Subexponential Parameterized Algorithms on Disk Graphs

Daniel Lokshtanov* Fahad Panolan† Saket Saurabh‡ Jie Xue§ Meirav Zehavi¶

December 2, 2021

Abstract

One of the most celebrated results in Parameterized Complexity is the Bidimensionality theory of Demaine et al. [J. ACM, 2005], which has yielded, over the past two decades, numerous subexponential-time fixed-parameter tractable (FPT) algorithms for various problems on planar (and H-minor-free) graphs. At the heart of this theory is the proof of sublinear bounds in terms of solution size on the treewidth of a given graph. Inspired by this theory, in recent years, significant efforts have been devoted to design subexponential-time FPT algorithms for problems on geometric graph classes that utilize new treewidth bounds, in particular (but not only) for unit disk graphs [Fomin et al., SODA'12; Fomin et al., DCG'19; Panolan et al., SODA'19; Fomin et al. SoCG'20]. In this paper, we aim to attain such results on disk graphs, a broad class of graphs that generalizes both the classes of planar graphs and unit disk graphs, and thereby unify the aforementioned research frontiers for planar and unit disk graphs.

Our main contribution is an approach to design subexponential-time FPT algorithms for problems on disk graphs, which we apply to several well-studied graph problems. At the heart of our approach lie two new combinatorial theorems concerning the treewidth of disk graphs having a realization of bounded ply (or maximum clique size) that are of independent interest. In particular, we prove a stronger version of the following treewidth bound:

> Let G be a disk graph that has some realization of ply p and no false twins, and $M \subseteq V(G)$ such that G has no triangle with exactly one vertex from M, and $G - M$ has treewidth w. Then, the treewidth of G is $\mathcal{O}(\max\{\sqrt{|M| \cdot w} \cdot p^{2.5}, w \cdot p\})$.

Among our applications are the first subexponential-time FPT algorithms for several problems on disk graphs, including Triangle Hitting, Feedback Vertex Set and Odd Cycle Transversal (OCT). Previously, subexponential-time FPT algorithms for these problems were only known on planar graphs and unit disk graphs (excluding OCT, which was only known to admit such an algorithm on planar graphs). Our algorithms are robust, in particular, they do not require a geometric realization of the input graph (for all aforementioned problems), and they generalize to the weighted and counting versions of all aforementioned problems except for OCT.

*University of California, Santa Barbara, US. daniello@ucsb.edu
†IIT Hyderabad, India. fahad@cse.iith.ac.in
‡The Institute of Mathematical Sciences, HBNI, Chennai, India. saket@imsc.res.in
§New York University Shanghai, China. jiexue@nyu.edu
¶Ben-Gurion University of the Negev, Beersheba, Israel. meiravze@bgu.ac.il

1 Introduction

The design of algorithms for NP-hard optimization problems on intersection graphs of systems of geometric objects is a well studied area in Computer Science, motivated by various applications in wireless networks [30], computational biology [42] and map labeling [1]. While most problems remain NP-complete even when restricted to some of the simpler such classes like unit disk graphs, the restriction of a problem to a geometric class is usually much more tractable algorithmically than the unrestricted problem. In particular, a common theme that has emerged in paramaterized graph algorithms over the last two decades is that NP-hard problems become computationally more tractable on planar graphs and several classes of geometric intersection graphs. Usually, significantly better running times can be achieved. Sometimes, even problems that are W[1]-hard on general graphs can become fixed-parameter tractable (FPT) on planar graphs (and, more generally, on H-minor free graphs) parameterized by the solution size k. In most cases, the running time is of the form $2^{\mathcal{O}(\sqrt{k})}n^{\mathcal{O}(1)}$ or $n^{\mathcal{O}(\sqrt{k})}$, leading to the term "square root phenomenon" [34, 35].

The square root phenomenon exhibited by problems on planar graphs, and, in general, on H-minor free graphs, can also be observed on intersection graphs of objects such as unit disks, disks, unit rectangles, strings, similarly-sized convex fat objects or their generalizations in $\mathbb{R}^d$ for $d \geq 3$. Several algorithmic tools have been developed for designing subexponential-time FPT algorithms for intersection graphs defined by "similar sized fat objects" such as unit disks and unit squares [3, 2, 24, 23, 21, 22, 37] (also see the survey [38]). On the other hand, Marx and Pilipczuk [36] designed $n^{\mathcal{O}(\sqrt{k})}$-time algorithms for several facility location problems on disk graphs, including an algorithm with running time $n^{\mathcal{O}(\sqrt{k})}$ for Dominating Set. However, to the best of our knowledge, there are almost no tools and techniques that have been developed for designing subexponential-time fixed-parameter algorithms for problems on intersection graphs of fat objects that are allowed to differ substantially in size, such as disk graphs.

We aim to design subexponential-time FPT algorithms for fundamental graph problems—such as Feedback Vertex Set, and Odd Cycle Transversal—on disk graphs, a class simultaneously generalizing planar graphs and unit disk graphs.

Our algorithms are robust, in particular, they do not require a geometric realization of the input graph, and they generalize to the weighted and counting versions of all aforementioned problems except for Odd Cycle Transversal.

Bidimensionality and Sublinear Treewidth. The square root phenomenon for planar graphs, and, in general, for H-minor free graphs, can be explained by one of the most celebrated results in Parameterized Complexity, the Bidimensionality theory of Demaine et al. [15]. The Bidimensionality theory gave a uniform explanation for the existence of subexponential-time algorithms as well as subexponential-time FPT algorithms for NP-hard graph problems on a broad range of graph classes, including planar graphs, map graphs, bounded-genus graphs and graphs excluding any fixed graph as a minor [14, 13, 15, 17, 16] (see also [6, 24, 9, 9, 25] for other graph classes).

For all bidimensional problems, there is a relation between the size of the largest grid minor and the size of the optimum solution, which allows to bound the treewidth of the graph in terms of the parameter of the problem. In particular, let $\mathcal{G}$ be a family of graphs and Π be a graph problem. The problem Π is said to have *solution sensitive sublinear treewidth* (S3T) property on $\mathcal{G}$ if at least one of the following holds: (i) for all (G, k) such that $G \in \mathcal{G}$ and (G, k) is a Yes-instance, we have that $\mathtt{tw}(G) \in o(k)$; (ii) for all (G, k) such that $G \in \mathcal{G}$ and (G, k) is a No-instance, we have that $\mathtt{tw}(G) \in o(k)$. In general, this combinatorial result on all Yes-instances (or all No-instances) of a problem, together with a known dynamic programming algorithm on graphs of bounded treewidth, yields a $2^{o(k)}n^{\mathcal{O}(1)}$ time algorithm for a problem on $\mathcal{G}$. In order to apply Bidimensionality theory to obtain subexponential parameterized algorithms for a problem, the key step is to prove that it satisfies the S3T property on a graph class. For more information, we refer to the chapter on Bidimensionality in [10].

Extension of Bidimensionality to Unit Disk Graphs and Map Graphs. Inspired by this theory, in recent years, significant efforts have been devoted in designing subexponential-time FPT algorithms for problems on geometric graph classes, such as unit disk graphs, unit square graphs, string graphs and map graphs, with a particular focus on unit disk graphs. To do so, one approach is to show that the problem admits the S3T property on the corresponding families of graphs. Unfortunately, unit disk graphs can have large cliques as induced subgraphs, and thus we cannot even show that if G is an arbitrary unit disk graph on n vertices, then $\mathtt{tw}(G) \in o(n)$. However, for many parameterized problems (such as VERTEX COVER, FEEDBACK VERTEX SET, and ODD CYCLE TRANSVERSAL) there are simple generic procedures that reduce the problem on a graph G to the same problem on an induced subgraph of G that does not contain any large cliques, at the cost of a sub-exponential in k overhead in the running time. Fomin et al. [24] showed that several problems on unit disk graphs of bounded clique size (and hence bounded degree) and map graphs of bounded clique size, satisfy the S3T property, and obtained subexponential-time FPT algorithms for these problems on unit disk graphs and map graphs. Fomin et al. [23, 21, 22] further refined this theory on unit disk graphs and map graph and obtained algorithms with running time nearly matching the asymptotic running time lower bounds assuming the Exponential Time Hypothesis (ETH). More recently, Panolan et al. [37] gave a weaker form of a contraction decomposition theorem for unit disk graphs and used this to design faster FPT algorithms than on general graphs for several cut problems such as MINIMUM BISECTION, STEINER CUT, and EDGE MULTIWAY CUT-UNCUT.

Difficulty in Extending Bidimensionality to Disk Graphs. Next, we explain some of the difficulties one faces when trying to extend the key idea of Bidimensionality from unit disk graphs to disk graphs. For this purpose, we make use of the concept of ply, a widely used notion related to geometric graphs that has turned out be very useful algorithmically (see, e.g., [9]). A system of disks has *ply* p if every point in the plane is contained inside at most p disks. The ply of a system of disks is upper bounded by the size of the maximum clique in the corresponding intersection graph. Recall that for unit disk graphs, we can branch to reduce the maximum clique size, and hence also the ply. On disk graphs, for problems such as VERTEX COVER, TRIANGLE HITTING, FEEDBACK VERTEX SET and ODD CYCLE TRANSVERSAL, we can branch similarly to unit disk graph and reduce the ply (or the maximum clique size) in parameterized subexponential time. However, unlike unit disk graphs, we cannot show that the aforementioned problems satisfy the S3T property on disk graphs of bounded ply (or clique size). In particular, Fomin et al. [24] (in Section 8.3) present an infinite subfamily of disk graphs, say, $\mathcal{F}$, of ply 3 such that for every $k \in \mathbb{N}$, there exists $G \in \mathcal{F}$ such that (G, k) is a Yes-instance of TRIANGLE HITTING, FEEDBACK VERTEX SET and ODD CYCLE TRANSVERSAL, while $\mathtt{tw}(G) \in \Omega(k)$.

On the one hand, planar graphs exclude $K_{3,3}$ (complete bipartite graph with three vertices on each side) and in particular K_5 (clique on five vertices) as minors. This allows proving a linear relation between the treewidth of the graph and the maximum size of a grid it contains as a minor, which helps in proving the S3T property. On the other hand, unit disk graphs exclude $K_{1,6}$ (star with six leaves) as an induced subgraph. This allows to prove the S3T property on unit disk graphs of bounded ply. Thus, the exclusion of either K_5 as a minor or of $K_{1,6}$ as an induced subgraph significantly simplify both classes of graphs, and, in turn, the design of algorithms. For disk graphs, none of these exclusions holds. Among other complications, this means that a neighborhood set of a vertex can be very complicated—it can contain arbitrarily many pairwise-adjacent vertices as well as arbitrarily many pairwise-nonadjacent vertices simultaneously. As a simple example, note that in disk graphs, we can have a vertex with arbitrarily many vertex-disjoint large cliques in its neighborhood.

Nevertheless, for problems that admit linear-vertex kernels, the above difficulties are sidestepped, since then the solution size and total number of vertices are made linearly related. Specifically, for such problems, one only has to design a subexponential-time ($2^{o(n)}$) algorithm, and then a subexponential-time FPT ($2^{o(k)}$) algorithm is implied. In particular, such an algorithm is implied for VERTEX COVER from the work of [12]. However, for almost all natural parameterized problems, a linear-vertex kernel is provably unlikely to exist (unless the polynomial-hierarchy collapses) [10] or is not known to exist, even on disk graphs.

1.1 Our Contribution

Our Combinatorial Theorems and Their Use. We overcome the above-mentioned difficulties as follows. We start with some simple preprocessing. First, similarly to the case of unit disk graphs, we branch to decrease the ply p of the graph. (Throughout, we use p to denote the ply of the graph under consideration.) However, as already argued earlier, this is insufficient. So, we further branch to reach a case that we can analyze combinatorially. Specifically, we attain a "small" set M (of size $\mathcal{O}(k \cdot p)$) such that no vertex in M has two adjacent vertices that do not belong to M in its neighborhood. This takes time $k^{\mathcal{O}(\frac{k}{p})} n^{\mathcal{O}(1)}$. To analyze the instances yielded by branching as above, we prove two combinatorial theorems concerning the treewidth of disk graphs, which are of independent interest. Essentially, the correctness of each of our algorithms hinges on either one or both of them. The theorems are not used algorithmically, but only for the analysis of the algorithms. We remark that the bounds given in these theorems might not be tight, yet they suffice for the purpose of deriving subexponential-time FPT algorithms.

Informally, our first theorem states that the maximum number of disks that are combinatorially different with respect to M—that is, vertices whose neighborhoods inside M are different from one another—is bounded by $\mathcal{O}(|M| \cdot p^6) \subseteq \mathcal{O}(k \cdot p^7)$ (on general graphs, we only have a trivial upper bound of $2^{|M|}$). We remark that this theorem suffices to resolve Triangle Hitting. In particular, combined with Proposition 3.10 (stating that the treewidth of a disk graph G of ply p is $\mathcal{O}(\sqrt{|V(G)|p})$), this gives an upper bound on treewidth under the assumption that the graph contains neither *(i)* edges between vertices outside M, nor *(ii)* false twins (non-adjacent vertices with the same neighborhood). For Triangle Hitting, *(i)* holds due to our second branching step and explicit removal of leftover edges between vertices outside M (which become irrelevant edges, that is, not part of any triangle). False twins are not deleted, but we just keep one representative for each class of false twins (remembering how many vertices it represents). Formally, our first theorem is stated as follows.

Theorem 1. *Let G be a disk graph that has some realization of ply p, and let $M \subseteq V(G)$. Let $U \subseteq V(G) \setminus M$ be such that for all distinct $u, v \in U$, $N_G(u) \cap M \neq N_G(v) \cap M$. Then, $|U| \in \mathcal{O}(|M| \cdot p^6)$.*

Our second combinatorial theorem, for whose proof we invest most efforts in this paper, assumes the conditions arrived after both our preprocessing branching steps as well as another condition: no vertex outside M can have neighbors only inside M. In other words, every vertex outside M has at least one neighbor outside M. For the sake of clarity, we first present a special case of our second theorem. Roughly speaking, this special case, termed Corollary 1.1, asserts that if $G - M$ has "small" treewidth, then also the entire graph G has "small" treewidth (assuming that p is small). More precisely, it states the following.

Corollary 1.1. *Let G be a disk graph that has some realization of ply p, and let $M \subseteq V(G)$ be such that for all $v \in M$, $N_G(v) \setminus M$ is an independent set, there does not exist a vertex in $V(G) \setminus M$ whose neighborhood is contained in M, and $G - M$ has treewidth w. Then, the treewidth of G is $\mathcal{O}(\max\{\sqrt{|M| \cdot w} \cdot p^{2.5}, w \cdot p\})$.*

This corollary, combined with our first theorem, will suffice for Feedback Vertex Set. Notice that for this problem, if M contains a feedback vertex set of the given graph, then the treewidth of $G - M$ will just be 1. However, for Odd Cycle Transversal, this will not suffice, as even if M contains an odd cycle transversal of G, the graph $G - M$ (being a bipartite graph) can have large treewidth. For this reason, we prove the following stronger version of Corollary 1.1. Here, we use the following notation: for $U \subseteq V(G)$, G/U is the graph obtained from G by contracting each connected component of $G[U]$ into a single vertex.

Theorem 2. *Let G be a disk graph that has some realization of ply p, and let $M \subseteq V(G), Q \subseteq V(G) \setminus M$ be such that for all $v \in M$, $N_G(v) \setminus M$ is an independent set, there does not exist a vertex in $V(G) \setminus M$ whose neighborhood is contained in M, and $(G - M)/Q$ has treewidth w. Then, the treewidth of G/Q is $\mathcal{O}(\max\{\sqrt{|M| \cdot w} \cdot p^{2.5}, w \cdot p\})$.*

Note that Corollary 1.1 is the special case of Theorem 2 where $Q = \emptyset$. Specifically for Odd Cycle Transversal, while $G - M$ can have large treewidth, we will show how to find certain sets of vertices (each corresponding to Q in the theorem) so that the contraction of the edges of a spanning forest of any of them significantly decreases treewidth, and it will be "valid" to contract at least one of them.

One question that arises is how to satisfy the condition (in Theorem 2) that there does not exist a vertex in $V(G) \setminus M$ whose neighborhood is contained in M. This is where Theorem 1 comes into play—from this theorem, we know that, if we ignore false twins, then the number of vertices in $V(G) \setminus M$ whose neighborhood is contained in M is "small". Then, we can insert all these vertices into M, and thus force this condition. Specifically, combining Theorem 1 and Corollary 1.1 yields the following: Let G be a disk graph that has some realization of ply p and no false twins, and $M \subseteq V(G), Q \subseteq V(G) \setminus M$ such that G has no triangle with exactly one vertex from M, and $(G - M)/Q$ has treewidth w. Then, the treewidth of G/Q is $\mathcal{O}(\max\{\sqrt{|M| \cdot w} \cdot p^{2.5}, w \cdot p\})$. This implication (which will be used implicitly in our proofs) is essentially the core of the correctness of our algorithms for Feedback Vertex Set and Odd Cycle Transversal.

Applications. We attain the first subexponential-time FPT algorithms for several well-known parameterized problems on disk graphs with respect to solution size. Our algorithms are robust, in particular, they do not require a geometric realization of the input graph (for all aforementioned problems), and they generalize to the weighted and counting versions of all aforementioned problems except for Odd Cycle Transversal. Specifically, we prove the following:

Theorem 3. Triangle Hitting *and* Feedback Vertex Set *on disk graphs admit (deterministic)* $2^{\mathcal{O}(k^\alpha)} n^{\mathcal{O}(1)}$*-time algorithms, for some fixed constant* $\alpha < 1$. *The statement holds also for the weighted and counting generalizations of these problems.*

Theorem 4. Odd Cycle Transversal *on disk graphs admits a randomized* $2^{\mathcal{O}(k^\alpha)} n^{\mathcal{O}(1)}$*-time algorithm, for some fixed constant* $\alpha < 1$.

The problems Triangle Hitting, Feedback Vertex Set and Odd Cycle Transversal are all extensively and intensively studied problems in the realm of Parameterized Complexity [10]. There are numerous papers that study the fixed-parameter tractability and kernelization complexity of them, their variants, and their restrictions to special graph classes. In fact, Feedback Vertex Set is (arguably) the second most well studied problem in Parameterized Complexity (with Vertex Cover being the first). Here, we only briefly mention the fastest algorithms for these problems on general graphs, planar graphs and unit disk graphs. We remind that on disk graphs, none of these problems was previously known to admit a subexponential-time FPT algorithm. On general graphs, the current fastest algorithms for Triangle Hitting and Feedback Vertex Set and Odd Cycle Transversal run in time $2.1^k \cdot n^{\mathcal{O}(1)}$ [40] (as a special case of 3-Hitting Set, $2.7^k \cdot n^{\mathcal{O}(1)}$ [31] and $2.32^k \cdot n^{\mathcal{O}(1)}$ [32], respectively. Among these algorithms, only the one for Feedback Vertex Set (given by [31]) is randomized; for this problem, the current fastest deterministic algorithm runs in time $3.46^k \cdot n^{\mathcal{O}(1)}$ [28]. On planar graphs, subexponential-time FPT algorithms for Triangle Hitting and Feedback Vertex Set follow from the seminal work of Demaine et al. [15] on Bidimensionality. For Odd Cycle Transversal on planar graphs, a subexponential-time FPT algorithm was given in [33]. On unit disk graphs, subexponential-time FPT algorithms for Triangle Hitting and Feedback Vertex Set follow (although for Triangle Hitting this is not explicitly stated) from the works of Fomin et al. [24, 22].

Reading Guide. First, in Section 2, we present the outline of our proofs of Theorems 1 and 2, as well as our approach for deriving our applications. In Section 3, we present basic terminology used throughout the paper; readers unfamiliar with (standard) notions used in Section 2 are referred to this section. Later, in Sections 4 and 5, we present the full proofs of Theorems 1 and 2, respectively. In Section 6, we show how to use Theorems 1 and 2 (combined with two branching steps) to derive our applications in detail, in order to prove Theorems 3 and 4. Finally, we conclude the paper in Section 7.

2 Overview of Our Proof Techniques

Most efforts in this paper are invested in the proof of Theorem 2. We remark that both Theorems 1 and 2 are simultaneously required for two of our three applications. In what follows, we give an informal and brief overview of our proofs of both combinatorial theorems, and then of our approach to derive our applications. Readers unfamiliar with some of the terms used are referred to Section 3.

2.1 Overview of the Proof of Theorem 1

For the sake of clarity, we restate the theorem below.

Theorem 1. *Let G be a disk graph that has some realization of ply p, and let $M \subseteq V(G)$. Let $U \subseteq V(G) \setminus M$ be such that for all distinct $u, v \in U$, $N_G(u) \cap M \neq N_G(v) \cap M$. Then, $|U| \in \mathcal{O}(|M| \cdot p^6)$.*

There can exist at most one vertex in U with no neighbors in M, hence we can focus just on bounding the number of vertices in U that do have neighbors in M. So, for the sake of simplicity of the rest of this overview, we suppose that each vertex in $V(G) \setminus M$ has at least one neighbor in M. Further, we fix some realization $\mathcal{D}$ of G of ply at most p.

The heart of the proof is in a charging argument, where the vertices in $V(G) \setminus M$ are charged to vertices in M so that each vertex in $V(G) \setminus M$ is charged to at least one vertex in M. Then, the objective is to show that every vertex in M gets charged at most $\mathcal{O}(p^6)$ times. The way we charge vertices is as follows (see Fig. 1):

> **Charging.** Each vertex u in $V(G) \setminus M$ is charged to some neighbor v of u in M whose disk (in $\mathcal{D}$) is of minimum size among those of the neighbors of u in M.

So, we next fix some vertex $v \in M$, and aim to show that at most $\mathcal{O}(p^6)$ vertices from U were charged to it. For this purpose, we let A_v denote all vertices from $V(G) \setminus M$ charged to v, and classify them into two categories:

1. A_v^{large}: Those whose disk is at least as large as that of v.
2. A_v^{small}: Those whose disk is smaller than that of v.

Because the ply is at most p, we derive that $|A_v^{\mathsf{large}}| \in \mathcal{O}(p)$. Next, we focus on A_v^{small}. To upper bound the maximum number of vertices having different neighborhoods in M that belong to this set, we let B denote the union of the sets of neighbors in M of the vertices in A_v^{small}. By the way we charge v, we have that the disks of all vertices in B are at least as large as the disk of v. In turn, because the ply is at most p, we derive that $|B| \in \mathcal{O}(p)$.

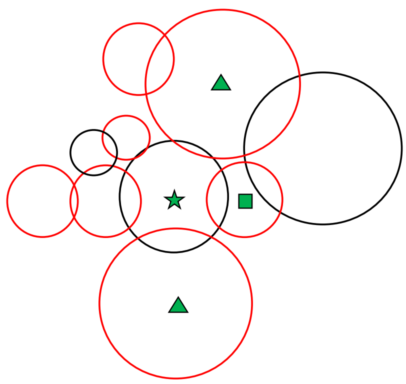

Figure 1: The disks corresponding to vertices in M are colored black. Let v be the vertex whose disk is marked by a star. Then, the vertices charged to v are those whose disks are marked by either a triangle (belonging to A_v^{large}) or a square (belonging to A_v^{small}).

Next, we prove an independent lemma, stating that in a disk graph G with a subset $S \subseteq V(G)$, there can be at most $\mathcal{O}(|S|^6)$ vertices whose neighborhoods have different intersections with S. To prove this, we consider a realization $\mathcal{D}$ in general position, and a disk D corresponding to a vertex v in $G - S$. Then, we modify D (in three steps) to become one of the following: *(i)* a single point; *(ii)* the smallest disk tangent to exactly two disks of vertices in S; *(iii)* tangent to exactly three disks of vertices in S. In particular, the disk obtained after modification intersects the same disks that it did before the modification, with the exception of the disks it is tangent to. So, every possible intersection of a neighborhood of a vertex in $G - S$ with S can be described by the choice of a point or (two or three) edges in the *graph drawn by* $\mathcal{D}$, which contains $\mathcal{O}(|S|^2)$ edges.

The bound $|B| \in \mathcal{O}(p)$ together with the aforementioned independent claim yield that $|A_v^{\mathsf{small}}| \in \mathcal{O}(p^6)$, which completes the proof.

2.2 Overview of the Proof of Theorem 2

For the sake of clarity, we restate the theorem below.

Theorem 2. *Let G be a disk graph that has some realization of ply p, and let $M \subseteq V(G), Q \subseteq V(G) \setminus M$ be such that for all $v \in M$, $N_G(v) \setminus M$ is an independent set, there does not exist a vertex in $V(G) \setminus M$ whose neighborhood is contained in M, and $(G - M)/Q$ has treewidth w. Then, the treewidth of G/Q is $\mathcal{O}(\max\{\sqrt{|M| \cdot w} \cdot p^{2.5}, w \cdot p\})$.*

Let us first explain how we derive the special case where $Q = \emptyset$, denoted as Corollary 1.1. For the sake of clarity, let us restate this corollary as well:

Corollary 1.1. *Let G be a disk graph that has some realization of ply p, and let $M \subseteq V(G)$ be such that for all $v \in M$, $N_G(v) \setminus M$ is an independent set, there does not exist a vertex in $V(G) \setminus M$ whose neighborhood is contained in M, and $G - M$ has treewidth w. Then, the treewidth of G is $\mathcal{O}(\max\{\sqrt{|M| \cdot w} \cdot p^{2.5}, w \cdot p\})$.*

We fix some realization $\mathcal{D}$ (say, in general position) of G of ply at most p. We observe that, because for all $v \in M$, $N_G(v) \setminus M$ is an independent set, and there does not exist a vertex outside M whose neighborhood is contained in M, it follows that there does not exist a disk corresponding to a vertex outside M that is contained in the union of the disks of vertices in M. Let us refer to this property (in the overview) as the *Exclusion Property*.

Roughly speaking, the *arrangement graph* of $\mathcal{D}$, denoted here by $\mathfrak{A}(G)$, is the plane graph where every face of $\mathcal{D}$ contained in at least one disk is represented by a vertex, and vertices are adjacent if the faces they represent share a boundary edge. Our starting point is the observation that to bound the treewidth of G by $\mathcal{O}(\max\{\sqrt{|M| \cdot w} \cdot p^{2.5}, w \cdot p\})$, it suffices to bound the treewidth of $\mathfrak{A}(G)$ by $\mathcal{O}(\max\{\sqrt{|M| \cdot w} \cdot p^{1.5}, w\})$. Indeed, the treewidth of G is upper bounded by the treewidth of $\mathfrak{A}(G)$ times a factor of $\mathcal{O}(p)$ (see Proposition 3.12). Further, the treewidth of the arrangement graph of $\mathcal{D}$ restricted to $G - M$ (that is, of $\mathcal{D}$ where all disks corresponding to vertices in M are removed), denoted here by $\mathfrak{A}(G - M)$, is $\mathcal{O}(w)$. So, we aim to relate the treewidth of $\mathfrak{A}(G)$ to the treewidth of $\mathfrak{A}(G - M)$. In what follows, we slightly abuse notation and use w to refer to the treewidth of $\mathfrak{A}(G - M)$.

Let U denote the vertices in $\mathfrak{A}(G)$ that represent faces that are each contained in at least one disk of a vertex from M. Our first insight is that, because for all $v \in M$, $N_G(v)$ is an independent set, and due to the Exclusion Property, the following equality holds: The removal of the vertices in U (and the edges incident to them) from $\mathfrak{A}(G)$ yields a graph isomorphic to $\mathfrak{A}(G - M)$. Hence, the objective is further "simplified", being to bound the treewidth of $\mathfrak{A}(G)$, which we here denote by $w^\star$, by making use of the supposition that the treewidth of $\mathfrak{A}(G) - U$ is w. How can we relate the treewidth of these two plane graphs?

To answer the above question, we turn to analyze U. More precisely, we analyze the set of faces (of $\mathcal{D}$) that are represented by the vertices in U—that is, the set of faces that are each contained in at least one disk of a vertex in M. Such a face is called *relevant*, and the set of such faces is denoted by $\mathsf{relevant}(\mathcal{D}, M)$. The core of the proof is in the identification of a subset of faces of $\mathsf{relevant}(\mathcal{D}, M)$, denoted by $\mathsf{critical}(\mathcal{D}, M)$, that has the two following properties:

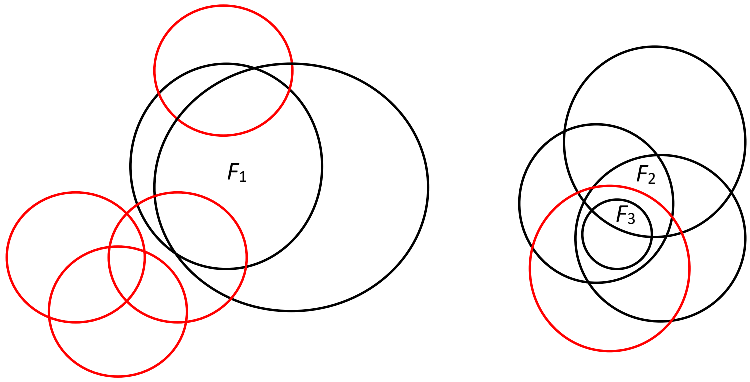

Figure 2: Each among faces F_1, F_2 and F_3 has the following properties: it is contained in at least one black disk, each one of its black boundary edges is convex with respect to it, and each one of its red boundary edges is concave with respect to it. Assuming that the black disks (only) correspond to vertices in M, we have that F_1, F_2 and F_3 are critical faces, and there is no other critical face.

- **Property I:** Each face in $\mathsf{relevant}(\mathcal{D}, M)$ is at distance at most $\mathcal{O}(p)$ from at least one face in $\mathsf{critical}(\mathcal{D}, M)$.
- **Property II:** $|\mathsf{critical}(\mathcal{D}, M)| \in \mathcal{O}(p \cdot |M|)$.

Before we discuss the definition of $\mathsf{critical}(\mathcal{D}, M)$ and our proofs of these two properties, let us see how this leads to the desired bound on the treewidth of $\mathfrak{A}(G)$. For this purpose, observe that the two properties imply that U contains a subset U' of size $\mathcal{O}(p \cdot |M|)$ such that every vertex in U is at distance (in $\mathfrak{A}(G)$) at most $\mathcal{O}(p)$ from at least one vertex from U'.

To proceed, we need to make a short detour and discuss grids. We note the following:

- It is known that the treewidth of a planar graph and the largest t such that it contains a $t \times t$ grid graph as a *minor* are linearly related [10].
- We observe that by removing at most s vertices from a $t \times t$ grid graph, we cannot destroy all $t' \times t'$ grid graphs for $t' \in \Omega(\min\{\frac{t^2}{s}, t\})$ that it contains as minors.
- We observe that by removing a set of at most r vertices, as well as all vertices at distance at most δ from at least one of them, from a planar graph having a $t \times t$ grid H as a minor, we can intersect at most $\mathcal{O}(\delta^2 \cdot r)$ sets from those of the planar graph tha are mapped to the vertices of H (according to the minor model of H in the planar graph).

Altogether, these three items imply that if we remove from $\mathfrak{A}(G)$ all of the vertices of U' as well as all vertices at distance at most $\mathcal{O}(p)$ from at least one vertex from U' (which is a superset of U), the treewidth of the obtained graph will be at least $\Omega(\min\{\frac{w^{\star 2}}{|U'| \cdot p^2}, w^\star\})$. Hence, because $|U'| \in \mathcal{O}(p \cdot |M|)$, we get that $w \in \Omega(\min\{\frac{w^{\star 2}}{|M| \cdot p^3}, w^\star\})$, which means that $w^\star \in \mathcal{O}(\max\{\sqrt{|M| \cdot w} \cdot p^{1.5}, w\})$. As argued above, this completes the proof.

In what follows, we discuss the issues left open so far.

The definition of critical faces and the proof of Property I. Towards the definition of critical faces, we first note that given a face F of $\mathcal{D}$, we have a natural classification of its boundary edges as *convex* or *concave* with respect to F (see Fig. 2). Then, we say that a face F of $\mathcal{D}$ is *critical* if it is relevant, every boundary edge of F that belongs to a disk whose vertex is in M is convex with respect to F, and every boundary edge of F that belongs to a disk whose vertex is outside M is concave with respect to F.

For the proof of Property I, we consider some $F \in \mathsf{relevant}(\mathcal{D}, M)$ and propose the following process:

1. Initialize $F^\star = F$.

2. As long as $F^\star$ has a boundary edge b that is either (i) concave and belongs to a disk whose vertex is in M or (ii) convex and belongs to a disk whose vertex is outside M, update $F^\star$ as follows:

   (a) Let $\widehat{F}$ be the other face of $\mathcal{D}$ that has b as a boundary edge.

   (b) Update $F^\star$ to be $\widehat{F}$.

At the end of this process, we reach a critical face. So, we just need to show that this process makes $\mathcal{O}(p)$ steps. On the one hand, we argue that, because the ply is p, we cannot take a step based on condition (i) most than p times. On the other hand, we argue that, because for all $v \in M$, $N_G(v) \setminus M$ is an independent set, we cannot take a step based on condition (ii) in two consecutive iterations. Overall, this yields Property I.

The proof of Property II. We first observe that for the proof of this property, it suffices to show that every face of the restriction of $\mathcal{D}$ to M contains at most one critical face. Indeed, this follows because it is known that the number of faces of a realization of a disk graph of ply p on x vertices is at most $\mathcal{O}(p \cdot x)$ (see Proposition 3.11). For simplicity, let us use $\mathcal{D}|_M$ to denote the restriction of $\mathcal{D}$ to M. The main ingredient in this proof, which will critically require the Exclusion Property, is the following sequence of containments:

> **Claim.** Let F be a critical face. Let $D_1, \ldots, D_\ell$ be the disks in $\mathcal{D}|_M$ corresponding to the boundary edges of F, and let $D'_1, \ldots, D'_t$ be the disks outside $\mathcal{D}|_M$ corresponding to the boundary edges of F. Let W be the (unique) face of $\mathcal{D}|_M$ that contains F. Then, $F \subseteq W \subseteq \bigcap_{i=1}^{\ell} D_i \subseteq F \cup (\bigcup_{i=1}^{t} D'_i)$.

Let us first see how to derive Property II from this claim. Here, targeting a contradiction, we suppose that there exist two critical faces, F and $\widehat{F}$, contained in the same face W of $\mathcal{D}|_M$. By using the claim above, we are able to derive that $\widehat{F}$ is contained $\bigcup_{i=1}^{t} D'_i$. Then, we can argue that, because for all $v \in M$, $N_G(v) \setminus M$ is an independent set, $\widehat{F}$ is contained in one of the D'_i's, and, furthermore, one of its boundary edges is convex with respect to that D'_i. However, that contradicts the supposition that $\widehat{F}$ is critical.

Now, let us briefly consider the proof of the above claim. Here, only the containment $\bigcap_{i=1}^{\ell} D_i \subseteq F \cup (\bigcup_{i=1}^{t} D'_i)$ is not immediate. In order to show it, we prove an auxiliary claim stating the following. Let I and U be the intersection and union of some set of disks, respectively. Also, suppose that I is convex, and let D be some other disk. Then, if the removal of $I \cap D$ from I splits I into two (or more) regions, it follows that $D \subseteq I$. In turn, due to the Exclusion Property, this auxiliary claim implies that the removal of $D'_j \cap (\bigcap_{i=1}^{\ell} D_i)$ from $\bigcap_{i=1}^{\ell} D_i$, for any of the D'_j's, does not split $\bigcap_{i=1}^{\ell} D_i$ into two (or more) regions. Then, we see that the same holds true also when we remove $(\bigcup_{i=1}^{t} D'_i) \cap (\bigcap_{i=1}^{\ell} D_i)$ from $\bigcap_{i=1}^{\ell} D_i$. In particular, this means that when we remove $(\bigcup_{i=1}^{t} D'_i) \cap (\bigcap_{i=1}^{\ell} D_i)$ from $\bigcap_{i=1}^{\ell} D_i$, we attain F. From this, we can derive the desired containment $\bigcap_{i=1}^{\ell} D_i \subseteq F \cup (\bigcup_{i=1}^{t} D'_i)$.

The general case (where Q may not be empty). We first note that there exists a natural way to associate a realization by objects that are connected unions of disks (rather than just disks) with G/Q, and, hence, also $(G - M)/Q$, based on $\mathcal{D}$: Each vertex in G/Q that belongs also to G is associated with its disk from $\mathcal{D}$, and each other vertex in G/Q is associated with the union of the disks of the connected component of $G[Q]$ that yielded this vertex. We denote this new realization by $\mathcal{D}/Q$. Because $Q \subseteq V(G) \setminus M$, and since for all $v \in M$, $N_G(v) \setminus M$ is an independent set, we are able to show that each of the relevant faces of $\mathcal{D}$ appears unchanged in $\mathcal{D}/Q$. Having this at hand, our earlier arguments for the case of $Q = \emptyset$ "go through" (in fact, most of them remain the same) also in case Q is not empty.

2.3 Overview of the Proofs of Our Applications

For the sake of simplicity, here we focus on the weighted versions of our problems (for Odd Cycle Transversal, we only resolve the decision unweighted version). At the end, we will comment about the counting versions. Let Π be any of our problems of interest, and let $(G, \mathfrak{w}, k, W)$ be an instance of it, where $\mathfrak{w}$ is the input vertex-weight function. Essentially, the general scheme that we use in order to derive all of our applications is the following. (We refer to Section 6.1 for more information on the first two steps.)

1. **Ply Reduction.** We observe that any solution to $(G, \mathfrak{w}, k, W)$ must contain all except for at most two vertices from each of the cliques of G. So, by recursively branching on which vertices to delete from a clique,[1] we produce $2^{\mathcal{O}(\frac{k}{p}\cdot\log p)}$ instances of Π where the maximum clique size and hence also the ply (of any realization) of the graph is at most p, and so that the original instance $(G, \mathfrak{w}, k, W)$ is a yes-instance if and only if at least one of them is. In what follows, we abuse notation and let $(G, \mathfrak{w}, k, W)$ denote one of the produced instances.

2. **Independent Neighborhoods.** We initialize M to be a triangle hitting set of G of size at most $3k$ (if we cannot find such an M, we can conclude that we have a no-instance). Then, as long as we have a vertex v in M such that the intersection of its neighborhood with $V(G) \setminus M$ has a matching of size larger than p, we branch on whether to delete v. In case we delete v, k is decreased by 1 (and W is updated accordingly). In the other case, we insert all the vertices in the matching into M, and "know" that we must delete at least one vertex from each edge in the matching (since, in all of our problems, we must hit all triangles)—so, we "virtually" decrease k by the size of the matching (here, the decrease is not done in the actual instance, but externally, just to bound the number of instances produced). Notice that, because M is a triangle hitting set, each of the newly inserted vertices already satisfies that its neighborhood outside M is an independent set. At the end of the above branching process, each of the vertices in M does not have a matching of size larger than p in the intersection of its neighborhood with $V(G) \setminus M$, and hence, by inserting all vertices of the corresponding maximum matchings, we increase M to be of size at most $\mathcal{O}(p \cdot |M|) \subseteq \mathcal{O}(p \cdot k)$, and attain the property that for all $v \in M$, $N_G(v) \setminus M$ is an independent set. Overall, we thus produce $2^{\mathcal{O}(\frac{k}{p}\cdot\log k)}$ instances of Π each accompanied by a set M, where the ply (of any realization) of the graph is at most p, where for all $v \in M$, its neighborhood outside M is an independent set, and where $|M| \in \mathcal{O}(p \cdot k)$, so that: the original instance $(G, \mathfrak{w}, k, W)$ is a yes-instance if and only if at least one of them is. In what follows, we abuse notation and let $(G, \mathfrak{w}, k, W)$ denote one of the produced instances.

3. **Application of Theorem 1.** Possibly, we mark some edges in G as "irrelevant". Afterwards, we consider the set X of vertices whose neighborhoods (excluding irrelevant edges) are contained in M. By Theorem 1 (whose usage is valid due to the first step in this scheme), the number of false twin classes in X (i.e., maximal sets of vertices having the same neighborhood) is at most $\mathcal{O}(p^6 \cdot |M|) \subseteq \mathcal{O}(p^7 \cdot k)$.

4. **Elimination of False Twins.** We update G (as well as $\mathfrak{w}$) so that, from each class of false twins in X, we keep only a constant number of vertices (for our applications, one or two vertices suffice); for one of the kept vertices, we "remember" that deletion would mean to decrease k by the size of its false twin class minus the number of its other representatives that were kept. Let X' denote the set we have thus attained from X.

5. **Updating M.** Let $M^\star = M \cup X'$. Then, $|M^\star| \in \mathcal{O}(p^7 \cdot k)$. If $V(G) = M^\star \cup X'$, then from Proposition 3.10, we already know that the treewidth of (the current) G is some $w \in \mathcal{O}(\sqrt{p \cdot |M^\star|}) \subseteq \mathcal{O}(p^4 \cdot \sqrt{k})$, and so we are done (by using dynamic programming over a tree decomposition of width $\mathcal{O}(w)$). Else, we proceed to the following steps. Possibly, we also insert into $M^\star$ an approximate solution for the instance of the problem Π at hand.

[1] Here, we use the fact that on disk graphs, one can a efficiently find a clique of size at least half than the size of a maximum clique.

6. **Bounding the Treewidth of $G - M^\star$ or $(G - M^\star)/Q$.** We compute a set $Q \subseteq V(G)$ so that $(G - M)/Q$ has "'small" treewidth w', in particular, being at most $\mathcal{O}(k^{1-\epsilon})$ for some fixed $\epsilon > 0$. (More generally, we compute a collection of such sets, so that the contraction of none of them turns a no-instance into a yes-instance, and the contraction of at least one of them does not turn a yes-instance into a no-instance.) Possibly, some additional preprocessing/marking of $(G - M)/Q$ is required so that it is indeed "valid" to contract Q.

7. **Application of Theorem 2.** From Theorem 2 (whose usage is valid due to the first two steps in this scheme), it follows that the treewidth of (the current) G is some $w \in \mathcal{O}(\max\{p^{2.5} \cdot \sqrt{|M^\star| \cdot w'}, p \cdot w'\}) \subseteq \mathcal{O}(p^7 \cdot \sqrt{k \cdot w'})$, and so we are done (by using dynamic programming over a tree decomposition of width $\mathcal{O}(w)$).

Applications. We now proceed to briefly consider the ways in which this scheme is applied to Triangle Hitting, Feedback Vertex Set and Odd Cycle Transversal. We start with Triangle Hitting, where the application is the simplest. In Step 3, we mark all edges outside M as irrelevant—indeed, because M contains a triangle hitting set and for all $v \in M$, $N_G(v) \setminus M$ is an independent set, there exists no triangle in G that contains at least two vertices from $V(G) \setminus M$. Next, in Step 4, we can keep only one vertex from each false twin class (updating its weight to be the sum of the weights of the vertices in the class) since, for Triangle Hitting, it can be shown that every minimal solution either contains all vertices in the class, or none of them. After this, the algorithm terminates in the next step (as $V(G) = M^\star \cup X'$). Overall, by choosing p appropriately, we attain a running time of $2^{\mathcal{O}(k^{\frac{9}{10}} \log k)} \cdot n^{\mathcal{O}(1)}$.

For Feedback Vertex Set and Odd Cycle Transversal, we do not mark any edge as irrelevant in Step 3. For Feedback Vertex Set, in Step 4, we can keep only two vertices (one of which having maximum weight) from each false twin class (updating the weight of the other one of them to be the sum of the weights of the vertices in the class excluding the other kept vertex) since, for Feedback Vertex Set, it can be shown that every minimal solution either contains all but at most one of the vertices in the class, or none of them. By inserting a feedback vertex set (of size at most $2k$, else we can conclude that we have a no-instance) into $M^\star$, we directly obtain that $G - M^\star$ has treewidth $w' \leq 1$ (so, we pick $Q = \emptyset$). Overall, by choosing p appropriately, we attain a running time of $2^{\mathcal{O}(k^{\frac{13}{14}} \log k)} \cdot n^{\mathcal{O}(1)}$.

For Odd Cycle Transversal, in Step 4, we can keep only one vertex from each false twin class since it can be shown that every minimal solution either contains all vertices in the class, or none of them. Next, we do not further update $M^\star$ as in the case of Feedback Vertex Set. Instead, we observe that since $G - M^\star$ is triangle-free disk graph, it follows that it is a planar graph. We use a known proposition (Proposition 6.24) to find a set $S^\star$ of size polynomial in k that must contain a solution (where we only consider the unweighted version of the problem), if one exists. The algorithm in this proposition is randomized, and we remark that due to its use, we are able to solve neither the weighted nor the counting version of the problem. In addition, we also use a known proposition (Proposition 6.27) to attain $t = \lceil \sqrt{k} \rceil$ sets of vertices in $G - M^\star$, $Z_1, \ldots, Z_t$, so that for any Z_i and any $Z' \subseteq Z_i$, $(G - M^\star)/(Z_i \setminus Z')$ has treewidth at most $\mathcal{O}(\sqrt{k} + |Z'|)$. We iterate over every choice of i, and every choice of $Z' \subseteq S^\star \cap Z_i$ of size at most $\sqrt{k}$; we note that there are only $\mathcal{O}(2^{\sqrt{k} \log k})$ such choices. In particular, by the pigeon-hole principle (and the choice of $S^\star$), there will be an iteration where the Z' under consideration will be precisely the intersection of some solution (if one exists) with Z_i. In that iteration, we will be able to identify a solution. We remark that when contracting a set Q (of the form $Z_i \setminus Z'$), we also add labels on the edges of the derived graph, to keep information regarding parities of paths through the contracted components. Overall, by choosing p appropriately, we attain a (randomized algorithm with a) running time of $2^{\mathcal{O}(k^{\frac{27}{28}} \log k)} \cdot n^{\mathcal{O}(1)}$.

Counting versions. For counting versions (of Triangle Hitting and Feedback Vertex Set), we observe that the first two steps in our scheme are relevant once we enable marking vertices as "undeletable"; if we work with weights, this can be just encoded by assigning a high enough weight to each undeletable vertex. Specifically, then the number of solutions is the sum of the number of solutions in each of the produced instances. The only other non-trivial modification concerns Step 4 in the scheme. Of course, when dealing with counting problems, our claims regarding minimal solutions still hold. Still, if one wants to eliminate false twins, it is required to remember how many vertices each kept vertex represents and make (standard) use of this information in the later dynamic programming procedure. Alternatively, we can just not eliminate them (and hence have the aforementioned information available), but then work with a tree decomposition of the graph where they are eliminated.

3 Preliminaries

Let $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. For a function $f : A \to B$ and a subset $A' \subseteq A$, we let $f|_{A'}$ denote the restriction of f to A', that is, $f|_{A'} : A' \to B$ where $f|_{A'}(a) = f(a)$ for all $a \in A'$.

Graphs. Whenever it is not stated otherwise, we deal only with simple undirected graphs. Let G be a graph. Let $V(G)$ and $E(G)$ denote the vertex set and edge set of G, respectively. Given $e = \{u, v\} \in E(G)$, the *contraction* of e in G, denoted by G/e, is the replacement of u and v by a new vertex adjacent to every vertex (excluding u and v) that was adjacent to at least one vertex among u and v in G. The *subdivision* of e in G is the replacement of e by a new vertex w and the edges $\{u, w\}$ and $\{v, w\}$. Let $U \subseteq V(G)$. Then, $G[U]$ denotes the subgraph induced by U, that is, $V(G[U]) = U$ and $E(G[U]) = \{\{u, v\} \in E(G) : u, v \in U\}$. We say that U is a *connected set* if $G[U]$ is a connected graph. Additionally, $G - U$ denotes the graph obtained from G by deleting all vertices in U (along with all edges incident to them). Lastly, G/U is defined as follows. First, $V(G/U) = (V(G) \setminus U) \cup \{v_C : C \text{ is a connected component of } G[U]\}$. Now, for every vertex $v \in V(G/U)$, if $v \in V(G)$, then $\mathsf{uncontract}_G(U, v) = \{v\}$, and otherwise $v = v_C$ for some connected component C of $G[U]$, and then $\mathsf{uncontract}_G(U, v) = V(C)$. Then, $E(G/U) = \{\{u, v\} : u, v \in V(G/U)$, there exist $u' \in \mathsf{uncontract}_G(U, u)$ and $v' \in \mathsf{uncontract}_G(U, v)$ such that $\{u', v'\} \in E(G)\}$. Given a subset $E' \subseteq E(G)$, $G - E'$ denotes the graph on vertex set $V(G)$ and edge set $E(G) \setminus E'$, and $V(E')$ denotes the set of vertices incident on at least one edge from E'. Given $u, v \in V(G)$, the *(graph) distance* between u and v, denoted by $\mathsf{distance}_G(u, v)$, is the length (number of edges) of a shortest path in G between u and v (which is defined to be ∞ if there is no path between u and v in G). Further, define $\mathsf{distance}_G(u, U) = \min_{v \in U} \mathsf{distance}_G(u, v)$. We let $N_G(v)$ denote the set of neighbors of v in G.

A *triangle* is a cycle on three vertices. When we say that a graph G has a triangle (or any other graph), we mean that it admits that graph as a subgraph. A graph is *triangle-free* if it does not have any triangle. A *triangle-free graph class* is a (possibly infinite) class of graphs such that no graph in it has a triangle. Throughout this paper, the term *path* refers to a simple path, while the term *walk* is used to allow vertex repetitions. Similarly, the term *cycle* refers to a simple cycle, while the term *closed walk* is used to allow vertex repetitions. An *odd (resp. even) path* is a path with an odd (resp. even) number of edges. The terms *odd cycle*, *even cycle*, *odd walk* and *even walk* are defined analogously. We have the following immediate proposition.

Proposition 3.1 (Folklore)**.** *Let G be a graph. Then, G has an odd cycle if and only if it has an odd closed walk.*

A *clique* is a graph such that every pair of vertices in it are adjacent. A *matching* is a collection of pairwise-disjoint edges (i.e., no two edges in the collection share an endpoint vertex). We say that matching $E' \subseteq E(G)$ is *maximal* (in G) if every edge in $E(G) \setminus E'$ shares an endpoint vertex with at least one edge in E'. A subset $U \subseteq V(G)$ is an *independent set* (in G) if there do not exist two adjacent vertices in U. A graph G is *bipartite* if there exists a partition (A, B) of $V(G)$ such that A is an independent set in G, and B is an independent set in G. We have the following well-known proposition.

Proposition 3.2 (See, e.g., [18])**.** *A graph G has no odd cycle if and only if it is bipartite.*

A *triangle hitting set* of G is a subset $U \subseteq V(G)$ such that $G - U$ is triangle-free. A *feedback vertex set* of G is a subset $U \subseteq V(G)$ such that $G - U$ is a forest (i.e., acyclic). An *odd cycle transversal* of

G is a subset $U \subseteq V(G)$ such that $G - U$ has no odd cycle. Two vertices $u, v \in V(G)$ are *false twins* if $N_G(u) = N_G(v)$ (which implies that $\{u, v\} \notin E(G)$). The *false twin partition* of G is the partition of $V(G)$ such that two vertices $u, v \in V(G)$ are in the same part if and only if u and v are false twins. The *false twin reduction* of G, denoted by $\mathsf{FTR}(G)$, is the graph obtained from G by removing, for each part in the false twin partition of G, all vertices from that part except for one (it is immaterial which one). Two graphs G and H are *isomorphic* if there exists a bijective function $f : V(G) \to V(H)$, called an *isomorphism*, such that $\{u, v\} \in E(G)$ if and only if $\{f(u), f(v)\} \in E(H)$.

The treewidth of a graph is a standard measure of its "closeness" to a tree, defined as follows.

Definition 3.3. *A* tree decomposition *of a graph G is a pair (T, β), where T is a rooted tree and β is a function from $V(T)$ to $2^{V(G)}$, that satisfies the following conditions.*

- *For every edge $\{u, v\} \in E(G)$, there exists $x \in V(T)$ such that $\{u, v\} \subseteq \beta(x)$.*
- *For every vertex $v \in V(G)$, $T[\{x \in V(T) : v \in \beta(x)\}]$ is a tree on at least one vertex.*

The width *of (T, β) is $\max_{x \in V(T)} |\beta(x)| - 1$. The* treewidth *of G, denoted by $\mathtt{tw}(G)$, is the minimum width over all tree decompositions of G. For every $x \in V(T)$, $\beta(x)$ is called a* bag. *Further, $\gamma(x)$ is the union of the bags of x and its descendants.*

For any $t \in \mathbb{N}$, a graph G is a $t \times t$ *grid* if its vertex set is of size t^2 and can be denoted as $V(G) = \{v_{i,j} : i, j \in \{1, 2, \ldots, t\}\}$ so that $E(G) = \{\{v_{i,j}, v_{x,y}\} : |i - x| + |j - y| = 1\}$. We have the following proposition.

Proposition 3.4 (See, e.g., [10]). *The treewidth of a $t \times t$ grid is t.*

A graph H is a *minor* of G if H can be obtained from G by a sequence of edge deletions, edge contractions and vertex deletions. Alternatively (and equivalently), H is a minor of G if there exists a function $\varphi : V(H) \to 2^{V(G)} \setminus \{\emptyset\}$ such that for for every $u \in V(G)$, $G[\varphi(u)]$ is connected, for every $u, v \in V(G)$, $\varphi(u) \cap \varphi(v) = \emptyset$, and for every $\{u, v\} \in E(H)$, there exist $u' \in \varphi(u)$ and $v' \in \varphi(v)$ such that $\{u', v'\} \in E(G)$. Such a function φ is called a *minor model*. We have the following proposition.

Proposition 3.5 (See, e.g., [10]). *Let G be a graph, and let H be a minor of G. Then, the treewidth of G is at least as large as the treewidth of H.*

Problem Definitions. In the Triangle Hitting problem, we are given a graph G and an integer $k \in \mathbb{N}_0$, and a *solution* is a subset $S \subseteq V(G)$ of size at most k that is a triangle hitting set of G. It is *minimal* if for all $v \in S$, $S \setminus \{v\}$ is not a solution. The objective is to decide if there exists a solution. In all version of Triangle Hitting (described immediately), the definition of a solution is unchanged. In the #Triangle Hitting problem, the input is the same, and the objective is to determine the number of solutions. In the Weighted Triangle Hitting problem, the input also includes a vertex-weight function $\mathfrak{w} : V(G) \to \mathbb{N}_0$ and a weight $W \in \mathbb{N}_0$, and the objective is to determine whether there exists a solution whose weight (being the sum of the weights of its vertices) is at most W. In the #Weighted Triangle Hitting problem, the input also includes a vertex-weight function $\mathfrak{w} : V(G) \to \mathbb{N}_0$ (but it does not include a weight $W \in \mathbb{N}_0$), and the objective is to determine the number of solutions having minimum weight as well as to return this minimum weight. Observe that #Weighted Triangle Hitting generalizes all the other three versions. Indeed, given an instance of #Triangle Hitting, by assigning weight 0 to every vertex, we obtain an equivalent instance of #Weighted Triangle Hitting, and #Triangle Hitting, in turn, generalizes Triangle Hitting.[2] Further, given an instance of Weighted Triangle Hitting (which, in turn, also generalizes Triangle Hitting), we may solve it as if it was an instance of #Weighted Triangle Hitting, and determine that we have a yes-instance if and only if the weight returned is at most W (with a positive number of solutions).

In $\mathcal{F}$-Vertex Deletion (where $\mathcal{F}$ is some class of graphs), Feedback Vertex Set and Odd Cycle Transversal, the input is the same as in Triangle Hitting. However, in $\mathcal{F}$-Vertex

[2]Given as input an instance of Triangle Hitting and the output for this input as an instance of #Triangle Hitting, we just need to check whether it is 0 or at least 1.

Deletion, a solution is a subset $S \subseteq V(G)$ of size at most k such that $G - S \in \mathcal{F}$. In Feedback Vertex Set, a solution is a subset $S \subseteq V(G)$ of size at most k that is a feedback vertex set. In Odd Cycle Transversal, a solution is a subset $S \subseteq V(G)$ of size at most k that is an odd cycle transversal. In all of these problems, the objective is to determine whether a solution exists. Observe that Triangle Hitting, Feedback Vertex Set and Odd Cycle Transversal are $\mathcal{F}$-Vertex Deletion where $\mathcal{F}$ is the class of triangle-free graphs, forests and bipartite graphs, respectively. The generalizations of all of these problems to their counting versions, their weighted versions, and their combined counting and weighted versions, are done analogously to the above for Triangle Hitting, and the combined counting and weighted versions always generalize each of the other versions.

Planar Graphs. A graph G is *planar* if there exists a mapping from every vertex in $V(G)$ to a point on the plane, and from every edge $e \in E(G)$ to a curve on the plane where the extreme points of the curve are the points mapped to the endpoints of e, and all curves are disjoint except on their extreme points. Such a mapping is called an *embedding in the plane*, or simply an *embedding*. A *plane graph* is a planar graph having a fixed embedding. For a planar graph G, we can define its faces as follows: delete all the edges and vertices of G from the plane. Then, the remaining part of the plane is a collection of disjoint areas. Each such area is called a *face*. The face whose area is unbounded is called the *outer-face*, and every other face is an *interior face*. The *dual graph* D of a plane graph G is the embedded graph whose vertex set includes a vertex v_F for each face F of G, and which contains an edge between v_F and $v_{F'}$ if the faces F and F' are adjacent (i.e., their boundaries share an edge) in G. Note that even if G is a simple graph, its dual D may contain multiedges. Moreover, the dual of a plane graph is a planar graph in itself. The distance between two faces F and F' of G, denoted by $\mathsf{distance}_G(F, F')$, is the (graph) distance between the vertices that represent these faces in the dual D of G. Correspondingly, for a set of faces $\mathcal{F}$, define $\mathsf{distance}_G(F, \mathcal{F}) = \min_{F' \in \mathcal{F}} \mathsf{distance}_G(F, F')$.

Proposition 3.6 (Consequence of Euler's Formula)**.** *Let G be a plane graph with f faces. Then, $f \leq \mathcal{O}(|E(G)|)$.*

Proposition 3.7 ([26])**.** *Let G be a planar graph of treewidth w. Then, G contains a $\lceil \frac{w}{5} \rceil \times \lceil \frac{w}{5} \rceil$ grid as a minor.*

Geometric and Disk Graphs. Let $\mathcal{O}$ be a collection of geometric objects. For the sake of clarity, in all figures in this paper, we will only illustrate the boundaries of objects of interest (e.g., see Figure 3). Then, the *geometric intersection graph* of $\mathcal{O}$ is the graph G with vertex set $V(G) = \{v_O : O \in \mathcal{O}\}$ (i.e., there is an implicit bijection between the objects and the vertices), where two vertices are adjacent if and only if their corresponding objects intersect (in at least one point). Here, the object that a vertex v represents is denoted by $\mathsf{obj}_{\mathcal{O}}(v)$. Further, for a subset $U \subseteq V(G)$, let $\mathsf{obj}_{\mathcal{O}}(U) = \{\mathsf{obj}_{\mathcal{O}}(v) : v \in U\}$. The set $\mathcal{O}$ is called a *realization* of G. A graph G that admits a realization where all the objects are disks (being closed disks on the Euclidean plane) is termed a *disk graph*. When we deal with a disk graph, we use the term realization to refer only to a realization where all objects are disks. To further stress this, for a disk graph with a realization $\mathcal{O}$, we use the notation $\mathsf{disk}_{\mathcal{O}}(\cdot)$ rather than $\mathsf{obj}_{\mathcal{O}}(\cdot)$. A graph G that admits a realization where all the objects are closed disks on the Euclidean plane that have the same radius is termed a *unit disk graph*. We remark all of our algorithms, when given disk graphs as input, are not assumed to also be given realizations of these graphs as disk graphs. This is considered to be a "sought after" property (see, e.g., [8]), particularly because determining whether a given graph is a disk graph (and hence obtaining a realization) is NP-hard [27]. We say that a collection of disks $\mathcal{D}$ is in *general position* if *(i)* no two disks in $\mathcal{D}$ are identical or intersect in exactly one point, and *(ii)* there do not exist four distinct points such that each of these points belongs to the boundary of a different disk in $\mathcal{D}$, and such that there exists a circle that goes through these four points. By slightly perturbing the radii of disks in a realization of a disk graph G, we have the following.

Proposition 3.8 (Folklore)**.** *Every disk graph G has a realization where the disks are in general position.*

The *ply* of a realization $\mathcal{D}$ of a disk graph G is the maximum number of disks in $\mathcal{D}$ that intersect in the same point. Observe that the vertices representing disks that intersect in the same point must be pairwise-adjacent (i.e. form a clique) in G. This directly yields the following.

Proposition 3.9 (Folklore)**.** *Let G be a disk graph that has not clique of size larger than p. Then, the ply of any realization of G is at most p.*

Concerning the treewidth of disk graphs, we have the following.

Proposition 3.10 ([39])**.** *Let G be a disk graph that has a realization of ply p. Then, the treewidth of G is $\mathcal{O}(\sqrt{|V(G)|p})$.*

Let G be a geometric graph with a realization $\mathcal{O}$. The *graph drawn by* $\mathcal{O}$ is the plane (multi)graph whose vertex set consists of: one vertex on every intersection point of the boundaries of at least two disks, and one vertex on some point on the boundary of every disk that does not intersect any other disk.[3] The edges are naturally defined by the parts of the boundary that connect the vertices. The *faces of* $\mathcal{O}$ are the faces of the graph drawn by $\mathcal{O}$. The *complexity* of $\mathcal{O}$ is the number of vertices and edges in the graph drawn by $\mathcal{O}$. The distance between faces F and F' of $\mathcal{O}$, denoted by $\mathsf{distance}_{\mathcal{O}}(F, F')$, is the distance between them in the graph drawn by $\mathcal{O}$. Correspondingly, for a set of faces $\mathcal{F}$, define $\mathsf{distance}_{\mathcal{O}}(F, \mathcal{F}) = \min_{F' \in \mathcal{F}} \mathsf{distance}_{\mathcal{O}}(F, F')$. The *arrangement graph* of $\mathcal{O}$ is the dual of the graph drawn by $\mathcal{O}$ with all vertices representing faces that are not contained in any object in $\mathcal{O}$ being removed. Thus, the arrangement graph is a planar graph.

Note that in a disk graph G with a realization $\mathcal{D}$ of ply p, every disk D can intersect only $\mathcal{O}(p)$ disks that are at least as large as D. This means that the number of edges (and hence also of vertices) in the graph drawn by $\mathcal{D}$ is $\mathcal{O}(p \cdot |V(G)|)$. Due to Proposition 3.6, this yields the following.

Proposition 3.11 (Folklore)**.** *Let G be a disk graph that has a realization of ply p. Then, the number of vertices in the corresponding arrangement graph is $\mathcal{O}(p \cdot |V(G)|)$.*

Let G be a geometric graph with a realization $\mathcal{O}$. Given a tree decomposition of the arrangement graph of $\mathcal{O}$, consider the replacement of every vertex v (of the arrangement graph) in every bag $\beta(x)$ where it occurs by the set of vertices in G whose objects (in $\mathcal{O}$) contain the face represented by v. It is easy to see that this yields a tree decomposition for G. In particular, this yields the following.

Proposition 3.12 (Folklore; see also [39])**.** *Let G be a geometric (not necessarily disk) graph that has a realization of ply p whose arrangement graph has treewidth w. Then, the treewidth of G is $\mathcal{O}(w \cdot p)$.*

4 Proof of Theorem 1

Towards the proof of Theorem 1, we start with a simple observation.

Observation 4.1. *Let $\mathcal{D}$ be a collection of disks. Suppose that each disk in $\mathcal{D}$ has radius at least r and the Euclidean distance between the closest point in it and (x, y) is at most d, for some $r, x, y, d \in \mathbb{R}$. Let $\mathcal{P} = \{(x + i \cdot r, y + j \cdot r) : i, j \in \{-\lceil d/r \rceil, -\lceil d/r \rceil + 1, \ldots, \lceil d/r \rceil\}\}$. Then, every disk in $\mathcal{D}$ intersects at least one point in $\mathcal{P}$.*

For the proof of Theorem 1, we will "charge" each disk in $V(G) \setminus M$ to the smallest disk in M that it intersects. So, we will need to analyze the charge of each disk in M. For this purpose, based on Observation 4.1, we prove the following.

Lemma 4.2. *Let G be a disk graph with a realization $\mathcal{D}$ of ply p, and let $M \subseteq V(G)$. Let $v \in M$. Let A_v be the set of all vertices $u \in V(G) \setminus M$ such that $v \in N(u)$ and all disks in $\mathsf{disk}_{\mathcal{D}}(N(u))$ are at least as large as $\mathsf{disk}_{\mathcal{D}}(v)$. Let A_v^{large} be the set of vertices in A_v whose disks in $\mathcal{D}$ are at least as large as $\mathsf{disk}_{\mathcal{D}}(v)$. Then, $|A_v^{\mathsf{large}}| \in \mathcal{O}(p)$.*

[3]Observe that the graph drawn by $\mathcal{O}$ is unique up to the precise placement of the vertices on the boundary of every disk that does not intersect any other disk.

Proof. Let r be the radius of $\mathsf{disk}_{\mathcal{D}}(v)$, and let (x, y) be its centre. Then, all disks in $\mathsf{disk}_{\mathcal{D}}(A_v^{\mathsf{large}})$ have radius at least r. Moreover, because all of these disks intersect $\mathsf{disk}_{\mathcal{D}}(v)$, the Euclidean distance between (x, y) and the closest point to (x, y) in each of them is at most r, which we denote by d. Let $\mathcal{P} = \{(x + i \cdot r, y + j \cdot r) : i, j \in \{-\lceil d/r \rceil, -d/r + 1, \ldots, \lceil d/r \rceil\}\}$. Then, $|\mathcal{P}| \in \mathcal{O}(1)$. So, because the ply of $\mathcal{D}$ is p, by Observation 4.1 it follows that $|A_v^{\mathsf{large}}| \in \mathcal{O}(p)$. $\square$

Having analyzed the charge given to $v \in M$ by disks at least as large as its own disk, we turn to consider the charge given to v by disks smaller than its own disk. Based on Observation 4.1 yet again, we first prove the following.

Lemma 4.3. *Let G be a disk graph with a realization $\mathcal{D}$ of ply p, and let $M \subseteq V(G)$. Let $v \in M$. Let A_v be the set of all vertices $u \in V(G) \setminus M$ such that $v \in N(u)$ and all disks in $\mathsf{disk}_{\mathcal{D}}(N(u))$ are at least as large as $\mathsf{disk}_{\mathcal{D}}(v)$. Let A_v^{small} be the set of vertices in A_v whose disks in $\mathcal{D}$ are smaller than $\mathsf{disk}_{\mathcal{D}}(v)$. Let $B \subseteq M$ be the set of vertices in M whose disks in $\mathcal{D}$ have non-empty intersection with at least one disk in $\mathcal{D}$ whose vertex is in A_v^{small}. Then, $|B| \in \mathcal{O}(p)$.*

Proof. Let r be the radius of $\mathsf{disk}_{\mathcal{D}}(v)$, and let (x, y) be its centre. Due to the definition of A_v, for every $u \in A_v$, all disks in $\mathsf{disk}_{\mathcal{D}}(N(u))$ have radius at least r. Thus, by the definition of B, all disks in $\mathsf{disk}_{\mathcal{D}}(B)$ have radius at least r. Moreover, by the definitions of A_v^{small} and B, for every disk D in $\mathsf{disk}_{\mathcal{D}}(B)$, there exists a disk of radius smaller than r (which belongs to $\mathsf{disk}_{\mathcal{D}}(A_v^{\mathsf{small}})$) that has non-empty intersection with both D and $\mathsf{disk}_{\mathcal{D}}(v)$. Thus, for every disk in $\mathsf{disk}_{\mathcal{D}}(B)$, the distance between (x, y) and the closest point to (x, y) in it is at most $3r$, which we denote by d. Let $\mathcal{P} = \{(x + i \cdot r, y + j \cdot r) : i, j \in \{-\lceil d/r \rceil, -d/r + 1, \ldots, \lceil d/r \rceil\}\}$. Then, $|\mathcal{P}| \in \mathcal{O}(1)$. So, because the ply of $\mathcal{D}$ is p, by Observation 4.1 it follows that $|B| \in \mathcal{O}(p)$. $\square$

We proceed with an observation and a lemma that we will afterwards use to analyze the aforementioned small disks while making use of the bound on $|B|$ given by Lemma 4.3. First, from the fact that any two distinct circles can intersect in at most two points, we derive the following.

Observation 4.4. *Let G be a disk graph. Then, the complexity of any realization of G is $\mathcal{O}(|V(G)|^2)$.*

Additionally, we prove the following.

Lemma 4.5. *Let G be a disk graph, and let $B \subseteq V(G)$. Let $Q \subseteq V(G) \setminus B$ such that for all distinct $u, v \in Q$, $N(u) \cap B \neq N(v) \cap B$. Then, $|Q| \in \mathcal{O}(|B|^6)$.*

Proof. By Proposition 3.8, without loss of generality, we can consider a realization $\mathcal{D}$ in general position. Let x denote the complexity of $\mathsf{disk}_{\mathcal{D}}(B)$. By Observation 4.4, $x \in \mathcal{O}(|B|^2)$.

We first observe that for any disk $D \in \mathsf{disk}_{\mathcal{D}}(Q)$, we can change D continuously without changing the set of disks in $\mathsf{disk}_{\mathcal{D}}(B)$ that it intersects to make D either a point (disk of radius 0) or externally tangent to two disks in $\mathsf{disk}_{\mathcal{D}}(B)$. To see this, first fix the centre of D and then continue shrinking D, until the moment the set of disks in $\mathsf{disk}_{\mathcal{D}}(B)$ that it intersects is about to change or D becomes a point. If D becomes a point, then we are done. Else, at the moment when the set of disks in $\mathsf{disk}_{\mathcal{D}}(B)$ that D intersects is about to change, D must be externally tangent to a disk A in $\mathsf{disk}_{\mathcal{D}}(B)$. Next, we fix the tangent point of D and A and accordingly shrink D (so in the shrinking process, D is always tangent to A). Again we do this until the moment the set of disks in $\mathsf{disk}_{\mathcal{D}}(B)$ that D intersects is about to change or D becomes a point. When we terminate, either D becomes a point or there is another disk A' in $\mathsf{disk}_{\mathcal{D}}(B)$ that is tangent to D.

If at the end of the last step, D is externally tangent to A and A' in $\mathsf{disk}_{\mathcal{D}}(B)$, we further change D to make it tangent to another disk A'' in $\mathsf{disk}_{\mathcal{D}}(B)$ or become the smallest disk tangent to A and A'. For this purpose, we continuously change D while keeping it tangent to A and A'. There are two directions to do this: one direction increases the size of D and the other decreases the size of D—we choose the latter. We do this until D becomes tangent to another disk A'' in $\mathsf{disk}_{\mathcal{D}}(B)$ or it becomes the smallest disk externally tangent to A and A'. Observe that (i) a disk in $\mathsf{disk}_{\mathcal{D}}(B)$ that intersects D at the beginning always intersects D during the entire procedure, and (ii) a disk in $\mathsf{disk}_{\mathcal{D}}(B)$ that is disjoint from D at the beginning may intersect (but only be externally tangent to) D at the end. Therefore, if we use $D^\star$ to denote the disk after the change, then it satisfies the following property with respect to D, which we refer to as *D-compatibility*: the set of disks in $\mathsf{disk}_\mathcal{D}(B)$ that intersect D is a subset of the set of disks in $\mathsf{disk}_\mathcal{D}(B)$ that intersect $D^\star$, and every disk in $\mathsf{disk}_\mathcal{D}(B)$ that intersects $D^\star$ but not D, intersects $D^\star$ in a single point.

To summarize, for any disk D in $\mathsf{disk}_\mathcal{D}(Q)$, either it intersects the same set of disks in $\mathsf{disk}_\mathcal{D}(B)$ as some point P, or we can find a disk $D^\star$, which is externally tangent to three disks in $\mathsf{disk}_\mathcal{D}(B)$ or is the smallest disk externally tangent to two disks in $\mathsf{disk}_\mathcal{D}(B)$, such that $D^\star$ is D-compatible. In the first case, the number of possible disks with different neighbourhoods in B is $O(x)$ because the number of choices of points is bounded from above by the number of vertices and intersection points in the intersection graph of $\mathsf{disk}_\mathcal{D}(B)$ (which is $\mathcal{O}(x)$), plus the number of faces in this graph. By Proposition 3.6, the latter is also bounded by $\mathcal{O}(x)$. In the second case, there are $O(x^3)$ possible disks $D^\star$, corresponding to choosing the intersection points with A, A' and possibly also A'' (corresponding to choosing three vertices or edges from the intersection graph). For each such disk $D^\star$, due to its D-compatibility and the general position property of $\mathcal{D}$, only A, A' and A'' can be intersected by $D^\star$ but not D, which gives rise for $2^3 = 8$ additional possibilities to consider (that is, choosing whether the intersection point of $D^\star$ with each of the disks A, A' and A'' should be taken into account when determining the neighborhood of D). Thus, the size of Q is bounded by $\mathcal{O}(x + 8 \cdot x^3) \subseteq \mathcal{O}(|B|^6)$. $\square$

Having Lemmata 4.3 and 4.5, we can prove the following.

Lemma 4.6. *Let G be a disk graph with a realization $\mathcal{D}$ of ply p, and let $M \subseteq V(G)$. Let $v \in M$. Let A_v be the set of all vertices $u \in V(G) \setminus M$ such that $v \in N(u)$ and all disks in $\mathsf{disk}_\mathcal{D}(N(u))$ are at least as large as $\mathsf{disk}_\mathcal{D}(v)$. Let A_v^{small} be the set of vertices in A_v whose disks in $\mathcal{D}$ are smaller than $\mathsf{disk}_\mathcal{D}(v)$. Let $Q \subseteq A_v^{\mathsf{small}}$ such that for all distinct $u, v \in Q$, $N(u) \cap M \neq N(v) \cap M$. Then, $|Q| \in \mathcal{O}(p^6)$.*

Proof. Let $B \subseteq M$ be the set of vertices in M whose disks in $\mathcal{D}$ have non-empty intersection with at least one disk in $\mathcal{D}$ whose vertex is in A_v^{small}. By Lemma 4.3, $|B| \in \mathcal{O}(p)$. Note that $Q \subseteq V(G) \setminus B$, and by the supposition of the lemma, for all distinct $u, v \in Q$, $N(u) \cap M \neq N(v) \cap M$. Thus, by Lemma 4.5, $|Q| \in \mathcal{O}(|B|^6) \subseteq \mathcal{O}(p^6)$. $\square$

Combining Lemmata 4.2 and 4.6, we immediately derive the following.

Corollary 4.7. *Let G be a disk graph with a realization $\mathcal{D}$ of ply p, and let $M \subseteq V(G)$. Let $v \in M$. Let A_v be the set of all vertices $u \in V(G) \setminus M$ such that $v \in N(u)$ and all disks in $\mathsf{disk}_\mathcal{D}(N(u))$ are at least as large as $\mathsf{disk}_\mathcal{D}(v)$. Let $Q \subseteq A_v$ such that for all distinct $u, v \in Q$, $N(u) \cap M \neq N(v) \cap M$. Then, $|Q| \in \mathcal{O}(p^6)$.*

Finally, having Corollary 4.7 at hand, we are ready the prove Theorem 1.

Theorem 1. *Let G be a disk graph that has some realization of ply p, and let $M \subseteq V(G)$. Let $U \subseteq V(G) \setminus M$ be such that for all distinct $u, v \in U$, $N_G(u) \cap M \neq N_G(v) \cap M$. Then, $|U| \in \mathcal{O}(|M| \cdot p^6)$.*

Proof. Let $\mathcal{D}$ be a realization of ply p of G. For every $v \in M$, let A_v be the set of all vertices $u \in V(G) \setminus M$ such that $v \in N(u)$ and all disks in $\mathsf{disk}_\mathcal{D}(N(u))$ are at least as large as $\mathsf{disk}_\mathcal{D}(v)$. Let $U_v = U \cap A_v$. Let U_{nil} be the set of vertices in U with no neighbors from M. Observe that $U = (\bigcup_{v \in M} U_v) \cup U_{\mathsf{nil}}$. Recall that for all distinct $u, v \in Q$, $N(u) \cap M \neq N(v) \cap M$. Thus, $|U_{\mathsf{nil}}| \leq 1$, and from Corollary 4.7, $|U_v| \in \mathcal{O}(p^6)$ for every $v \in M$. Thus, $|U| \in |M| \cdot \mathcal{O}(p^6) + 1 \subseteq \mathcal{O}(|M| \cdot p^6)$. $\square$

5 Proof of Theorem 2

The main ingredient in the proof of Theorem 2 is to analyze the faces of $\mathcal{D}$. Here, the key players will be *relevant faces*, defined as follows.

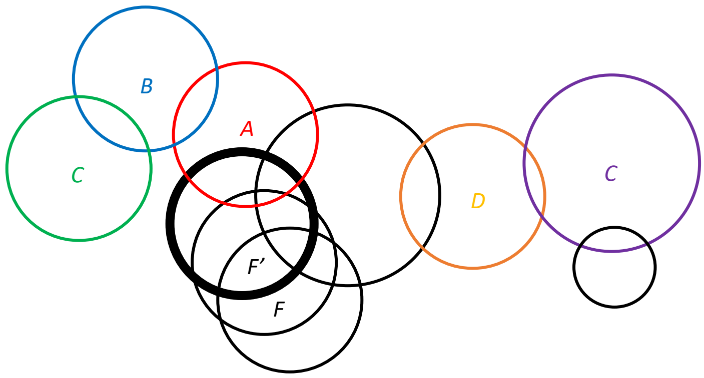

Figure 3: (I) The cases in the proof of Observation 5.2. The set of black (without letters) disks is $\mathsf{disk}_{\mathcal{D}}(M)$. All of the faces contained in disks B and C, as well as the face corresponding to the intersection of D and C, correspond to case (i). Disk A has one face not contained in disk B, which is the union of one irrelevant face and five relevant faces. Disks C and D each has a face not contained in the other, which is the union of one irrelevant face and one relevant face. (II) The faces F and F' in Observation 5.4. Here $\mathcal{D}_b$ includes exactly one disk, which is the black disk with a thicker boundary.

Definition 5.1. *Let G be a disk graph with a realization $\mathcal{D}$, and let $M \subseteq V(G)$. Let F be a face in the arrangement graph of $\mathcal{D}$. Then, F is* relevant *with respect to M if it is contained in at least one disk in $\mathsf{disk}_{\mathcal{D}}(M)$, and otherwise it is* irrelevant *with respect to M. When M is clear from context, we drop the phrase "with respect to M". The set of all relevant faces is denoted by $\mathsf{relevant}(\mathcal{D}, M)$.*

To bound the treewidth of G, we will attempt to bound the treewidth of the arrangement graph of a realization of G. Towards that, we will analyze the arrangement graph of $G - M$. We first observe the relevance of relevant faces to the analysis of the relations between these two arrangement graphs in the following.

Observation 5.2. *Let G be a disk graph with a realization $\mathcal{D}$, and let $M \subseteq V(G)$ such that for all $v \in M$, $N_G(v) \setminus M$ is an independent set, and no disk $\mathsf{disk}_{\mathcal{D}}(V(G) \setminus M)$ is contained in the union of disks in $\mathsf{disk}_{\mathcal{D}}(M)$. Let A and B be the arrangement graphs of $\mathcal{D}$ and $\mathcal{D} \setminus \mathsf{disk}_{\mathcal{D}}(M)$, respectively. Let U be the set of vertices in A that represent the faces in $\mathsf{relevant}(\mathcal{D}, M)$. Then, $A - U$ and B are isomorphic.*

Proof. To see this, consider the faces of $\mathcal{D} \setminus \mathsf{disk}_{\mathcal{D}}(M)$. Because for all $v \in M$, $N_G(v) \setminus M$ is an independent set, and no disk $\mathsf{disk}_{\mathcal{D}}(V(G) \setminus M)$ is contained in the union of disks in $\mathsf{disk}_{\mathcal{D}}(M)$, every face F of $\mathcal{D} \setminus \mathsf{disk}_{\mathcal{D}}(M)$ is either: (i) a face of $\mathcal{D}$ (in which case more than one disk in $\mathcal{D} \setminus \mathsf{disk}_{\mathcal{D}}(M)$ can contain it); (ii) the union of exactly one irrelevant face of $\mathcal{D}$ and at least one relevant face of $\mathcal{D}$, and in this case, F is contained in only one disk among those in $\mathcal{D} \setminus \mathsf{disk}_{\mathcal{D}}(M)$. So, we obtain an isomorphism by mapping a vertex in B representing a face F of $\mathcal{D} \setminus \mathsf{disk}_{\mathcal{D}}(M)$ to either the vertex representing the same face in $\mathcal{D}$ (in case (i)) or the vertex representing the unique irrelevant face that it contains from the faces of $\mathcal{D}$ (in case (ii)); see Fig. 3. □

To analyze relevant faces, we further identify within them so called *critical faces*. For their definition below, we will use the following terminology. Let G be a disk graph with a realization $\mathcal{D}$. Let F be a face in $\mathcal{D}$. Notice that every edge b in the boundary of F is either *concave* or *convex* with respect to F (see Fig. 2). When F is clear from context, we drop the phrase "with respect to F".

Definition 5.3. *Let G be a disk graph with a realization $\mathcal{D}$, and let $M \subseteq V(G)$. Let F be a face of $\mathcal{D}$. Then, F is* critical *with respect to M if F is relevant with respect to M, and for every edge b in the boundary of F: if b belongs to the boundary of a disk in $\mathsf{disk}_{\mathcal{D}}(M)$, then b is convex, and if b belongs* *to the boundary of a disk in* $\mathsf{disk}_{\mathcal{D}}(M)$*, then* b *is concave.*[4] *The set of all critical faces is denoted by* $\mathsf{critical}(\mathcal{D}, M)$*.*

The following observation is a direct consequence of the definition of convex and concave edges.

Observation 5.4. *Let* $\mathcal{D}$ *be a collection of disks. Let* F *and* F' *be two faces of* $\mathcal{D}$ *that share a boundary edge* b *which is concave with respect to* F *(and hence convex with respect to* F'*); see Fig. 3. Let* $\mathcal{D}_b$ *be the (identical) disks in* $\mathcal{D}$ *to which* b *belongs. Then, the set of disks in* $\mathcal{D}$ *that contain* F' *is the union of* $\mathcal{D}_b$ *with the set of disks in* $\mathcal{D}$ *that contain* F*. Moreover, no disk in* $\mathcal{D}_b$ *contains* F*.*

Having Observation 5.4 at hand, we prove the following.

Lemma 5.5. *Let* G *be a disk graph with a realization* $\mathcal{D}$ *in general position, and let* $M \subseteq V(G)$ *such that for all* $v \in M$*,* $N_G(v) \setminus M$ *is an independent set. Let* F *and* F' *be two faces of* $\mathcal{D}$ *that share a boundary edge* b *that belongs to a (unique)*[5] *disk* $A \in \mathcal{D} \setminus \mathsf{disk}_{\mathcal{D}}(M)$*. Suppose that* F *is relevant, and that* b *is convex with respect to* F *(and hence concave with respect to* F'*); see Fig. 2. Then,* F' *does not have a boundary edge that belongs to a disk in* $\mathcal{D} \setminus \mathsf{disk}_{\mathcal{D}}(M)$ *and which is convex with respect to* F'*.*

Proof. Targeting a contradiction, suppose F' has a boundary edge b' that belongs to a disk $B \in \mathcal{D} \setminus \mathsf{disk}_{\mathcal{D}}(M)$ and which is convex with respect to F'. Necessarily, $B \neq A$. Let F'' be the other face of $\mathcal{D}$ that shares the boundary edge b'. Let $\mathcal{D}_F$, $\mathcal{D}_{F'}$ and $\mathcal{D}_{F''}$ be the set of disks in $\mathcal{D}$ that contain F and F', respectively. By Observation 5.4, $\mathcal{D}_F = \mathcal{D}_{F'} \cup \{A\} = \mathcal{D}_{F''} \cup \{A, B\}$. Because F is relevant, there exists a disk, say, C, that belongs to $\mathcal{D}_F \cap \mathsf{disk}_{\mathcal{D}}(M)$. Let a, b and c be the vertices in $V(G)$ represented by A, B and C, respectively. Note that all disks in $\mathcal{D}_F$ intersect each other (in particular, all of them contain F), which means that a, b and c are neighbors of each other in G. However, $c \in M$ while $a, b \notin M$. This is a contradiction to the supposition that for all $v \in M$, $N_G(v) \setminus M$ is an independent set. □

In particular, we use Observation 5.4 and Lemma 5.5 to prove the following, which asserts that every relevant face is "close" to at least one critical face. This is the first of two properties that play a major role in our analysis, exhibiting the connection between relevant and critical faces.

Lemma 5.6. *Let* G *be a disk graph with a realization* $\mathcal{D}$ *in general position of ply* p*, and let* $M \subseteq V(G)$ *such that for all* $v \in M$*,* $N_G(v) \setminus M$ *is an independent set. Then, for every* $F \in \mathsf{relevant}(\mathcal{D}, M)$*,* $\mathsf{distance}_{\mathcal{D}}(F, \mathsf{critical}(\mathcal{D}, M)) \in \mathcal{O}(p)$*.*

Proof. Let $F \in \mathsf{relevant}(\mathcal{D}, M)$. Consider the following process:

1. Initialize $F^\star = F$.

2. As long as $F^\star$ has a boundary edge b that is either (i) concave and belongs to a (unique) disk in $\mathsf{disk}_{\mathcal{D}}(M)$ or (ii) convex and belongs to a (unique) disk in $\mathcal{D} \setminus \mathsf{disk}_{\mathcal{D}}(M)$, update $F^\star$ as follows:

   (a) Let $\widehat{F}$ be the other face of $\mathcal{D}$ that has b as a boundary edge.

   (b) Update $F^\star$ to be $\widehat{F}$.

Let $F_1, F_2, F_3, \ldots$ be the sequence of faces encountered in the above process. So, $F_1 = F$. For every i, let $\mathcal{D}_i$ be the set of disks in $\mathsf{disk}_{\mathcal{D}}(M)$ that contain F_i. Due to Observation 5.4, and by the definition of the process itself, $\mathcal{D}_1 \subseteq \mathcal{D}_2 \subseteq \mathcal{D}_3 \subseteq \ldots$, and for every $i \geq 2$ such that F_i was chosen according to condition (i) in Step 2, $\mathcal{D}_{i-1}$ is a strict subset of $\mathcal{D}_i$. Because the ply of $\mathcal{D}$ is p, this means that the total number of times in the process where condition (i) in Step 2 was true is at most p. Now, note that because F_1 is relevant, $\mathcal{D}_1 \neq \emptyset$. So, by Lemma 5.5, there cannot exist two consecutive times in the process where condition (ii) in Step 2 was true. We use $F^\star$ to denote the face attained at the end of the process. Overall, it means that the number of faces in the sequence $F_1, F_2, F_3, \ldots$ is $\mathcal{O}(p)$, and hence $\mathsf{distance}(F, F^\star) \in \mathcal{O}(p)$. Notice that, by the conditions of Step 2, $F^\star \in \mathsf{critical}(\mathcal{D}, M)$. Thus, we derive that $\mathsf{distance}(F, \mathsf{critical}(\mathcal{D}, M)) \in \mathcal{O}(p)$. □

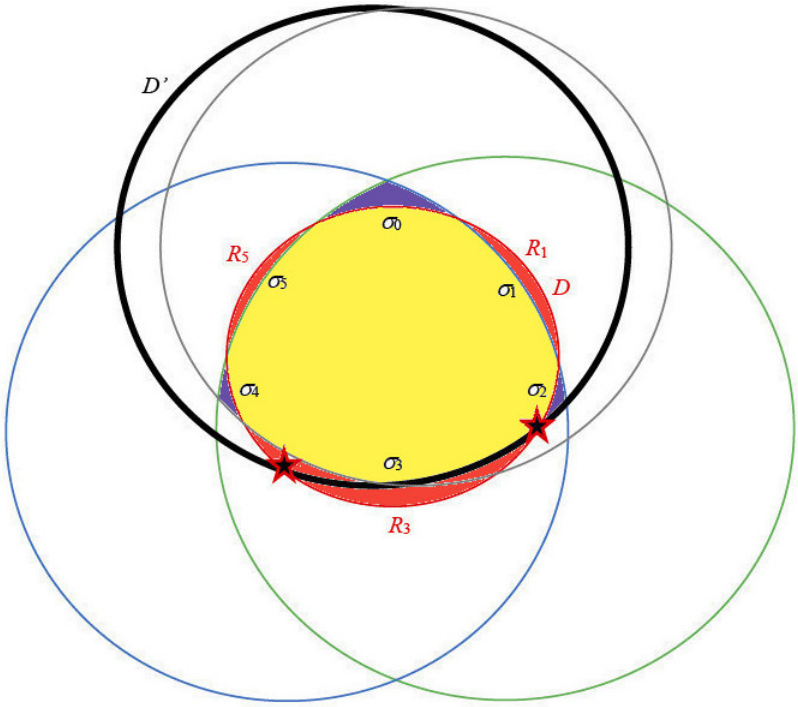

Figure 4: Illustration for the proof of Lemma 5.7. The disk D is (circled) in red. The intersection I is the union of the yellow and purple regions. The intersection $I \cap D$ is the yellow region. The regions R_1, R_3, R_5 are the red regions (note that R_3 is the union of three faces). The disk D' is (circled) in thick black. The intersection points of the boundaries of D and D' are marked by stars.

For the proof of the second property, we will need the following observation and lemma.

Lemma 5.7. *Let $\mathcal{D}$ be a collection of disks. Let D be another disk. Let $I = \bigcap \mathcal{D}$ be convex, and let $U = \bigcup \mathcal{D}$. Suppose that the removal of $D \cap I$ from I splits I into two (or more) disjoint regions. Then, $D \subseteq U$.*

Proof. First, observe that because both I and D are convex, $D \cap I$ is a single (i.e., connected) region (see Fig. 4). Second, observe that the boundary of $D \cap I$ can be uniquely (up to cyclic shifts) broken into segments $\sigma_0, \sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{r-1}, \sigma_r$ (see Fig. 4) such that:

- For every $i \in \{0, 1, \ldots, r\}$, σ_i and $\sigma_{(i+1) \mod r}$ intersect exactly at one of their endpoints. There are no other intersections between the segments.
- $r \geq 3$ is odd (hence, the number of segments is even at is at least 4) .
- For every even $i \in \{0, 1, \ldots, r\}$, σ_i belongs to the boundary of D, and it may be a single point.
- For every odd $i \in \{0, 1, \ldots, r\}$, σ_i belongs to the boundary of I, it must not be a single point, and every internal point of σ_i does not belong to the boundary of D.

For every odd $i \in \{0, 1, \ldots, r\}$, let R_i denote the region outside I formed by σ_i and part of the boundary of D (see Fig. 4). Notice that $D \subseteq I \cup (\bigcup_i R_i)$. So, to complete the proof, it suffices to prove that each R_i is contained in U. We will only show this for R_1, since the other cases are symmetric.

Towards this, let $D' \in \mathcal{D}$ be a disk whose boundary is a non-trivial part (that is, not a single point) of σ_3 and which contains an endpoint of σ_3 (see Fig. 4). (Possibly, the boundary of D' can contain σ_3.) Then, the boundary of D' intersects the boundary of D in R_3 twice (see Fig. 4). But then, because

[4]If F has a boundary edge that belongs to the boundary of a disk whose vertex is in M as well as to the boundary of a disk whose vertex is not in M, then F cannot be critical.

[5]Uniqueness follows from the general position supposition.

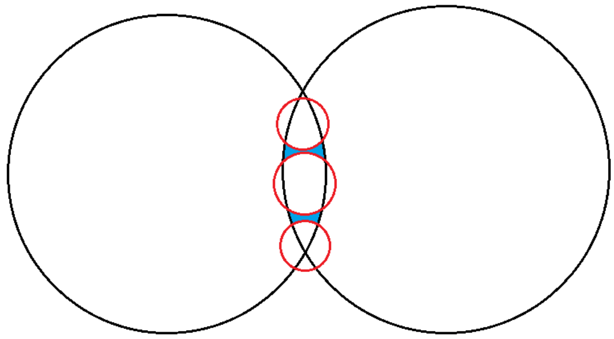

Figure 5: The intersection of the two black disks is split into two regions. In particular, if F is one of the blue regions, D_1 and D_2 are the black disks, and D_1' and D_2' are the two red disks such that parts of their boundaries are edges of the boundary of F, then it does not hold that $\bigcap_{i=1}^{2} D_i \subseteq F \cup (\bigcup_{i=1}^{2} D_i')$.

two (non-identical) circles can intersect at more twice, there are no other intersection points between the boundaries of D' and D. However, this implies that $R_1 \subseteq D'$, and hence $R_1 \subseteq U$. $\square$

Lemma 5.8. *Let G be a disk graph with a realization $\mathcal{D}$ of ply p, and let $M \subseteq V(G)$ such that for all $v \in M$, $N_G(v) \setminus M$ is an independent set, and no disk $\mathsf{disk}_{\mathcal{D}}(V(G) \setminus M)$ is contained in the union of disks in $\mathsf{disk}_{\mathcal{D}}(M)$. Let $F \in \mathsf{critical}(\mathcal{D}, M)$. Let $D_1, \ldots, D_\ell$ be the disks in $\mathsf{disk}_{\mathcal{D}}(M)$ that correspond to the boundary edges of F, and let $D_1', \ldots, D_t'$ be the disks in $\mathcal{D} \setminus \mathsf{disk}_{\mathcal{D}}(M)$ that correspond to the boundary edges of F. Let W be the (unique) face of $\mathsf{disk}_{\mathcal{D}}(M)$ that contains F. Then,*

$$F \subseteq W \subseteq \bigcap_{i=1}^{\ell} D_i \subseteq F \cup (\bigcup_{i=1}^{t} D_i').$$

Moreover, $D_1', \ldots, D_t'$ are pairwise disjoint, and, furthermore, have empty intersection within W with all other disks from $\mathcal{D} \setminus \mathsf{disk}_{\mathcal{D}}(M)$.

Proof. It is immediate that $F \subseteq W \subseteq \bigcap_{i=1}^{\ell} D_i$. Observe that there exists a disk in $\mathsf{disk}_{\mathcal{D}}(M)$ that intersects all of the disks $D_1', \ldots, D_t'$ (to see this, consider a disk in $\mathsf{disk}_{\mathcal{D}}(M)$ that contains F—its existence follows because F is critical and hence, in particular, relevant). So, because for all $v \in M$, $N_G(v) \setminus M$ is an independent set, it follows that the disks $D_1', \ldots, D_t'$ are pairwise disjoint and, furthermore, have empty intersection within W with all other disks from $\mathcal{D} \setminus \mathsf{disk}_{\mathcal{D}}(M)$. Due to Lemma 5.7, and because no disk in $\mathsf{disk}_{\mathcal{D}}(V(G) \setminus M)$ is contained in the union of disks in $\mathsf{disk}_{\mathcal{D}}(M)$, the removal of none of the disks $D_1', \ldots, D_t'$ can split $\bigcap_{i=1}^{\ell} D_i$ into two disjoint regions. Because $D_1', \ldots, D_t'$ are pairwise disjoint, we further get that $(\bigcap_{i=1}^{\ell} D_i) \setminus (\bigcup_{i=1}^{t} D_i')$ is a *single* connected region. So, in particular, this region must be F. That is, $\bigcap_{i=1}^{\ell} D_i = F \cup \Big((\bigcup_{i=1}^{t} D_i') \cap (\bigcap_{i=1}^{\ell} D_i)\Big)$, and hence the lemma follows. $\square$

We remark that Fig 5 shows that without the supposition that no disk $\mathsf{disk}_{\mathcal{D}}(V(G) \setminus M)$ is contained in the union of disks in $\mathsf{disk}_{\mathcal{D}}(M)$, the statement of Lemma 5.8 would have been false.

We are now ready to prove the second property, which we will use in order to derive an upper bound on $|\mathsf{critical}(\mathcal{D}, M)|$.

Lemma 5.9. *Let G be a disk graph with a realization $\mathcal{D}$ of ply p, and let $M \subseteq V(G)$ such that for all $v \in M$, $N_G(v) \setminus M$ is an independent set, and no disk $\mathsf{disk}_{\mathcal{D}}(V(G) \setminus M)$ is contained in the union of disks in $\mathsf{disk}_{\mathcal{D}}(M)$. Then, each face of $\mathsf{disk}_{\mathcal{D}}(M)$ contains at most one face from $\mathsf{critical}(\mathcal{D}, M)$.*

Proof. Targeting a contradiction, suppose that there exist two distinct faces $F, \widehat{F} \in \mathsf{critical}(\mathcal{D}, M)$ that are contained in the same face W of $\mathsf{disk}_{\mathcal{D}}(M)$. Let $D_1', \ldots, D_t'$ be the the disks in $\mathcal{D} \setminus \mathsf{disk}_{\mathcal{D}}(M)$ that correspond to the boundary edges of F, which, by Lemma 5.8, are pairwise disjoint, and, furthermore, have empty intersection within W with all other disks from $\mathcal{D} \setminus \mathsf{disk}_{\mathcal{D}}(M)$. By Lemma 5.8, $W \subseteq F \cup (\bigcup_{i=1}^{t} D'_i)$. As $\widehat{F} \subseteq W$ and $\widehat{F} \cap F = \emptyset$, it follows that $\widehat{F} \subseteq \bigcup_{i=1}^{t} D'_i$. Since the disks $D'_1, \ldots, D'_t$ are pairwise disjoint, there exists $i \in \{1, 2, \ldots, t\}$ such that $\widehat{F} \subseteq D'_i \cap W$. Observe that $D'_i \cap W$ is a face in $\mathcal{D}$: it cannot contain a boundary edge of another disk from $\mathcal{D} \setminus \mathsf{disk}_{\mathcal{D}}(M)$ (because, as argued earlier, D'_i intersects no other disk in $\mathcal{D} \setminus \mathsf{disk}_{\mathcal{D}}(M)$), and it cannot contain strictly inside a boundary edge of another disk from $\mathsf{disk}_{\mathcal{D}}(M)$ because W is a face of $\mathsf{disk}_{\mathcal{D}}(M)$. So, $\widehat{F} = D'_i \cap W$. But then, since D'_i does not contain W (since part of its boundary is concave with respect to $F \subseteq W$), one of the boundary edges of $\widehat{F}$ is convex and belongs to D'_i. This is a contradiction to the supposition that $\widehat{F}$ is critical. ◻

The following is a corollary of Lemma 5.9.

Corollary 5.10. *Let G be a disk graph with a realization $\mathcal{D}$ of ply p, and let $M \subseteq V(G)$ such that for all $v \in M$, $N_G(v) \setminus M$ is an independent set, and no disk $\mathsf{disk}_{\mathcal{D}}(V(G) \setminus M)$ is contained in the union of disks in $\mathsf{disk}_{\mathcal{D}}(M)$. Then, $|\mathsf{critical}(\mathcal{D}, M)|$ is bounded from above by the number of faces of $\mathsf{disk}_{\mathcal{D}}(M)$.*

In turn, due to Proposition 3.11, we further have the following consequence of Corollary 5.10, which is the anticipated upper bound on $|\mathsf{critical}(\mathcal{D}, M)|$.

Corollary 5.11. *Let G be a disk graph with a realization $\mathcal{D}$ of ply p, and let $M \subseteq V(G)$ such that for all $v \in M$, $N_G(v) \setminus M$ is an independent set, and no disk $\mathsf{disk}_{\mathcal{D}}(V(G) \setminus M)$ is contained in the union of disks in $\mathsf{disk}_{\mathcal{D}}(M)$. Then, $|\mathsf{critical}(\mathcal{D}, M)| \in \mathcal{O}(p \cdot |M|)$.*

With the following lemma, we will be able to substitute the condition posed on $\mathcal{D}$ in Lemma 5.11 (that no disk $\mathsf{disk}_{\mathcal{D}}(V(G) \setminus M)$ is contained in the union of disks in $\mathsf{disk}_{\mathcal{D}}(M)$) with a "realization-free" condition posed directly on G and M. Note that this is also the condition stated in Theorem 2.

Lemma 5.12. *Let G be a disk graph with a realization $\mathcal{D}$ of ply p, and let $M \subseteq V(G)$ such that for all $v \in M$, $N_G(v) \setminus M$ is an independent set, and there does not exist a vertex in $V(G) \setminus M$ whose neighborhood is contained in M. Then, no disk in $\mathsf{disk}_{\mathcal{D}}(V(G) \setminus M)$ is contained in the union of disks in $\mathsf{disk}_{\mathcal{D}}(M)$.*

Proof. Targeting a contradiction, suppose that there exists a disk $D \in \mathsf{disk}_{\mathcal{D}}(V(G) \setminus M)$ that is contained in the union of disks in $\mathsf{disk}_{\mathcal{D}}(M)$. Because there does not exist a vertex in $V(G) \setminus M$ whose neighborhood is contained in M, the disk D must intersect another disk $D' \in \mathsf{disk}_{\mathcal{D}}(V(G) \setminus M)$. However, because D is contained in the union of disks in $\mathsf{disk}_{\mathcal{D}}(M)$, so does the intersection of D and D'. In particular, this means that there exists a disk in $\mathsf{disk}_{\mathcal{D}}(M)$ with non-empty intersection with both D and D'. However, this is a contradiction to the supposition that for all $v \in M$, $N_G(v) \setminus M$ is an independent set. ◻

So, we combine Corollary 5.11 and Lemma 5.12 to attain the following.

Lemma 5.13. *Let G be a disk graph with a realization $\mathcal{D}$ of ply p, and let $M \subseteq V(G)$ such that for all $v \in M$, $N_G(v) \setminus M$ is an independent set, and there does not exist a vertex in $V(G) \setminus M$ whose neighborhood is contained in M. Then, $|\mathsf{critical}(\mathcal{D}, M)| \in \mathcal{O}(p \cdot |M|)$.*

Proof. By Lemma 5.12, no disk in $\mathsf{disk}_{\mathcal{D}}(V(G) \setminus M)$ is contained in the union of disks in $\mathsf{disk}_{\mathcal{D}}(M)$. Hence, we can use Corollary 5.11, and derive that $|\mathsf{critical}(\mathcal{D}, M)| \in \mathcal{O}(p \cdot |M|)$. ◻

For the proof of Theorem 2 rather than only its corollary (Corollary 1.1), we will need to consider the the arrangement graphs of G/Q and $(G - M)/Q$ (rather than of G and $G - M$). Thus, we now turn to focus on Q. Here, we first need to clarify which realization is used with respect to $(G - M)/Q$.

Definition 5.14. *Let G be a disk graph with a realization $\mathcal{D}$, and let $Q \subseteq V(G)$. The $(\mathcal{D}, Q)$-*contraction*, denoted by $\mathcal{D}/Q$, is the set of objects associated with vertices of G/Q as follows: with every vertex $v \in V(G) \setminus Q$, associate $\mathsf{disk}_{\mathcal{D}}(v)$, and with every vertex $v \in V(G/Q) \setminus V(G)$, which corresponds to the contraction of some connected set C of $G[Q]$, associate $\bigcup_{u \in V(C)} \mathsf{disk}_{\mathcal{D}}(u)$.*

Now, we verify that Definition 5.14 indeed corresponds to a realization.

Observation 5.15. *Let G be a disk graph with a realization $\mathcal{D}$ of ply p, and let $Q \subseteq V(G)$. Then, $\mathcal{D}/Q$ is a realization of G/Q of ply at most p.*

Proof. First, we note that the ply of $\mathcal{D}/Q$ is clearly not larger than that of $\mathcal{D}$. We focus on the claim that $\mathcal{D}/Q$ is a realization of G/Q. Towards this, let $v \in V(G/Q)$. Let $\mathcal{O}$ be the set of objects in $\mathcal{D}/Q$ intersected by $\mathsf{obj}_{\mathcal{D}/Q}(v)$. Now, notice that for every object $O \in \mathcal{O}$, there exist a disk $D \in \mathcal{D}$ that is part of[6] $\mathsf{obj}_{\mathcal{D}/Q}(v)$ and a disk $D' \in \mathcal{D}$ that is part of O that have non-empty intersection. In turn, by Definition of 5.14 and the operation of contraction in graphs, this means that two objects in $\mathcal{D}/Q$ have non-empty intersection if and only if they are associated with adjacent vertices u, v in G/Q. $\square$

We proceed to show that the fact that we need to deal with G/Q (and $(G-M)/Q$) rather than G (and $(G-M)/Q$) is inconsequential to our proof strategy that relies on the analysis of relevant faces as these faces remain unchanged, as stated below.

Lemma 5.16. *Let G be a disk graph with a realization $\mathcal{D}$, and let $M \subseteq V(G)$ such that for all $v \in M$, $N_G(v) \setminus M$ is an independent set. Let $Q \subseteq V(G) \setminus M$. Then, $\mathsf{relevant}(\mathcal{D}, M)$ is a subset of the set of faces of $\mathcal{D}/Q$.*

Proof. Let $F \in \mathsf{relevant}(\mathcal{D}, M)$. To show that F is a face of $\mathcal{D}/Q$ (rather than just contained in a face of $\mathcal{D}/Q$), we need to argue that all of the boundary edges of F exist in $\mathcal{D}/Q$. To this end, consider a boundary b of F. If b is not common to two (adjacent) disks in $\mathsf{disk}_{\mathcal{D}}(Q)$, then, by Definition 5.14, it exists in $\mathcal{D}/Q$. Otherwise, there exist two disks $D, D' \in \mathsf{disk}_{\mathcal{D}}(Q)$ such that b is a boundary edge of both. We will show that this case cannot occur. Towards this, observe that since F is a relevant face, there exists a disk in $\mathsf{disk}_{\mathcal{D}}(M)$ that contains F and therefore intersects both D and D'. However, as D and D' share a boundary edge, they also intersect each other. Because $Q \cap M = \emptyset$, this is a contradiction to the supposition that for all $v \in M$, $N_G(v) \setminus M$ is an independent set. $\square$

As a corollary of Observation 5.2 and Lemma 5.16, we obtain the following.

Corollary 5.17. *Let G be a disk graph with a realization $\mathcal{D}$, and let $M \subseteq V(G)$ such that for all $v \in M$, $N_G(v) \setminus M$ is an independent set, and there does not exist a vertex in $V(G) \setminus M$ whose neighborhood is contained in M. Let $Q \subseteq V(G) \setminus M$. Let A and B be the arrangement graphs of $\mathcal{D}/Q$ and $(\mathcal{D} \setminus \mathsf{disk}_{\mathcal{D}}(M))/Q$. Let U be the set of vertices in A that represent the faces in $\mathsf{relevant}(\mathcal{D}, M)$. Then, $A - U$ and B are isomorphic.*

We proceed with a sequence of statements whose purpose is to eventually assert (in Lemma 5.22) that the deletion of "few" vertices, as well as vertices "close" to them, from a planar graph, cannot significantly decrease its treewidth. Towards that, we start with the following.

Definition 5.18. *Let H be a $3t \times 3t$ grid with a corresponding vertex set $V(H) = \{v_{i,j} : i, j \in \{1, 2, \dots, 3t\}\}$. Then, the H-*interior *is the grid $H[\{v_{i,j} : i, j \in \{t{+}1, \dots, 2t\}\}]$. The H-*complementation *is the graph obtained from H by adding the two following sets of edges: $\{\{v_{i,j}, v_{x,y}\}\} : i, j, x, y \in \{1, 3t\}\}$ and $\{\{v_{i,j}, v_{x,y}\} : |i - x| = 1, |j - y| = 1|\}$.*[7]

Observation 5.19. *Let G be a planar graph that has a $3t \times 3t$ grid $\widehat{H}$ as a minor with a minor model φ. Let H be the $\widehat{H}$-interior. Let $x, y \in V(H)$, $u \in \varphi(x)$ and $v \in \varphi(y)$. Then, $\dfrac{\mathsf{distance}_H(x, y)}{2} \leq \mathsf{distance}_G(u, v)$.*

Proof. Let H' be the $\widehat{H}$-completion. Consider any path P between u and v in G. From the definitions of planarity and a minor model, it follows that there exists a path P' between x and y in H' such that for every $z \in V(P')$, $\varphi(z) \cap V(P) \neq \emptyset$. Thus, $\mathsf{distance}_{H'}(x, y) \leq \mathsf{distance}_G(u, v)$. Because x, y belong to $V(H)$ (rather than just to $V(\widehat{H})$, $\frac{\mathsf{distance}_H(x,y)}{2} \leq \mathsf{distance}_{H'}(x, y)$. Thus, the observation follows. $\square$

[6] That is, either $\mathsf{obj}_{\mathcal{D}/Q}(v)$ is simply D, or $\mathsf{obj}_{\mathcal{D}/Q}(v)$ was obtained by taking a union over a set of disks that contains D.

[7] Informally, we make every two vertices on the "boundary" of H adjacent, as well as add two edges inside each of its internal faces (whose boundaries each corresponds to a cycle on four vertices); see Fig. 6.

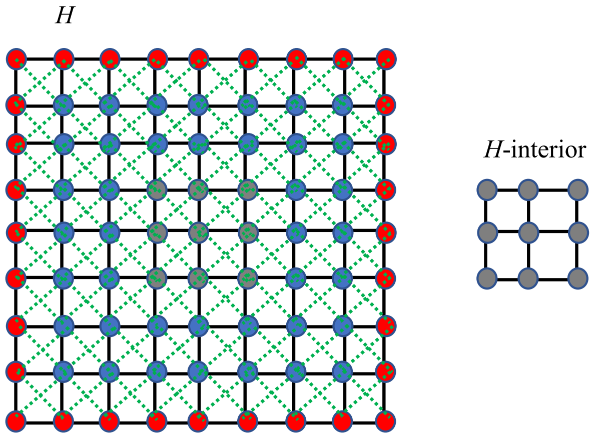

Figure 6: A $3t \times t$ grid H. The H-interior is the subgraph of H induced by the set of grey vertices (also drawn on the right). The H-completion of H is obtained from H by adding all the green dashed edges as well as turning the subgraph induced by the set of red vertices into a clique.

Based on Observation 5.19, we prove the following.

Lemma 5.20. *Let G be a planar graph that has a $3t \times 3t$ grid $\widehat{H}$ as a minor with a minor model φ. Let H be the $\widehat{H}$-interior. Let $D \subseteq D' \subseteq V(G)$ such that for every $v \in D'$, $\mathsf{distance}_G(v, D) \leq \delta$ for some $\delta \in \mathbb{N}$. Then, there exist at most $\mathcal{O}(\delta^2 \cdot |D|)$ vertices $x \in V(H)$ such that $\varphi(x) \cap D' \neq \emptyset$.*

Proof. Let $U = \{x \in V(H) : \varphi(x) \cap D \neq \emptyset\}$. So, $|U| \leq |D|$. Let $U' = \{x \in V(H) : \varphi(x) \cap D' \neq \emptyset\}$. Additionally, let $U^\star = \{x \in V(H) : \mathsf{distance}_H(x, U) \leq 2\delta\}$. By the supposition of the lemma, for every $v \in D'$, $\mathsf{distance}_G(v, D) \leq \delta$. Hence, by Observation 5.19, for every $v \in D' \cap (\bigcup_{x \in V(H)} \varphi(x))$, the vertex $x \in V(H)$ such that $v \in \varphi(x)$ must belong to $U^\star$. Thus, $U' \subseteq U^\star$. Now, notice that $|U^\star| \leq \cdot (2\delta)^2 \cdot |U|$. Hence, $|U'| \leq (2\delta)^2 \cdot |U| \leq (2\delta)^2 \cdot |D| \in \mathcal{O}(\delta^2 \cdot |D|)$. □

We proceed to prove that the removal of "few" vertices from a grid cannot hit all "large" grids that it contains as minors.

Observation 5.21. *Let H be a $t \times t$ grid. Let $X \subseteq V(H)$. Then, $H - X$ contains a $t' \times t'$ grid as a minor where $t' \in \Omega(\min\{\frac{t^2}{|X|}, t\})$.*

Proof. Let $s = \max\{10|X|, 2t\}$. Let $r = \lfloor \frac{t^2}{s} \rfloor$. Observe that H contains the following $\ell = (\lfloor \frac{t}{r} \rfloor)^2$ pairwise vertex-disjoint $r \times r$ subgraphs that are grids: $H_{i,j} = H[\{((i-1) \cdot \ell + x, (j-1) \cdot \ell + y) : x, y \in \{1, 2, \ldots, r\}\}]$ for every $i, j \in \{1, 2, \ldots, \ell\}$. Note that $\ell \geq (\lfloor \frac{s}{t} \rfloor)^2 > (\frac{s}{t} - 1)^2 \geq (\frac{s}{2t})^2$ (where the last two inequalities follow since $s > 2t$). Let $t' = r - \lceil \frac{t^2}{2s} \rceil \in \Omega(\min\{\frac{t^2}{|X|}, t\})$. Targeting a contradiction, suppose $H - X$ does not contain a $t' \times t'$ grid as a minor. Then, from each $H_{i,j}$, X must contain at least $\lceil \frac{t^2}{2s} \rceil \geq \frac{t^2}{2s}$ vertices. Indeed, otherwise there will be at least t' "columns" as well as t' "rows" in $H_{i,j}$ with no vertex from X, which will result in a $t' \times t'$ grid as a minor already in $H_{i,j} - (X \cap V(H_{i,j}))$, which is a subgraph of $H - X$. So, from all aforementioned ℓ grids, X must altogether contain at least $\ell \cdot \frac{t^2}{2s} > (\frac{s}{2t})^2 \cdot \frac{t^2}{2s} = \frac{s}{8} > |X|$ vertices, which is a contradiction. □

We use Lemma 5.20 and Observation 5.21 to prove the following.

Lemma 5.22. *Let G be a planar graph of treewidth w. Let $D \subseteq D' \subseteq V(G)$ such that for every $v \in D'$, $\mathsf{distance}(v, D) \leq \delta$ for some $\delta \in \mathbb{N}$. Then, the treewidth of $G - D'$ is $\Omega(\min\{\frac{w^2}{\delta^2 \cdot |D|}, w\})$.*

Proof. By Proposition 3.7, G contains a $3t \times 3t$ grid $\widehat{H}$ as a minor where $t = \lfloor (\lceil \frac{w}{5} \rceil)/3 \rfloor$. Let H be the $\widehat{H}$-interior, which is a $t \times t$ grid. By Lemma 5.20, there exist at most $\mathcal{O}(\delta^2 \cdot |D|)$ vertices $x \in V(H)$

such that $\varphi(x) \cap D' \neq \emptyset$. Denote this set of vertices by X. By Observation 5.21, $H - X$ contains a $t' \times t'$ grid as a minor where $t' \in \Omega(\min\{\frac{t^2}{|X|}, t\}) \subseteq \Omega(\min\{\frac{w^2}{\delta^2 \cdot |D|}, t\})$. However, this implies that $G - D'$ contains a $t' \times t'$ grid as a minor. By Propositions 3.4 and 3.5, this implies that the treewidth of $G - D'$ is $\Omega(\min\{\frac{w^2}{\delta^2 \cdot |D|}, w\})$. $\square$

Finally, having Lemmata 5.6 and 5.13, Observation 5.15, and Lemmata 5.16 and 5.22, we are ready the prove Theorem 1.

Theorem 2. *Let G be a disk graph that has some realization of ply p, and let $M \subseteq V(G), Q \subseteq V(G) \setminus M$ be such that for all $v \in M$, $N_G(v) \setminus M$ is an independent set, there does not exist a vertex in $V(G) \setminus M$ whose neighborhood is contained in M, and $(G - M)/Q$ has treewidth w. Then, the treewidth of G/Q is $\mathcal{O}(\max\{\sqrt{|M| \cdot w} \cdot p^{2.5}, w \cdot p\})$.*

Proof. Let $\mathcal{D}$ denote the realization of G of ply p. Let A denote the arrangement graph of $\mathcal{D}/Q$, and let B denote the arrangement graph of $(\mathcal{D} \setminus \mathsf{disk}_{\mathcal{D}}(M))/Q$. Let U_{relevant} and U_{critical} denote the sets of vertices in A that represent the faces in $\mathsf{relevant}(\mathcal{D}, M)$ and in $\mathsf{critical}(\mathcal{D}, M)$, respectively; the notations U_{relevant} and U_{critical} are well defined due to Lemma 5.16. Let w_A denote the treewidth of A. By Observation 5.15, $\mathcal{D}/Q$ is a realization of G/Q of ply at most p (and, hence, $(\mathcal{D} \setminus \mathsf{disk}_{\mathcal{D}}(M))/Q$ is a realization of $(G - M)/Q$ of ply at most p). Thus, by Proposition 3.12, the treewidth of G/Q is $\mathcal{O}(w_A \cdot p)$. So, to prove the theorem, it suffices to prove that $w_A \in \mathcal{O}(\max\{\sqrt{|M| \cdot w} \cdot p^{1.5}, w\})$. Note that by Lemma 5.12 and Corollary 5.17, $A - U_{\mathsf{relevant}}$ and B are isomorphic, and by Proposition 3.12, the treewidth of $(G - M)/Q$ is upper bounded by $\mathcal{O}$ of the treewidth of B. Hence, w is upper bounded by $\mathcal{O}$ of the treewidth of $A - U_{\mathsf{relevant}}$. In what follows, we slightly abuse notation and let w denote the treewidth of $A - U_{\mathsf{relevant}}$.

By Lemma 5.6, for every $F \in \mathsf{relevant}(\mathcal{D}, M)$, $\mathsf{distance}_{\mathcal{D}}(F, \mathsf{critical}(\mathcal{D}, M)) \leq \delta$ for some $\delta \in \mathcal{O}(p)$, and by Lemma 5.13, $|\mathsf{critical}(\mathcal{D}, M)| \in \mathcal{O}(p \cdot |M|)$. Also, recall that all the faces in $\mathsf{relevant}(\mathcal{D}, M)$ appear (unchanged) in $\mathcal{D}/Q$ (by Lemma 5.16). So, $|U_{\mathsf{critical}}| \in \mathcal{O}(p \cdot |M|)$, and for every $v \in U_{\mathsf{relevant}}$, $\mathsf{distance}_A(v, U_{\mathsf{critical}}) \in \mathcal{O}(p)$. Thus, by Lemma 5.22 (and since arrangement graphs are planar),

$$w \in \Omega(\min\{\frac{w_A^2}{\delta^2 \cdot |U_{\mathsf{critical}}|}, w_A\}) \subseteq \Omega(\min\{\frac{w_A^2}{p^3 \cdot |M|}, w_A\}).$$

So, we know that at least one of the following holds.

- $w_A \in \mathcal{O}(w)$ (if $w_A \geq \frac{w_A^2}{p^3 \cdot |M|}$).
- $\frac{w_A^2}{p^3 \cdot |M|} \in \mathcal{O}(w)$ (if $w_A \leq \frac{w_A^2}{p^3 \cdot |M|}$), which means that $w_A \in \mathcal{O}(\sqrt{|M| \cdot w} \cdot p^{1.5})$.

Thus, $w_A \in \mathcal{O}(\max\{w, \sqrt{|M| \cdot w} \cdot p^{1.5}\})$. As argued earlier, this completes the proof. $\square$

6 Applications

In this section, we prove Theorem 3 (in Sections 6.2 and 6.3) and Theorem 4 (in Section 6.4). Towards this, we first present in Section 6.1 lemmata that will be used by all our applications. In particular, this refers to the two reductions given by Lemmata 6.6 and 6.13, used to meet conditions that will allow us to make use of Theorems 1 and 2. Note that none of the algorithms in this section requires to be given a realization for a given disk graph.

6.1 Reduction to "Enriched" Versions

The purpose of the first reduction will be to reduce the ply of any realization of a given disk graph G. Towards its presentation, we will need the following.

Proposition 6.1 ([8]). *There exists a polynomial-time algorithm that, given a disk graph G, returns a clique C in G whose size (in terms of number of vertices) is at least half the size of a maximum-sized clique in G.*[8]

We start with the main lemma using which we will reduce the maximum clique number (and hence also the ply of any realization) of a given disk graph.

Lemma 6.2. *There exists a $2^{\mathcal{O}(\frac{k}{p}\cdot\log p)}\cdot n^{\mathcal{O}(1)}$-time algorithm that, given a disk graph G and $p,k\in\mathbb{N}_0$ with $p\geq 6$, returns a collection $\mathcal{Y}\subseteq\{(D,U,\mathcal{K}) : D,U\subseteq V(G), D\cap U=\emptyset, \mathcal{K}$ is a partition of $D\}$ of size $2^{\mathcal{O}(\frac{k}{p}\cdot\log p)}$ such that:*

1. *For every $(D,U,\mathcal{K})\in\mathcal{Y}$, $G-D$ does not have a clique of size larger than p.*
2. *For every $S\subseteq V(G)$ of size at most k such that $G-S$ is triangle-free, there exists a unique $(D,U,\mathcal{K})\in\mathcal{Y}$ such that $D\subseteq S$ and $S\cap U=\emptyset$.*

Proof. We first present the algorithm. Let (G,X,p,k) denote the input. This is a recursive algorithm, which takes as input the graph G, the input integer p, $k'\in\mathbb{N}_0$ with $k'\leq k$, $=D',U'\subseteq V(G)$ such that $D'\cap U'=\emptyset$ and a partition $\mathcal{K}'$ of D'. In the first call, $k'=k$ and $D'=U'=\mathcal{K}'=\emptyset$.

1. Call the algorithm in Proposition 6.1 on $G-D'$. Let C be its output clique.
2. If $|V(C)|>\frac{p}{2}$, then:
   (a) Initialize $\mathcal{Y}'=\emptyset$.
   (b) For $X\subseteq V(C)\setminus U'$ such that $|X|\geq|V(C)|-2$ and $k'-|X|\geq 0$:
      i. Let $\widehat{\mathcal{Y}}$ be the output to the recursive call with $G,p,k'\leftarrow k'-|X|, D'\leftarrow D'\cup X, U'\leftarrow U'\cup(V(C)\setminus X)$ and $\mathcal{K}'\leftarrow\mathcal{K}'\cup\{X\}$.
      ii. Update $\mathcal{Y}'\leftarrow\mathcal{Y}'\cup\widehat{\mathcal{Y}}$.
   (c) Return $\mathcal{Y}'$.
3. Else: Return $\mathcal{Y}'=\{(D',U',\mathcal{K}')\}$.

We now consider the correctness of the algorithm. The satisfaction of the first property in the statement of the lemma immediately follows from the condition in Step 2 and the correctness of the algorithm in Proposition 6.1. The satisfaction of the second property follows from the following claim.

Claim 6.3. *Consider a call to the algorithm with G,p,k',D' and U'. For every $S\subseteq V(G)\setminus(D'\cup U')$ of size at most k' such that $G-(S\cup D')$ is triangle-free, there exists a unique $(D,U,\mathcal{K})\in\mathcal{Y}'$ such that $D\subseteq S\cup D'$ and $S\cap U=\emptyset$.*

The claim easily follows from induction on the number of recursive calls, and from the observation that any set that intersects all triangles in a graph, must contain all vertices except for at most two from each clique in the graph.

We now consider the time complexity of the algorithm. Observe that in each call, for some $x>p/2\geq 3$, the algorithm makes at most $1+x+\binom{x}{2}\leq x^2$ recursive calls, each with the parameter k' decreased by at least $x-2$. Observe that recursive calls are never made with a negative parameter. Let us denote the time complexity of a call to the algorithm with parameter k' by $T(k')$. For $k'\leq\frac{p}{2}$, $T(k')\in n^{\mathcal{O}(1)}$. For $k'>\frac{p}{2}$,

$$T(k')\in\max_{x>\frac{p}{2}\text{ s.t. }k'-(x-2)\geq 0} x^2\cdot T(k'-(x-2))+n^{\mathcal{O}(1)}.$$

[8]IBy [8], the problem of finding a maximum clique in a disk graph admits an EPTAS (Efficient Polynomial-Time Approximation Scheme), but the result as stated is sufficient for us. In fact, any constant-factor (and even worse) approximation algorithm is sufficient for us.

So, for $k' > \frac{p}{2}$,

$$T(k') \in (\frac{p}{2})^2 \cdot T(k' - \frac{p}{2} + 2) + n^{\mathcal{O}(1)}.$$

This resolves to $T(k) \in \left((\frac{p}{2})^2\right)^{\frac{2k}{p}} \cdot n^{\mathcal{O}(1)} \subseteq 2^{\mathcal{O}(\frac{k}{p}\cdot\log p)} \cdot n^{\mathcal{O}(1)}$.

So, the time complexity of the algorithm is as stated in the lemma. The size of the output collection $\mathcal{Y}$ is bounded using the same recursive formula as the one above used to bound the time complexity, with the exception that the term $n^{\mathcal{O}(1)}$ is dropped. Thus, as in the analysis above, it is bounded by $2^{\mathcal{O}(\frac{k}{p}\cdot\log p)}$. □

Due to Proposition 3.9, we have the following corollary of Lemma 6.2.

Corollary 6.4. *There exists a $2^{\mathcal{O}(\frac{k}{p}\cdot\log p)}\cdot n^{\mathcal{O}(1)}$-time algorithm that, given a disk graph G and $p, k \in \mathbb{N}_0$ with $p \geq 6$, returns a collection $\mathcal{Y} \subseteq \{(D, U, \mathcal{K}) : D, U \subseteq V(G), D \cap U = \emptyset, \mathcal{K}$ is a partition of $D\}$ of size $2^{\mathcal{O}(\frac{k}{p}\cdot\log p)}$ such that:*

1. *For every $(D, U, \mathcal{K}) \in \mathcal{Y}$, every realization of $G - D$ has ply at most p.*
2. *For every $S \subseteq V(G)$ of size at most k such that $G - S$ is triangle-free, there exists a unique $(D, U, \mathcal{K}) \in \mathcal{Y}$ such that $D \subseteq S$ and $S \cap U = \emptyset$.*

We now turn to present the precise definition of the intermediate problem to which we reduce.

Definition 6.5. *Let $\mathcal{F}$ be a graph class. Let Π be* #Weighted $\mathcal{F}$-Vertex Deletion *on disk graphs. Then,* Ply-Enriched Π *(on disk graphs) is defined as follows. The input of* Ply-Enriched Π *includes a disk graph G, $\mathfrak{w} : V(G) \to \mathbb{N}_0$, $p, k \in \mathbb{N}_0$ with $6 \leq p \leq k$ and $U \subseteq V(G)$ such any realization of G has ply at most p. The objective is to count the number of solutions $S \subseteq V(G)$ for Π of minimum weight among those such that $U \cap S = \emptyset$, and return this weight.*

The unweighted counting case, the weighted decision case, and the unweighted decision case are defined similarly, except that: in both the (weighed and unweighted) decision versions, it can be assumed that $U = \emptyset$; in the weighted decision version, we are given also $W \in \mathbb{N}_0$; in both the (counting and decision) unweighted versions, we are not given $\mathfrak{w}$.

We are now ready to present the first reduction.

Lemma 6.6. *Let $\mathcal{F}$ be a triangle-free graph class. Let Π be* #Weighted (Connected) $\mathcal{F}$-Vertex Deletion *on disk graphs or special case of it.*[9] *Suppose that there exists a T-time algorithm for* Ply-Enriched Π. *Then, there exists a $T \cdot 2^{\mathcal{O}(\frac{k}{p}\cdot\log p)} \cdot n^{\mathcal{O}(1)}$-time algorithm for Π.*

Proof. We will only consider #Weighted $\mathcal{F}$-Vertex Deletion; the proofs for the special cases is similar, and, in fact, simpler for the decision special cases (in particular, the "uniqueness" in the second property of Corollary 6.4 is not required for them).

We first present the algorithm, called $\mathcal{B}$. For this purpose, we let $\mathcal{A}$ denote the algorithm in the supposition of the lemma. Let $(G, \mathfrak{w}, k)$ denote the input (where $\mathfrak{w}$ is the vertex-weight function).

1. Call the algorithm in Corollary 6.4 on (G, p, k) to obtain a collection $\mathcal{Y}$.
2. Initialize $r = 0, W = \infty$.
3. For every $(D, U) \in \mathcal{Y}$ such that $k \geq |D|$:
   (a) Call $\mathcal{A}$ on $(G - D, \mathfrak{w}, U, p, k - |D|)$, and let $(r^\star, W^\star)$ denote its output.
   (b) Update $W^\star \leftarrow W^\star + \sum_{v\in D} \mathfrak{w}(v)$.
   (c) If $W = W^\star$: Update $r \leftarrow r + r^\star$.
   (d) Else if $W^\star < W$: Let $r = r^\star$ and $W = W^\star$.

[9]That is, the unweighted counting case, the weighted decision case, and the unweighted decision case.

4. Return (r, W).

First, from the first property in Corollary 6.4, it directly follows that all calls to $\mathcal{A}$ are done with a valid input. From the time complexity bounds of the algorithm in Corollary 6.4 and algorithm $\mathcal{A}$, it directly follows that $\mathcal{B}$ runs in time $T \cdot 2^{\mathcal{O}(\frac{k}{p} \cdot \log p)} \cdot n^{\mathcal{O}(1)}$.

Next, we consider the correctness of $\mathcal{B}$. For this purpose, consider a solution S for Π (which is, in particular, of size at most k and such that $G - S$ is triangle-free) of minimum weight. By the second property in Corollary 6.4, there exists a unique $(D, U, \mathcal{K}) \in \mathcal{Y}$ such that $D \subseteq S$ and $S \cap U = \emptyset$. So, S will be counted exactly once—in the call to $\mathcal{A}$ done in the iteration corresponding to this $(D, U, \mathcal{K})$. This completes the proof. $\square$

For the second reduction, we will need the following proposition.

Proposition 6.7 ([19]). *There exists a polynomial-time algorithm that, given a graph G, compute a maximum matching of G.*

The purpose of the second reduction will be to ensure, for some vertex subset M, that the neighborhood of each vertex in M outside M is an independent set. The following lemma will lie at the heart of this reduction.

Lemma 6.8. *There exists a $2^{\mathcal{O}(\frac{k}{\lambda} \cdot \log k)} \cdot n^{\mathcal{O}(1)}$-time algorithm that, given a graph G, $\lambda \in \mathbb{N}, k \in \mathbb{N}_0$, and a set $X \subseteq G$ of size at most k such that $G - X$ is triangle-free, returns a collection $\mathcal{Z} \subseteq \{(D, U, Z) : D, U \subseteq X, D \cap U = \emptyset, Z \subseteq V(G) \setminus X\}$ of size $2^{\mathcal{O}(\frac{k}{\lambda} \cdot \log k)}$ such that:*

1. *For every $(D, U, Z) \in \mathcal{Z}$, $|Z| \leq 2k$ and for every $v \in X \setminus D$, $G[N_G(v) \setminus (X \cup Z)]$ has maximum matching of size at most λ.*
2. *For every $S \subseteq V(G)$ of size at most k such that $G - S$ is triangle-free, there exists a unique $(D, U, Z) \in \mathcal{Z}$ such that $D \subseteq S$ and $S \cap U = \emptyset$.*

Proof. We first present the algorithm. Let (G, X, λ, k) denote the input. This is a recursive algorithm, which takes as input the graph G, the input integer $\lambda, k' \in \mathbb{N}_0$ with $k' \leq k$, $D', U' \subseteq X$ such that $D' \cap U' = \emptyset$, and $Z' \subseteq V(G) \setminus X$. In the first call, $k' = k$ and $D' = U' = Z' = \emptyset$.

1. For every vertex $v \in X \setminus D'$:
   (a) Call the algorithm in Proposition 6.7 on $G[N_G(v) \setminus (X \cup Z)]$ to obtain a maximum matching[10] M_v in this graph.
   (b) If $|M_v| > \lambda$, then:
      i. If $k' \geq 1$, then: let $\mathcal{Z}_1$ be the output of the recursive call with $G, \lambda, k' \leftarrow k' - 1, D' \leftarrow D' \cup \{v\}, U', Z'$. Otherwise: $\mathcal{Z}_1 = \emptyset$.
      ii. If $k' \geq |M_v|$, then: let $\mathcal{Z}_2$ be the output of the recursive call with $G, \lambda, k' \leftarrow k' - |M_v|, D', U' \leftarrow U' \cup \{v\}, Z' \leftarrow Z' \cup V(M_v)$. Otherwise: $\mathcal{Z}_2 = \emptyset$.
      iii. Return $\mathcal{Z}_1 \cup \mathcal{Z}_2$.
2. Return $\mathcal{Z}' = \{(D', U', Z')\}$.

We now consider the correctness of the algorithm. First, note that the algorithm only reaches Step 2 if $|M_v| \leq \lambda$ for all $v \in X \setminus D'$, and hence for every $(D, U, Z) \in \mathcal{Z}$ and for every $v \in X \setminus D$, $G[N_G(v) \setminus (X \cup Z)]$ has maximum matching of size at most λ. Moreover, whenever the algorithm updates Z', the number of elements inserted is at most twice as large as the number by which k' is decreased. Hence, it follows that for every $(D, U, Z) \in \mathcal{Z}$, $|Z| \leq 2k$. The satisfaction of the second property follows from the third item of the following claim.

[10]If fact, the computation of a maximal matching suffices, but we choose to compute a maximum matching to simplify the presentation (this allows us to later derive Lemma 6.10 more immediately).

Claim 6.9. *Consider a call to the algorithm with G, λ, k', D', U' and Z'. Then,*

1. *$k' = k - |D'| - |Z'|/2$.*
2. *For every $S \subseteq V(G) \setminus U'$ such that $G - S$ is triangle-free, S must contain at least half of the vertices in Z'.*
3. *For every $S \subseteq V(G) \setminus U'$ of size at most k that contains D' such that $G - S$ is triangle-free, there exists a unique $(D, U, Z) \in \mathcal{Z}'$ such that $D \subseteq S \cup D'$, $S \cap U = \emptyset$.*

The first item of the claim directly follows from the way in which the algorithm updates k', D' and Z' (in Steps 1(b)i and 1(b)ii). For the second item of the claim, notice that the algorithm updates Z' only in Step 1(b)ii: in this step, a new vertex v is inserted into U', and the set of new vertices inserted into Z' correspond to a maximum matching in the neighborhood of v (in $G - (X \cup Z')$ with respect to the former set Z'). Now, notice that every $S \subseteq V(G)$ that excludes v and such that $G - S$ is triangle-free, must include at least one vertex from each edge from the aforementioned matching. In turn, this implies the correctness of the second item. For the third item, first note that the first two items imply that if k' drops below 0, then there does not exist $S \subseteq V(G) \setminus U'$ of size at most k that contains D' such that $G - S$ is triangle-free. This justifies the assignments of $\emptyset$ done in Steps 1(b)i and 1(b)ii. With this at hand, the correctness of the third item directly follows from induction on the number of calls; indeed, Steps 1(b)i and 1(b)ii correspond to an "exhaustive branching" with v being inserted into D' in one branch, and v being inserted into U' in the second branch.

We now consider the time complexity of the algorithm. Towards this, observe that recursive calls are never made with a negative parameter k'. Let us denote the time complexity of a call to the algorithm with parameter k' by $T(k')$. For $k' = 0$, $T(k') \in n^{\mathcal{O}(1)}$. For $1 \leq k' < \lambda$,

$$T(k') \in T(k' - 1) + n^{\mathcal{O}(1)}.$$

For $k' \geq \lambda$,

$$T(k') \in T(k' - 1) + T(k' - \lambda) + n^{\mathcal{O}(1)}.$$

This resolves to $T(k) \in 2^{\mathcal{O}(\frac{k}{\lambda} \log k)} \cdot n^{\mathcal{O}(1)}$. This can be easily seen by bounding the number of leaves in the recursion tree corresponding to $T(k)$—observe that the number of nodes on a root to leaf path is at most k, and among them, at most k/λ can correspond to a decrease of the parameter by λ; hence, the number of leaves is bounded from above by $\binom{k}{(k/\lambda)} \in 2^{\mathcal{O}(\frac{k}{\lambda} \log k)}$. The size of the output collection $\mathcal{Z}$ is bounded using the same recursive formula as the one above used to bound the time complexity, with the exception that the term $n^{\mathcal{O}(1)}$ is dropped. Thus, as in the analysis above, it is bounded by $2^{\mathcal{O}(\frac{k}{\lambda} \cdot \log k)}$. ◻

We use Lemma 6.8 in order to prove the following.

Lemma 6.10. *There exists a $2^{\mathcal{O}(\frac{k}{\lambda} \cdot \log k)} \cdot n^{\mathcal{O}(1)}$-time algorithm that, given a graph G, $\lambda, k \in \mathbb{N}_0$, and a set $X \subseteq G$ of size at most k such that $G - X$ is triangle-free, returns a collection $\mathcal{Z} \subseteq \{(D, U, Z) : D, U \subseteq X, D \cap U = \emptyset, Z \subseteq V(G) \setminus X\}$ of size $2^{\mathcal{O}(\frac{k}{\lambda} \cdot \log k)}$ such that:*

1. *For every $(D, U, Z) \in \mathcal{Z}$, $|Z| \leq 2(k + \lambda|X \setminus D|)$ and for every $v \in V(G) \setminus D$, $N_G(v) \setminus (X \cup Z)$ is an independent set.*
2. *For every $S \subseteq V(G)$ of size at most k such that $G - S$ is triangle-free, there exists a unique $(D, U, Z) \in \mathcal{Z}$ such that $D \subseteq S$ and $S \cap U = \emptyset$.*

Proof. We first present the algorithm. Let (G, X, λ, k) denote the input.

1. Call the algorithm in Lemma 6.8 to obtain a collection $\mathcal{Z}'$.
2. For every $t' = (D', U', Z') \in \mathcal{Z}'$, let $\alpha(t') = (D', U', Z)$ where Z is computed as follows:

(a) For every vertex $v \in X \setminus D'$: Call the algorithm in Proposition 6.7 on $G[N_G(v) \setminus (X \cup Z')]$ to obtain a maximum matching M_v in this graph.

(b) Let $Z = Z' \cup (\bigcup_{v \in X \setminus D'} V(M_v))$.

3. Return $\mathcal{Z} = \{\alpha(t') : t' \in \mathcal{Z}'\}$.

We now consider the correctness of the algorithm. Because we do not change the first two arguments of any triple in $\mathcal{Z}'$, the satisfaction (by $\mathcal{Z}$) of the second property follows from the satisfaction of this property by $\mathcal{Z}'$. For the first property, consider some $(D', U', Z') \in \mathcal{Z}'$. Note that $|Z'| \leq 2k$ and for every $v \in X \setminus D'$, $|M_v| \leq \lambda$. So, for the corresponding Z, we have $|Z| \leq 2(k + \lambda|X \setminus D'|)$. Moreover, for every $v \in X \setminus D'$, $N_G(v) \setminus (X \cup Z)$ is an independent set because $N_G(v) \setminus (X \cup Z' \cup M_v)$ is an independent set (because M_v is a maximum matching in $G[N_G(v) \setminus (X \cup Z')]$). Additionally, for every vertex $v \in V(G) \setminus X$, $N_G(v) \setminus X$ is an independent set since $G - X$ is triangle-free, and thus also $N_G(v) \setminus (X \cup Z)$ is an independent set.

The stated running time bound and the bound on $|\mathcal{Z}|$ directly follow from the corresponding bounds in Lemma 6.8. □

We now turn to present the precise definition of the problem we will reduce to in order to resolve TRIANGLE HITTING, FEEDBACK VERTEX SET and ODD CYCLE TRANSVERSAL.

Definition 6.11. *Let $\mathcal{F}$ be a graph class. Let Π be #WEIGHTED (CONNECTED) $\mathcal{F}$-VERTEX DELETION. Then, ENRICHED Π (on disk graphs) is defined as follows. The input of ENRICHED Π includes a disk graph G, $\mathfrak{w} : V(G) \to \mathbb{N}$, $p, k \in \mathbb{N}_0$ with $6 \leq p \leq k$, and $U, M \subseteq V(G)$ such that $G - M$ is triangle-free, any realization of G has ply at most p, $|M| = \mathcal{O}(p \cdot k)$ and for every $v \in V(G)$, $N_G(v) \setminus M$ is an independent set. The objective is to count the number of solutions $S \subseteq V(G)$ for Π of minimum weight among those such that $U \cap S = \emptyset$, and return this weight.*

The unweighted counting case, the weighted decision case, and the unweighted decision case are defined similarly, except that: in both the (weighed and unweighted) decision versions, it can be assumed that $U = \emptyset$; in the weighted decision version, we are given also $W \in \mathbb{N}_0$; in both the (counting and decision) unweighted versions, we are not given $\mathfrak{w}$.

It is folklore that TRIANGLE HITTING (being a special case of 3-HITTING SET) admits a 3-approximation algorithm: as long as the graph has a triangle, delete all of its vertices.

Proposition 6.12 ([41]). TRIANGLE HITTING *admits a polynomial-time 3-approximation algorithm.*

We are now ready to present the second reduction.

Lemma 6.13. *Let $\mathcal{F}$ be a triangle-free graph class. Let Π be #WEIGHTED $\mathcal{F}$-VERTEX DELETION on disk graphs or special case of it. Suppose that there exists a T-time algorithm for ENRICHED Π. Then, there exists a $T \cdot 2^{\mathcal{O}(\frac{k}{p} \cdot \log k)} \cdot n^{\mathcal{O}(1)}$-time algorithm for Π.*

Proof. We will only consider #WEIGHTED $\mathcal{F}$-VERTEX DELETION; the proofs for the special cases is similar, and, in fact, simpler for the decision special cases (in particular, the "uniqueness" in the second properties of Corollary 6.4 and Lemma 6.10 is not required for them). By Lemma 6.6, it suffices to show that PLY-ENRICHED Π can be solved in time $T \cdot 2^{\mathcal{O}(\frac{k}{p} \cdot \log k)} \cdot n^{\mathcal{O}(1)}$.

We first present the algorithm (for PLY-ENRICHED Π), called $\mathcal{B}$. For this purpose, we let $\mathcal{A}$ denote the algorithm in the supposition of the lemma. Let $(G, \mathfrak{w}, U, \mathcal{K}, k)$ denote the input.

1. Call the algorithm in Proposition 6.12 on G to obtain a 3-approximate solution X.

2. If $|X| > 3k$: Return $(0, \infty)$.

3. Call the algorithm in Lemma 6.10 on (G, X, p, k) (i.e., $\lambda = p$) to obtain a collection $\mathcal{Z}$.

4. Initialize $r = 0, W = \infty$.

5. For every $(D', U', Z') \in \mathcal{Z}$ such that $D' \cap U = \emptyset$ and $k \geq |D'|$:
   (a) Call $\mathcal{A}$ on $(G - D', \mathfrak{w}, U^\star = U \cup U', M^\star = (X \cup Z') \setminus D', p, k - |D'|)$, and let $(r^\star, W^\star)$ denote its output.
   (b) Update $W^\star \leftarrow W^\star + \sum_{v \in D'} \mathfrak{w}(v)$.
   (c) If $W = W^\star$: Update $r \leftarrow r + r^\star$.
   (d) Else if $W^\star < W$: Let $r = r^\star$ and $W = W^\star$.
6. Return (r, W).

First, from Definition 6.5 and first property in Lemma 6.10, because X is a solution to Triangle Hitting and due to the test in Step 5, it directly follows that all calls to $\mathcal{A}$ are done with a valid input. From the time complexity bounds of all the algorithms that algorithm $\mathcal{B}$ calls (stated in the lemma supposition, Lemma 6.10 and Proposition 6.12), it directly follows that $\mathcal{B}$ runs in time $T \cdot 2^{\mathcal{O}(\frac{k}{p} \cdot \log k)} \cdot n^{\mathcal{O}(1)}$.

Next, we consider the correctness of $\mathcal{B}$. For this purpose, consider a solution S for Ply-Enriched Π (which is, in particular, of size at most k and such that $G - S$ is triangle-free) of minimum weight. By the second property in Lemma 6.10, there exists a unique $(D', U', Z') \in \mathcal{Z}$ such that $D' \subseteq S$ and $S \cap U' = \emptyset$. So, S will be counted exactly once—in the call to $\mathcal{A}$ done in the iteration corresponding to this (D', U', Z'). This completes the proof. ☐

6.2 Triangle Hitting

We start with a simple observation.

Observation 6.14. *Let (G, k) be an instance of* Triangle Hitting. *Let $u, v \in V(G)$ be false twins. Then, any minimal triangle hitting set S of G either contains both u and v or does not contain any of them. In particular, if $|S \cap \{u, v\}| = 1$, then $S \setminus \{u, v\}$ is also a triangle hitting set.*

Proof. Let S be a solution such that $|S \cap \{u, v\}| = 1$. Without loss of generality, suppose that $u \in S$. Targeting a contradiction, suppose that $S \setminus \{u\}$ is not a solution. This means that G has a triangle C that contains u. Let x, y be the two neighbors of u in C. By the definition of false twins, x, y are both neighbors of v. However, this means that $x - v - y - x$ is a cycle in G that does not contain any vertex from S, which contradicts the supposition that S is a solution. ☐

In addition, the following can be proved using standard dynamic programming over nice tree decompositions (see [10]). The only difference compared to the known algorithm is that the treewidth of $\mathsf{FTR}(G)$, rather than the treewidth of G, is bounded. However, this is essentially inconsequential, since by Observation 6.14, for each class of false twins, every minimal solution either contains the entire class, or none of the vertices in that class.

Proposition 6.15 (Similar to [10])**.** *There exists a $2^{\mathcal{O}(w \log w)} n^{\mathcal{O}(1)}$-time algorithm that, given an instance $(G, \mathfrak{w}, k)$ of* #Weighted Triangle Hitting *such that $\mathsf{FTR}(G)$ has treewidth at most w, returns the number of minimum-weight solutions S of $(G, \mathfrak{w}, k)$, and the corresponding weight.*

Thus, we can already present the main component of the algorithm in this subsection.

Lemma 6.16. Enriched #Weighted Triangle Hitting *on disk graphs is solvable in time $2^{\mathcal{O}(\ell \log \ell)} \cdot n^{\mathcal{O}(1)}$ where $\ell = \sqrt{k} \cdot p^4$.*

Proof. We first present the algorithm, denoted by $\mathcal{A}$. Let $(G, \mathfrak{w}, U, M, p, k)$ denote the input.

1. Let $G' = G - E(G - M)$. Define $\mathfrak{w}'$ to be $\mathfrak{w}|_{V(G')}$ with the exception that every vertex in U is assigned ∞.[11]

[11] Practically, ∞ refers to a large enough number, e.g., $\sum_{v \in V(G)} \mathfrak{w}(v) + 1$. The return of a pair whose second argument is ∞ is considered to mean "no-instance" (there exists no solution).

2. Call the algorithm in Proposition 6.15 on $(G', \mathfrak{w}', k)$, and return its output (r', W').

We first consider the correctness of the algorithm. Towards that, we have the following claim.

Claim 6.17. *There does not exist a triangle in G that contains an edge from $E(G - M)$.*

Towards the proof of this claim, suppose that there exists a triangle T in G that contains some edge $\{x, y\} \in E(G - M)$. Denote $V(T) = \{x, y, z\}$. In particular, we have that $z \in V(G)$, and that $\{u, v\} \subseteq N_G(z) \setminus M$ and $\{u, v\} \in E(G)$. However, by Definition 6.11, for every $v \in V(G)$, $N_G(v) \setminus M$ is an independent set. We have thus reached a contradiction. This completes the proof of the claim.

Having Claim 6.17 at hand, we proceed with the proof of correctness. On the one hand, consider a solution S counted by $(G, \mathfrak{w}, U, M, p, k)$. Then, S is a minimum-weight triangle hitting set of G among those of size at most k such that $S \cap U = \emptyset$. Let W be its weight (by $\mathfrak{w}$). So, S is a triangle hitting set of G' (because it is a subgraph of G) of size at most k such that $S \cap U = \emptyset$, hence its weight (by $\mathfrak{w}'$) is W. On the other hand, let S be a solution of $(G', \mathfrak{w}', k)$ of minimum weight, and let W' be its weight (by $\mathfrak{w}'$). Then, $S \cap U = \emptyset$ (else $W' = \infty$, but then S is not considered to be a solution). By Claim 6.17, we derive that S is a solution of $(G, \mathfrak{w}, k)$ of minimum weight (being W') among those such that $S \cap U = \emptyset$. Thus, the correctness of algorithm $\mathcal{A}$ follows from the correctness of the algorithm in Proposition 6.15.

We now consider the running time of algorithm $\mathcal{A}$. Let w denote the treewidth of $\mathsf{FTR}(G')$. Then, from the pseudocode and the time complexity of the algorithm in Proposition 6.15, we derive that algorithm $\mathcal{A}$ runs in time $2^{\mathcal{O}(w \log w)} n^{\mathcal{O}(1)}$. So, to complete the proof, it remains to show that $w \in \mathcal{O}(\sqrt{k} \cdot p^4)$. To this end, by Proposition 3.10, it suffices to show that $|V(\mathsf{FTR}(G'))| \in \mathcal{O}(k \cdot p^7)$. By Definition 6.5, any realization of G has ply at most p. Moreover, by Definition 6.11, $|M| \in \mathcal{O}(p \cdot k)$, and for every $v \in V(G)$, $N_G(v) \setminus M$ is an independent set. So, by Theorem 1, for any $X \subseteq V(G) \setminus M$ such that for all distinct $u, v \in X$, $N_{G'}(u) = N_G(u) \cap M \neq N_G(v) \cap M = N_{G'}(v)$, it follows that $|X| \in \mathcal{O}(|M| \cdot p^6) \subseteq \mathcal{O}(k \cdot p^7)$. Therefore, we have $|V(\mathsf{FTR}(G'))| \in \mathcal{O}(|M| + k \cdot p^7) \subseteq \mathcal{O}(k \cdot p^7)$, which completes the proof. □

We conclude with our main result for #Weighted Triangle Hitting (and hence also for its special cases Triangle Hitting, Weighted Triangle Hitting and #Connected Triangle Hitting).

Lemma 6.18. #Weighted Triangle Hitting *on disk graphs is solvable in time* $2^{\mathcal{O}(k^{\frac{9}{10}} \log k)} \cdot n^{\mathcal{O}(1)}$.

Proof. The correctness follows from Lemmata 6.13 and 6.16 with $p = k^{\frac{1}{10}}$. □

6.3 Feedback Vertex Set

For our result for Feedback Vertex Set and its generalizations, we will use an approximation algorithm for Feedback Vertex Set:

Proposition 6.19 ([7]). Feedback Vertex Set *admits a 2-approximation polynomial-time algorithm.*

We start with a simple observation.

Observation 6.20. *Let (G, k) be an instance of* Feedback Vertex Set. *Let $u, v, q \in V(G)$ be false twins. Then, any minimal triangle hitting set S of G either contains at least two among u, v and q or does not contain any of them. In particular, if $|S \cap \{u, v, q\}| = 1$, then $S \setminus \{u, v, q\}$ is also a feedback vertex set.*

Proof. Let S be a solution such that $|S \cap \{u, v, q\}| = 1$. Without loss of generality, suppose that $u \in S$. Targeting a contradiction, suppose that $S \setminus \{u\}$ is not a solution. This means that G has a cycle C that contains u. Let x, y be the two neighbors of u in C. By the definition of false twins, x, y are both neighbors of v and q. However, this means that $x - v - y - q - x$ is a cycle in G that does not contain any vertex from S, which contradicts the supposition that S is a solution. □

In addition, the following can be proved using standard dynamic programming over nice tree decompositions (see [10]). The only difference compared to the known algorithm is that the treewidth of $\mathsf{FTR}(G)$, rather than the treewidth of G, is bounded. However, this is essentially inconsequential, since by Observation 6.20, for each class of false twins, every minimal solution either contains the entire class except for at most one vertex, or none of the vertices in that class.

Proposition 6.21 (Similar to [10])**.** *There exists a $2^{\mathcal{O}(w \log w)} n^{\mathcal{O}(1)}$-time algorithm that, given an instance $(G, \mathfrak{w}, k)$ of #Weighted Feedback Vertex Set such that $\mathsf{FTR}(G)$ has treewidth at most w, returns the number of minimum-weight solutions S of $(G, \mathfrak{w}, k)$, and the corresponding weight.*

Next, we present the main component of the algorithm in this subsection.

Lemma 6.22. Enriched #Weighted Feedback Vertex Set *on disk graphs is solvable in time $2^{\mathcal{O}(\ell \log \ell)} \cdot n^{\mathcal{O}(1)}$ where $\ell = \sqrt{k} \cdot p^6$.*

Proof. We first present the algorithm, denoted by $\mathcal{A}$. Let $(G, \mathfrak{w}, U, M, p, k)$ denote the input.

1. Call the algorithm in Proposition 6.19. Let C be its output feedback vertex set.
2. If $|C| > 2k$: Return $(0, \infty)$.
3. Define $\mathfrak{w}'$ to be $\mathfrak{w}$ with the exception that every vertex in U is assigned ∞.
4. Call the algorithm in Proposition 6.21 on $(G, \mathfrak{w}', k)$, and return its output (r', W').

The correctness of algorithm $\mathcal{A}$ directly follows from the correctness of the algorithms in Propositions 6.19 and 6.21 (as well as the definition of $\mathfrak{w}'$).

We now consider the running time of algorithm $\mathcal{A}$. Let w denote the treewidth of $\mathsf{FTR}(G)$. Then, from the pseudocode and the time complexity of the algorithm in Proposition 6.21, we derive that algorithm $\mathcal{A}$ runs in time $2^{\mathcal{O}(w \log w)} n^{\mathcal{O}(1)}$. So, to complete the proof, it remains to show that $w \in \mathcal{O}(\sqrt{k} \cdot p^6)$. For this purpose, observe that by Definition 6.11, any realization of G has ply at most p. Moreover, by Definition 6.11, $|M| \in \mathcal{O}(p \cdot k)$, and for every $v \in V(G)$, $N_G(v) \setminus M$ is an independent set. Thus, by Theorem 1, for any $X \subseteq V(G) \setminus M$ such that for all distinct $u, v \in X$, $N_G(u) \cap M \neq N_G(v) \cap M$, it follows that $|X| \in \mathcal{O}(|M| \cdot p^6) \subseteq \mathcal{O}(k \cdot p^7)$. Let $I = \{v \in V(G) \setminus M : N_G(v) \subseteq M\}$. Then, $|I \cap V(\mathsf{FTR}(G))| \in \mathcal{O}(k \cdot p^7)$. Let $M^\star = M \cup I \cup C$, and note that $|C| \leq 2k$. Then, $|M^\star \cap V(\mathsf{FTR}(G))| \leq |M| + |I \cap V(\mathsf{FTR}(G))| + |C| \in \mathcal{O}(k \cdot p^7)$. Notice that, since $M \subseteq M^\star$, for every $v \in V(G)$, $N_G(v) \setminus M^\star$ is an independent set. Moreover, since $I \subseteq M^\star$, there does not exist a vertex in $V(G) \setminus M^\star$ whose neighborhood is contained in $M^\star$. Additionally, since $C \subseteq M^\star$, the treewidth of $G - M^\star$ is 1. Thus, from Corollary 1.1, it follows that the treewidth of $\mathsf{FTR}(G)$ is $\mathcal{O}(\sqrt{|M^\star \cap V(\mathsf{FTR}(G))|} \cdot p^{\frac{5}{2}}) \subseteq \mathcal{O}(\sqrt{k \cdot p^7} \cdot p^{\frac{5}{2}}) \subseteq \mathcal{O}(\sqrt{k} \cdot p^6)$, which completes the proof. ☐

We conclude with our main result for #Weighted Feedback Vertex Set (and hence also for its special cases Feedback Vertex Set, Weighted Feedback Vertex Set and #Connected Feedback Vertex Set).

Lemma 6.23. #Weighted Feedback Vertex Set *on disk graphs is solvable in time $2^{\mathcal{O}(k^{\frac{13}{14}} \log k)} \cdot n^{\mathcal{O}(1)}$.*

Proof. The correctness follows from Lemmata 6.13 and 6.22 with $p = k^{\frac{1}{14}}$. ☐

6.4 Odd Cycle Transversal

The application of Theorems 1 and 4 to Odd Cycle Transversal is more involved than the previous applications. First, we will need the following algorithm, which yields a "small" set guaranteed to contain a solution (if one exists). We remark that due to our use of this lemma, we only resolve the unweighted decision of the problem, and our algorithm is randomized.

Proposition 6.24 ([33])**.** *There exists a randomized*[12] *polynomial-time algorithm that, given an instance (G,k) of* Odd Cycle Traversal*, outputs a subset $S^\star \subseteq V(G)$ such that $|S^\star| \in k^{\mathcal{O}(1)}$, and (with high probability) (G,k) is a yes-instance if and only if G has an odd cycle transversal of size at most k contained in $S^\star$.*

We proceed with several other results required for our application and/or the proof of its correctness. For this purpose, let $\mathcal{D}$ be some realization of a disk graph G. Now, consider the following drawing of G: each vertex is drawn on the centre of the disk in $\mathcal{D}$ that represents it, and each edge is drawn as a straight line between its two vertices. It is known that if the drawing of some two distinct edges intersect not at their endpoints, then G contains a triangle (consisting of three of the four endpoints of the aforementioned edges), which in particular yields the following proposition.

Proposition 6.25 (Folklore, see, e.g., [29])**.** *Let G be a disk graph that has no triangle. Then, G is a planar graph.*

Now, we present the following observation concerning false twins, which we will later use to eliminate some of them (unlike the previous problems, here we deal only with the decision problem, hence such elimination is possible).

Observation 6.26. *Let (G,k) be an instance of* Odd Cycle Transversal*. Let $u,v \in V(G)$ be false twins. Then, any minimal odd cycle transversal S of G either contains both u and v or does not contain any of them. In particular, if $|S \cap \{u,v\}| = 1$, then $S \setminus \{u,v\}$ is also an odd cycle transversal.*

Proof. Here, we implicitly use Proposition 3.2. Consider an odd cycle transversal S of G such that $|S \cap \{u,v\}| = 1$. So, $G - S$ is a bipartite graph with bipartition (A,B). Without loss of generality, suppose that $v \in S$ and $u \in A$. Then, because u and v are false twins, $G-(S\setminus\{v\})$ is also a bipartite graph, having $(A\cup\{v\},B)$ as a bipartition. Hence, $S\setminus\{v\}$ is also an odd cycle transversal of G. ☐

Additionally, we will use the following in order to bound the treewidth of part of the input graph after the contraction of some edges.

Proposition 6.27 ([4])**.** *Let G be a planar graph. For any integer p, one can compute in polynomial time disjoint sets $Z_1,\dots,Z_p \subseteq V(G)$ such that for any $i \in \{1,2,\dots,t\}$ and $Z' \subseteq Z_i$, the treewidth of $G/(Z_i \setminus Z')$ is at most $\mathcal{O}(p+|Z'|)$.*

Although we use p for ply, we stated the proposition above with the letter p denoting the given integer to be consistent with [4]. When we use it, we will consider a different letter (being t).

Since we can only bound the treewidth of part of the input graph after the contraction of some edges, we will make use of two additional definitions and one lemma in order to deal with the issues that arise from the performed contraction operations. These are presented below.

Definition 6.28. *Let G be a graph with an edge-labeling $\mathfrak{lab} : E(G) \to \{\{1\},\{2\},\{1,2\}\}$. Then, the* $\mathfrak{lab}$-subdivision *of G is the graph H obtained from G as follows: every edge labeled $\{2\}$ is subdivided once, and every edge labeled $\{1,2\}$ is duplicated and then one of its copies is subdivided once.*

Definition 6.29. *The* Weighted Edge-Labeled Odd Cycle Transversal *problem is defined as follows. The input consists of a graph G that has a vertex-weight function $\mathfrak{w} : V(G) \to \mathbb{N}_0$ and an edge-labeling $\mathfrak{lab} : E(G) \to \{\{1\},\{2\},\{1,2\}\}$, and $k, W \in \mathbb{N}_0$. The objective is to determine whether there exists a subset $S \subseteq V(G)$ of size at most k and weight at most W that is an odd cycle transversal for the $\mathfrak{lab}$-subdivision of G.*

Definition 6.30. *Let $I = (G,\mathfrak{w},W,k)$ be an instance of* Weighted Odd Cycle Transversal*. Let $<$ be an ordering of $V(G)$. Let $U \subseteq V(G)$. Given a vertex $v \in V(G/U)$, let $\mathsf{origin}_{G,<}(U,v)$ denote the smallest (according to $<$) vertex in $\mathsf{uncontract}_G(U,v)$. Then, the* $(<,U)$-edge-labeled instance corresponding to I *is $(G',\mathfrak{w}',\mathfrak{lab},W,k)$ where:*

[12]This randomized algorithm has success probability at least $(1-\frac{1}{2^n})$ (which can, in fact, be made much smaller), hence it should be clear that the success probably of our algorithm can also be made arbitrarily close to 1. For this reason, we do not discuss this issue further.

- $G' = G/U$.
- $\mathfrak{w}' : V(G') \to \mathbb{N}_0$ *is defined as follows: For every* $v \in V(G')$, $\mathfrak{w}'(v) = \mathfrak{w}(\mathsf{origin}_{G,<}(U,v))$.
- $\mathfrak{lab} : E(G/U) \to \{\{1\},\{2\},\{1,2\}\}$ *is defined as follows. Consider an edge* $e = \{u,v\} \in E(G)$, *and let* $u' = \mathsf{origin}_{G,<}(U,u)$ *and* $v' = \mathsf{origin}_{G,<}(U,v)$. *Then,*
  - *If between* u' *and* v' *there exist both an even path and an odd path:* $\mathfrak{lab}(e) = \{1,2\}$.
  - *If between* u' *and* v' *there exist an even path but not an odd path:* $\mathfrak{lab}(e) = \{2\}$.
  - *If between* u' *and* v' *there exist an odd path but not an even path:* $\mathfrak{lab}(e) = \{1\}$.

Lemma 6.31. *Let* $I = (G, \mathfrak{w}, W, k)$ *be an instance of* Weighted Odd Cycle Transversal. *Let* $U \subseteq V(G)$ *such that* $G[U]$ *has no odd cycle. Let* $S \subseteq V(G) \cup V(G')$. *Then,* $S \subseteq V(G)$ *and is a solution for* $(G, \mathfrak{w}, W, k)$ *such that* $S \cap U = \emptyset$ *if and only if* $S \subseteq V(G')$ *and is a solution for the* $(<,U)$*-edge-labeled instance corresponding to* I, $(G', \mathfrak{w}', \mathfrak{lab}, W, k)$, *such that* $S \cap U' = \emptyset$ *where* $U' = V(G') \setminus V(G)$.

Proof. In this proof, we implicitly rely on Proposition 3.1. Let $\mathcal{C}$ denote the connected components of $G[U]$. By Definitions 6.28, 6.29 and 6.30, to prove the lemma, it suffices to prove the two following claims:

1. For every odd cycle O in G (that is not contained in $G[U]$), there exists a closed walk O' in G' such that there exists a choice for a label for each one of its edges (where the label is chosen from the subset assigned to the edge by $\mathfrak{lab}$) such that the sum of labels is odd, and $V(O) \setminus U = V(O') \setminus U'$.
2. For every cycle O' in G' such that there exists a choice for a label for each one of its edges (where the label is chosen from the subset assigned to the edge by $\mathfrak{lab}$) such that the sum of labels is odd, there exists an odd closed walk O in G such that $V(O) \setminus U = V(O') \setminus U'$.

First, let O be an odd cycle in G that is not contained in $G[U]$. Suppose that $V(O) \cap U \neq \emptyset$, and let $\widehat{U}$ be a maximal non-empty subset of vertices from U that appear consecutively in O. Let $u, v \in V(O) \setminus U$ be the two vertices (possibly $u = v$) in O that are closest in O to $\widehat{U}$. Let P be the subpath of O between u and v that goes through $\widehat{U}$. Let $C \in \mathcal{C}$ be the connected component that contains $\widehat{U}$, and let x be the vertex that represents it in G'. We will slightly abuse notation and use x to refer also to $\mathsf{origin}_{G,<}(G,x)$. Let $p \in V(P)$ such that there exists a path P' between p and x that intersects P only in p (possibly $p = x$). Let $P_{u,p}$ and $P_{p,v}$ denote the subpaths of P between u and p and between p and v, respectively. Then, in G', $\{u,x\}$ will be assigned by $\mathfrak{lab}$ a subset that contains $((|E(P_{u,p})| + |E(P')|) \mod 2) + 1$, and $\{x,v\}$ will be assigned by $\mathfrak{lab}$ a subset that contains $((|E(P_{p,v})| + |E(P')|) \mod 2) + 1$, and the sum of these two labels modulo 2 equals $|E(P)| \mod 2$. So, by replacing in O each $\widehat{U}$ as above by its corresponding vertex x in U', we obtain a closed walk such that there exists a choice for a label for each one of its edges (where the label is chosen from the subset assigned to the edge by $\mathfrak{lab}$) such that the sum of labels is odd. This implies the correctness of the first claim.

Second, let O' be a cycle in G' such that there exists a choice for a label for each one of its edges (where the label is chosen from the subset assigned to the edge by $\mathfrak{lab}$) such that the sum of labels is odd. Suppose that there exists a vertex $x \in V(O') \cap U'$. We will slightly abuse notation and use x to refer also to $\mathsf{origin}_{G,<}(G,x)$. Let v be one of its neighbors in O', and note that $v \notin U'$. Let ℓ be the label chosen for $\{v,x\}$. Then, in G, there is a path $P_{v,x}$ between v and x whose number of edges modulo 2 plus 1 is ℓ. Then, for each such edge $\{v,x\}$, by replacing in O' the subpath consisting just of this edge by $P_{v,x}$, we obtain an odd closed walk in G. ◻

The last required component is the following, which can be proved using standard dynamic programming over nice tree decompositions (see [10]).

Proposition 6.32 (Similar to [10])**.** *There exists a* $2^{\mathcal{O}(w \log w)} n^{\mathcal{O}(1)}$*-time algorithm that, given an instance* $(G, \mathfrak{w}, \mathfrak{lab}, W, k)$ *of* Weighted Edge-Labeled Odd Cycle Transversal *such that* G *has treewidth at most* w, *determines whether the minimum-weight of* S *a solution of* $(G, \mathfrak{w}, \mathfrak{lab}, W, k)$ *is at most* W.

We are now ready to present the main component of the algorithm in this subsection.

Lemma 6.33. Enriched Odd Cycle Transversal *on disk graphs admits a randomized algorithm with running time* $2^{\mathcal{O}(\ell\log\ell)}\cdot n^{\mathcal{O}(1)}$ *where* $\ell = k^{\frac{3}{4}}\cdot p^6$.

Proof. We first present the algorithm, denoted by $\mathcal{A}$. Let (G,M,p,k) denote the input. (Note that the input includes neither the argument U (being empty) nor a vertex-weight function. The objective is to determine whether that exists an odd cycle transversal of size at most k.)

1. Let $X = \{u \in V(G)\setminus M : N_G(u) \subseteq M\}$. Let $(X_1, X_2 \dots, X_r)$, for some $r \in \mathbb{N}_0$, be a partition of X so that two vertices of X are in the same part if and only if they are false twins. For every $i \in \{1,2,\dots,r\}$, let x_i be some vertex (arbitrarily chosen) from X_i.
2. Call the algorithm in Proposition 6.27 on $G-(M\cup X)$ with integer $t = \lceil\sqrt{k}\rceil$. (By Definition 6.11 and Proposition 6.25, $G-M$ is triangle-free and hence planar, so the usage of Proposition 6.27 is legal.) Let $Z_1, Z_2, \dots, Z_t \subseteq V(G)\setminus(M\cup X)$ be its output sets.
3. Call the algorithm in Proposition 6.24 on (G,k). Let $S^\star$ be its output set.
4. Let $G^\star = G - (\bigcup_{i=1}^{r}(X_i\setminus\{x_i\})$. Let $\mathfrak{w}^\star : V(G^\star)\to\mathbb{N}_0$ be defined as follows: for every $v \in V(G^\star)\setminus X$, $\mathfrak{w}^\star(v) = 1$; for every x_i, $i\in\{1,2,\dots,r\}$, $\mathfrak{w}^\star(x_i) = |X_i|$. Let $W = k$.
5. For every $i \in \{1,2,\dots,t\}$:
   (a) For every subset $A \subseteq S^\star\cap Z_i$ of size at most $\sqrt{k}$:
      i. Let $<$ be some arbitrary order on $V(G^\star)$. Let $I^\star = (G^\star,\mathfrak{w}^\star,W',k')$. Then, let $I' = (G',\mathfrak{w}',\mathfrak{lab}',W',k')$ denote the $(<,U)$-edge-labeled instance corresponding to $I^\star$ where $U = Z_i\setminus A$. Let $U' = V(G')\setminus V(G^\star)$.
      ii. If $G^\star[U]$ contains an odd cycle: Proceed to the next iteraation.
      iii. Call the algorithm in Proposition 6.32 on I' with $\mathfrak{w}'$ updated to assign ∞ to every vertex in U'.
      iv. If the algorithm above outputs "yes-instance": Return "yes-instance".
6. Return "no-instance".

We first consider the correctness of the algorithm. On the one hand, suppose that (G,k) is a yes-instance. By Proposition 6.24, (G,k) admits a solution that is contained in $S^\star$. So, by Observation 6.26, $(G^\star,\mathfrak{w}^\star,W',k')$ admits a solution S such that $S\setminus(M\cup X)\subseteq S^\star$. Additionally, note that since $|S|\leq k$, there exists $i\in\{1,2,\dots,t\}$ such that $|S\cap Z_i|\leq\sqrt{k}$. So, for this i, there exists an iteration where $A = S\cap Z_i$. Hence, in this iteration, $S\cap U=\emptyset$, and so, $G^\star[U]$ does not have an odd cycle, and by Lemma 6.31, the call the algorithm in Proposition 6.32 returns "yes". Thus, algorithm $\mathcal{A}$ returns "yes-instance".

On the other hand, suppose that the algorithm returns "yes-instance". Hence, there exist i and A in whose iteration the computed instance I' of Weighted Edge-Labeled Odd Cycle Transversal admits a solution that is disjoint from U' and $G^\star[U]$ does not admit an odd cycle. In what follows, we consider this iteration. By Lemma 6.31, $(G^\star,\mathfrak{w}^\star,W',k')$ admits a solution that is disjoint from U. Then, from Observation 6.26, it follows that (G,k) is a yes-instance of Odd Cycle Transversal.

We now consider the running time of algorithm $\mathcal{A}$. Let $T_{\max}$ denote the maximum running time of any single iteration corresponding to some choice of i and A. Observe that, since $|S^\star|\in k^{\mathcal{O}(1)}$ and since the algorithms in Propositions 6.27 and 6.27 run in polynomial time, we derive that the time complexity of $\mathcal{A}$ is bounded from above by $(\sum_{j=0}^{\lfloor\sqrt{k}\rfloor}\binom{S^\star}{j})\cdot T_{\max}\cdot n^{\mathcal{O}(1)} \subseteq 2^{\mathcal{O}(\sqrt{k}\log k)}\cdot T_{\max}\cdot n^{\mathcal{O}(1)}$. Hence, to complete the proof, we fix an iteration corresponding to some i and A, and show that its running time, denoted by T, is bounded from above by $2^{\mathcal{O}(\ell\log\ell)}\cdot n^{\mathcal{O}(1)}$. For this purpose, let w' denote the treewidth of G'. Then, from the pseudocode and the time complexity of the algorithm in Proposition 6.21, we derive that $T\in 2^{\mathcal{O}(w'\log w')}n^{\mathcal{O}(1)}$. So, to complete the proof, it remains to show that $w'\in\mathcal{O}(\ell)$.

For the purpose above, let $M' = M \cup \{x_1, x_2, \dots, x_r\}$. Let w denote the treewidth of $(G^\star - M')/U$. By Definition 6.11, it follows that any realization of $G^\star$ has ply at most p. Moreover, by Definition 6.11, it follows that $|M| \in \mathcal{O}(p \cdot k)$, and for every $v \in V(G^\star)$, $N_{G^\star}(v) \setminus M$ is an independent set. So, by Theorem 1, $r \in \mathcal{O}(|M| \cdot p^6) \subseteq \mathcal{O}(k \cdot p^7)$. Observe that, by the definition of $G^\star$, there does not exist a vertex in $V(G^\star) \setminus M'$ whose neighborhood is contained in M'. So, we can apply Theorem 2, and derive that $w' \in \mathcal{O}(\max\{\sqrt{|M'| \cdot w} \cdot p^{2.5}, w \cdot p\}) \subseteq \mathcal{O}(\max\{\sqrt{k \cdot w} \cdot p^6, w \cdot p\})$. By Proposition 6.27, $w \in \mathcal{O}(r + |A|) \leq \mathcal{O}(\sqrt{k})$. Hence, we further get that $w' \in \mathcal{O}(\max\{\sqrt{k \cdot \sqrt{k}} \cdot p^6, \sqrt{k} \cdot p\}) \subseteq \mathcal{O}(\ell)$. As argued above, this completes the proof. □

We conclude with our main result for Odd Cycle Transversal.

Lemma 6.34. Odd Cycle Transversal *on disk graphs admits a randomized algorithm with running time* $2^{\mathcal{O}(k^{\frac{27}{28}} \log k)} \cdot n^{\mathcal{O}(1)}$.

Proof. The correctness follows from Lemmata 6.13 and 6.33 with $p = k^{\frac{1}{28}}$. □

7 Conclusion

In this paper, we presented two new combinatorial theorems concerning disk graphs. These combinatorial theorems are of particular algorithmic interest, as they can be used to yield sublinear bounds on the treewidth of a given disk graph in terms of solution size. By making use of them, we derived the first subexponential-time FPT algorithms for three central problems in Parameterized Complexity on disk graphs, namely, Triangle Hitting, Feedback Vertex Set and Odd Cycle Transversal. Our algorithms do not require a geometric realization of the input graph, and they generalize to the weighted and counting versions of all aforementioned problems except for Odd Cycle Transversal. Notably, this is the first time that the research frontiers in Parameterized Complexity for planar and disk graphs are unified. Moreover, these are the first subexponential-time FPT algorithms for problems on disk graphs that are not known to admit linear-vertex kernels (in those cases, one just has to design a subexponential-time algorithm in the number of vertices).

While we have made the first step in the systematic study of the design of subexponential-time FPT algorithms on disk graphs, much remains to be explored. In particular, we pose the following questions.

1. The three problems studied in this paper are vertex-deletion problems to graph classes that are triangle-free. Can we design subexponential-time FPT algorithms for (non-trivial) vertex-deletion problems (on disk graphs) to graph classes that are not triangle-free? Given that we are able to resolve Feedback Vertex Set on disk graphs, the natural next step is to try to resolve Treewidth η-Deletion for every fixed $\eta \in \mathbb{N}$ on disk graphs (given a disk graph G, remove at most k vertices from G to turn it into a graph of treewidth at most η). More broadly, we propose to consider the Planar $\mathcal{F}$-Deletion problem [20] on disk graphs. For this purpose, the work of [5] may come in handy.

2. Besides vertex-deletion problems, other central problems in Parameterized Complexity concern connectivity and packing problems. In particular, two central connectivity and packing problems in Parameteried Complexity are k-Path and Cycle Packing. Do these problems admit subexponential-time FPT algorithms on disk graphs?

3. On planar and unit disk graphs, various problems are known to admit subexponential-time FPT algorithm with running times of the form $2^{\mathcal{O}(\sqrt{k} \cdot \log^{\mathcal{O}(1)} k)} \cdot n^{\mathcal{O}(1)}$ (or even $2^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$), which are tight under the ETH (up to polylog factors in the exponent). We believe that our algorithms can be sped-up; indeed, we have not tried to optimize the power of k in the exponent. However, it is not clear how —and whether it is even possible—to attain a running time of $2^{\mathcal{O}(\sqrt{k} \cdot \log^{\mathcal{O}(1)} k)} \cdot n^{\mathcal{O}(1)}$. So, we pose the following question: under the ETH (and up to polylog factors in the exponent), what is the best running times that can be attained for Triangle Hitting, Feedback Vertex Set and Odd Cycle Transversal on disk graphs?

References

[1] P. K. Agarwal, M. J. van Kreveld, and S. Suri, *Label placement by maximum independent set in rectangles*, Comput. Geom., 11 (1998), pp. 209–218. 1

[2] J. Alber, H. Fernau, and R. Niedermeier, *Graph separators: a parameterized view*, J. Comput. Syst. Sci., 67 (2003), pp. 808–832. 1

[3] J. Alber and J. Fiala, *Geometric separation and exact solutions for the parameterized independent set problem on disk graphs*, J. Algorithms, 52 (2004), pp. 134–151. 1

[4] S. Bandyapadhyay, W. Lochet, D. Lokshtanov, S. Saurabh, and J. Xue, *Subexponential parameterized algorithms for cut and cycle hitting problems on h-minor-free graphs*, in 33rd ACM-SIAM Symposium on Discrete Algorithms, January 9-12, 2022, Alexandria, Virginia, USA, 2022, p. TBA. 33

[5] J. Baste, I. Sau, and D. M. Thilikos, *Hitting minors on bounded treewidth graphs. i. general upper bounds*, SIAM J. Discret. Math., 34 (2020), pp. 1623–1648. 36

[6] J. Baste and D. M. Thilikos, *Contraction-bidimensionality of geometric intersection graphs*, in 12th International Symposium on Parameterized and Exact Computation, IPEC 2017, September 6-8, 2017, Vienna, Austria, D. Lokshtanov and N. Nishimura, eds., vol. 89 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, pp. 5:1–5:13. 1

[7] A. Becker and D. Geiger, *Optimization of pearl's method of conditioning and greedy-like approximation algorithms for the vertex feedback set problem*, Artificial Intelligence, 83 (1996), pp. 167–188. 31

[8] M. Bonamy, É. Bonnet, N. Bousquet, P. Charbit, P. Giannopoulos, E. J. Kim, P. Rzazewski, F. Sikora, and S. Thomassé, *EPTAS and subexponential algorithm for maximum clique on disk and unit ball graphs*, J. ACM, 68 (2021), pp. 9:1–9:38. 13, 25

[9] É. Bonnet and P. Rzazewski, *Optimality program in segment and string graphs*, Algorithmica, 81 (2019), pp. 3047–3073. 1, 2

[10] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh, *Parameterized Algorithms*, Springer, 2015. 1, 2, 4, 7, 12, 30, 32, 34

[11] ——, *Parameterized Algorithms*, Springer, 2015.

[12] M. de Berg, H. L. Bodlaender, S. Kisfaludi-Bak, D. Marx, and T. C. van der Zanden, *A framework for exponential-time-hypothesis-tight algorithms and lower bounds in geometric intersection graphs*, SIAM J. Comput., 49 (2020), pp. 1291–1331. 2

[13] E. D. Demaine, F. V. Fomin, M. T. Hajiaghayi, and D. M. Thilikos, *Bidimensional parameters and local treewidth*, SIAM J. Discret. Math., 18 (2004), pp. 501–511. 1

[14] ——, *Fixed-parameter algorithms for* (k, r)*-center in planar graphs and map graphs*, ACM Trans. Algorithms, 1 (2005), pp. 33–47. 1

[15] ——, *Subexponential parameterized algorithms on bounded-genus graphs and* H*-minor-free graphs*, J. ACM, 52 (2005), pp. 866–893. 1, 4

[16] E. D. Demaine and M. Hajiaghayi, *The bidimensionality theory and its algorithmic applications*, Comput. J., 51 (2008), pp. 292–302. 1

[17] ——, *Linearity of grid minors in treewidth with applications through bidimensionality*, Comb., 28 (2008), pp. 19–36. 1

[18] R. Diestel, *Graph Theory, 4th Edition*, vol. 173 of Graduate texts in mathematics, Springer, 2012. 11

[19] J. Edmonds, *Paths, trees, and flowers*, Canadian Journal of mathematics, 17 (1965), pp. 449–467. 27

[20] F. V. Fomin, D. Lokshtanov, N. Misra, and S. Saurabh, *Planar f-deletion: Approximation, kernelization and optimal FPT algorithms*, in 53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012, 2012, pp. 470–479. 36

[21] F. V. Fomin, D. Lokshtanov, F. Panolan, S. Saurabh, and M. Zehavi, *Decomposition of map graphs with applications*, in 46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece, C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi, eds., vol. 132 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, pp. 60:1–60:15. 1, 2

[22] ——, *Finding, hitting and packing cycles in subexponential time on unit disk graphs*, Discret. Comput. Geom., 62 (2019), pp. 879–911. 1, 2, 4

[23] ——, *Eth-tight algorithms for long path and cycle on unit disk graphs*, in 36th International Symposium on Computational Geometry, SoCG 2020, June 23-26, 2020, Zürich, Switzerland, S. Cabello and D. Z. Chen, eds., vol. 164 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, pp. 44:1–44:18. 1, 2

[24] F. V. Fomin, D. Lokshtanov, and S. Saurabh, *Excluded grid minors and efficient polynomial-time approximation schemes*, J. ACM, 65 (2018), pp. 10:1–10:44. 1, 2, 4

[25] A. Grigoriev, A. Koutsonas, and D. M. Thilikos, *Bidimensionality of geometric intersection graphs*, in SOFSEM 2014: Theory and Practice of Computer Science - 40th International Conference on Current Trends in Theory and Practice of Computer Science, Nový Smokovec, Slovakia, January 26-29, 2014, Proceedings, V. Geffert, B. Preneel, B. Rovan, J. Stuller, and A. M. Tjoa, eds., vol. 8327 of Lecture Notes in Computer Science, Springer, 2014, pp. 293–305. 1

[26] Q. Gu and H. Tamaki, *Improved bounds on the planar branchwidth with respect to the largest grid minor size*, Algorithmica, 64 (2012), pp. 416–453. 13

[27] P. Hliněný and J. Kratochvíl, *Representing graphs by disks and balls (a survey of recognition-complexity results)*, Discret. Math., 229 (2001), pp. 101–124. 13

[28] Y. Iwata and Y. Kobayashi, *Improved analysis of highest-degree branching for feedback vertex set*, in 14th International Symposium on Parameterized and Exact Computation, IPEC 2019, September 11-13, 2019, Munich, Germany, 2019, pp. 22:1–22:11. 4

[29] H. Kaplan, K. Klost, W. Mulzer, L. Roditty, P. Seiferth, and M. Sharir, *Triangles and girth in disk graphs and transmission graphs*, in 27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany, 2019, pp. 64:1–64:14. 33

[30] F. Kuhn, R. Wattenhofer, and A. Zollinger, *Ad hoc networks beyond unit disk graphs*, Wirel. Networks, 14 (2008), pp. 715–729. 1

[31] J. Li and J. Nederlof, *Detecting feedback vertex sets of size* k *in* $O^*(2.7^k)$ *time*, in Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020, 2020, pp. 971–989. 4

[32] D. Lokshtanov, N. Narayanaswamy, V. Raman, M. Ramanujan, and S. Saurabh, *Faster parameterized algorithms using linear programming*, ACM Transactions on Algorithms (TALG), 11 (2014), pp. 1–31. 4

[33] D. Lokshtanov, S. Saurabh, and M. Wahlström, *Subexponential parameterized odd cycle transversal on planar graphs*, in IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India, 2012, pp. 424–434. 4, 33

[34] D. Marx, *The square root phenomenon in planar graphs*, in Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part II, F. V. Fomin, R. Freivalds, M. Z. Kwiatkowska, and D. Peleg, eds., vol. 7966 of Lecture Notes in Computer Science, Springer, 2013, p. 28. 1

[35] ——, *The square root phenomenon in planar graphs*, in Frontiers in Algorithmics *and* Algorithmic Aspects in Information and Management, Third Joint International Conference, FAW-AAIM 2013, Dalian, China, June 26-28, 2013. Proceedings, M. R. Fellows, X. Tan, and B. Zhu, eds., vol. 7924 of Lecture Notes in Computer Science, Springer, 2013, p. 1. 1

[36] D. Marx and M. Pilipczuk, *Optimal parameterized algorithms for planar facility location problems using voronoi diagrams*, in Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings, N. Bansal and I. Finocchi, eds., vol. 9294 of Lecture Notes in Computer Science, Springer, 2015, pp. 865–877. 1

[37] F. Panolan, S. Saurabh, and M. Zehavi, *Contraction decomposition in unit disk graphs and algorithmic applications in parameterized complexity*, in Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019, T. M. Chan, ed., SIAM, 2019, pp. 1035–1054. 1, 2

[38] ——, *Parameterized computational geometry via decomposition theorems*, in WALCOM: Algorithms and Computation - 13th International Conference, WALCOM 2019, Guwahati, India, February 27 - March 2, 2019, Proceedings, G. K. Das, P. S. Mandal, K. Mukhopadhyaya, and S. Nakano, eds., vol. 11355 of Lecture Notes in Computer Science, Springer, 2019, pp. 15–27. 1

[39] W. D. Smith and N. C. Wormald, *Geometric separator theorems & applications*, in 39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA, 1998, pp. 232–243. 14

[40] M. Wahlström, *Algorithms, measures and upper bounds for satisfiability and related problems*, PhD thesis, Department of Computer and Information Science, Linköpings universitet, 2007. 4

[41] D. P. Williamson and D. B. Shmoys, *The design of approximation algorithms*, Cambridge university press, 2011. 29

[42] J. Xu and B. Berger, *Fast and accurate algorithms for protein side-chain packing*, J. ACM, 53 (2006), pp. 533–557. 1