a set of n points with integer coordinates in the plane.*

Proof. Let c^n be the number of triangulations of the input n point set, and let Λ be the number returned by the algorithm from Theorem 8. We have $\max\{\frac{c^n}{\Lambda},\frac{\Lambda}{c^n}\} \leq 2^{\epsilon n}$ by Theorem 8. By taking the n-th root on both sides, we obtain $\max\{\frac{c}{\Lambda^{\frac{1}{n}}},\frac{\Lambda^{\frac{1}{n}}}{c}\} \leq 2^{\epsilon}$. Now it is sufficient to observe that $2^{\epsilon} < 1+\epsilon$ for $\epsilon < \frac{1}{2}$. ◀

7 Extensions

Adamaszek and Wiese presented also an extension of their theorem on α-cheap cut of an independent set of triangles (Fact 2) to include independent polygons with at most K edges (Lemma 3.1 [1]). This makes possible to generalize our QPTAS to include the approximation of the number of maximum weight partitions into K-gons.

Acknowledgments

We thank Victor Alvarez, Artur Czumaj, Peter Floderus, Miroslaw Kowaluk, Christos Levcopoulos and Mia Persson for preliminary discussions on counting the number of triangulations of a planar point set.

References

1 A. Adamaszek and A.Wiese. A QPTAS for Maximum Weight Independent Set of Polygons with Polylogarithmically Many Vertices. Proc. 25th ACM-SIAM Symposium on Discrete Algorithms (SODA'14), 2014.

2 A. Adamaszek and A.Wiese. Approximation Schemes for Maximum Weight Independent Set of Rectangles. Proc. 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS'13), 2013.

3 V. Alvarez, K. Bringmann, S. Ray, and R. Seidel. Counting Triangulations Approximately. Proceedings of the 25th Canadian Conference on Computational Geometry, 2013, pp. 212–243. An extended version "Counting Triangulations and other Crossing Free Structures Approximately" will appear in the special issue of Computational Geometry, Theory and Applications devoted to the 25th CCCG.

4 V. Alvarez and R. Seidel. A Simple Aggregative Algorithm for Counting Triangulations of Planar Point Sets and Related Problems. Proc. ACM Symposium on Computational Geometry 2013, pp. 1-8 .

5 S. Har-Peled. Quasi-Polynomial Time Approximation Scheme for Sparse Subsets of Polygons. Proc. ACM Symposium on Computational Geometry 2014.

6 G. L. Miller. Finding small simple cycle separators for 2-connected planar graphs. Journal of Computer and System Sciences, 32(3), pp. 265-279, 1986.

7 R. Sibson. Locally equiangular triangulations. The Computer Journal 21(3), pp. 243-245, 1975.

8 M. Sharir and A. Sheffer. Counting triangulations of planar point sets. Electr. J. Comb., 18(1), 2011.